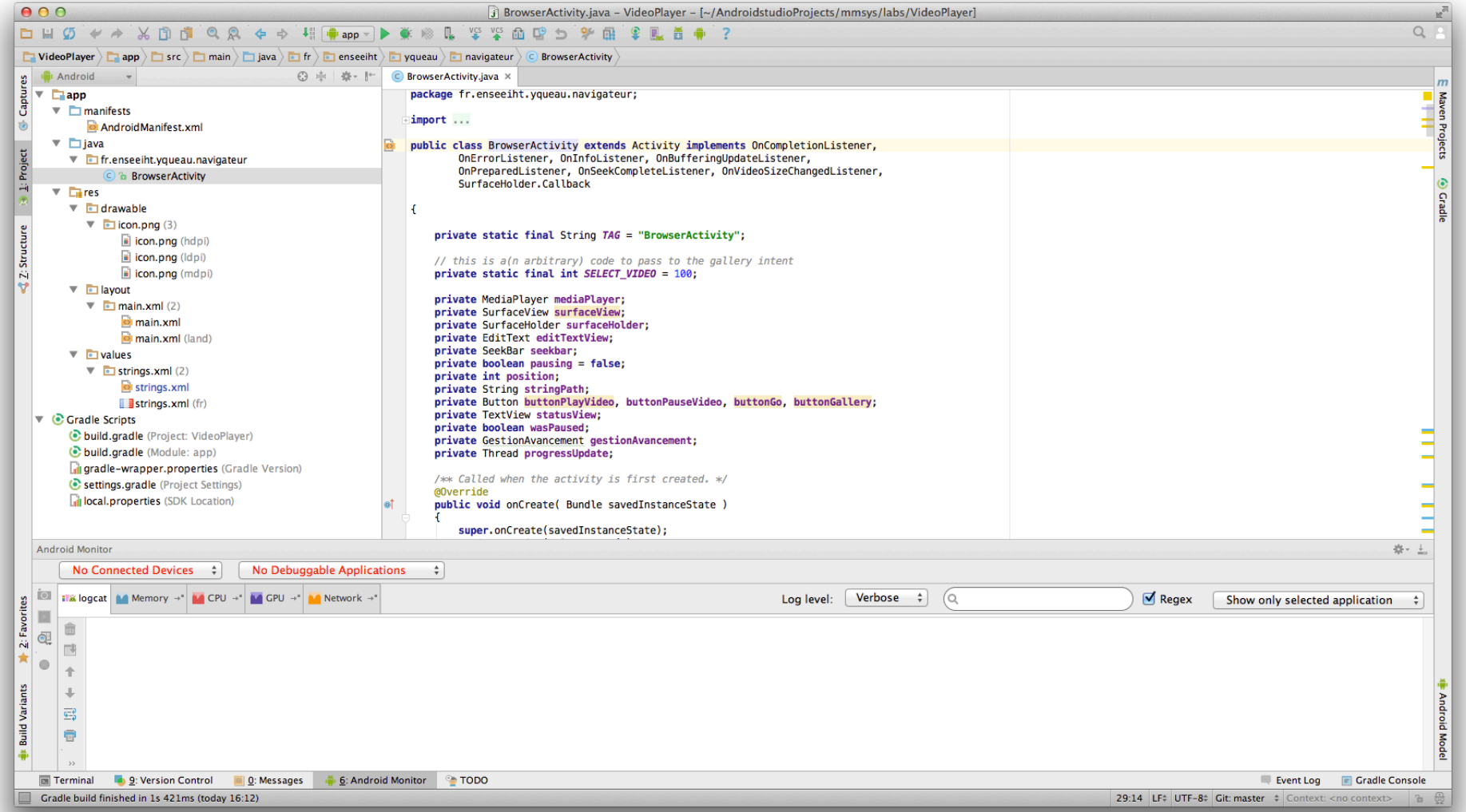
Plan

- Mobile devices
- Android overview
  - History
  - Architecture
  - Applications
- **The tools**
- Android Applications

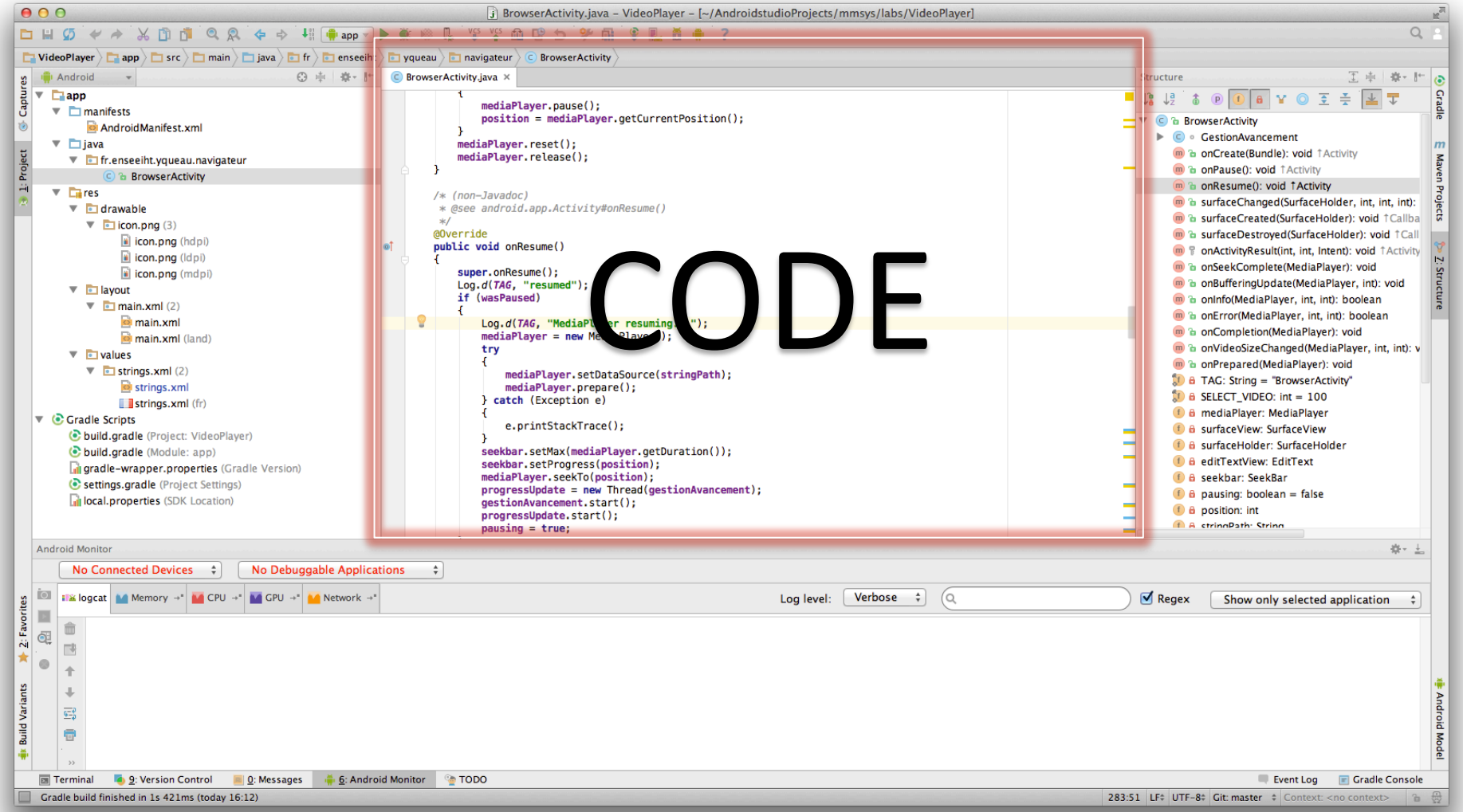
Development utilities

- **Android Studio**
  - Based on IntelliJ IDEA IDE for Java
  - Wizard for creating a new project
  - GUI editor (WYSIWYG)
  - Automated compilation process
  - Logging, debugging, code profiling ...
- **Android SDK**
  - API
  - Device emulators
  - Utilities for debugging, signing the code etc..
  - Sample code and Demos

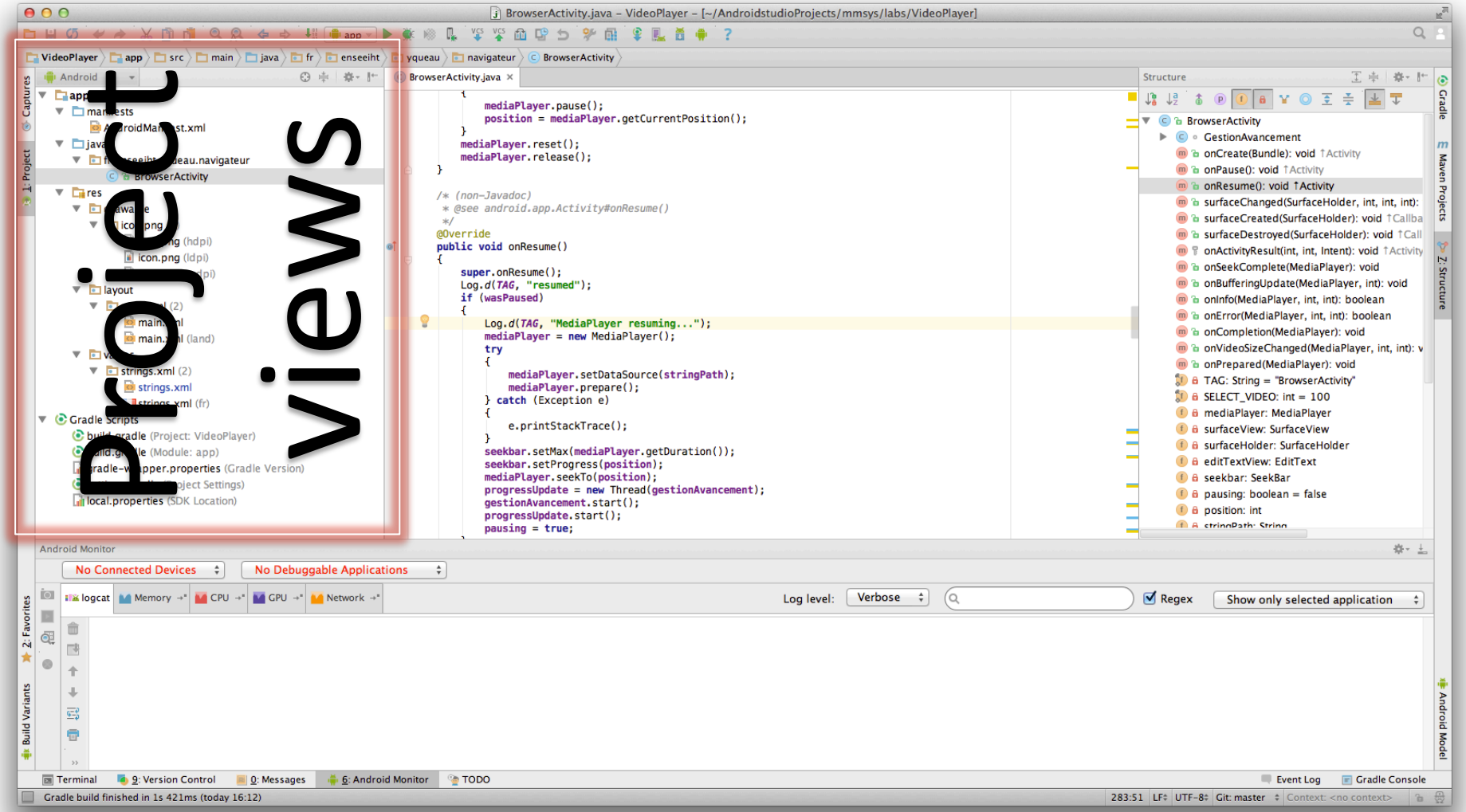
Android Studio



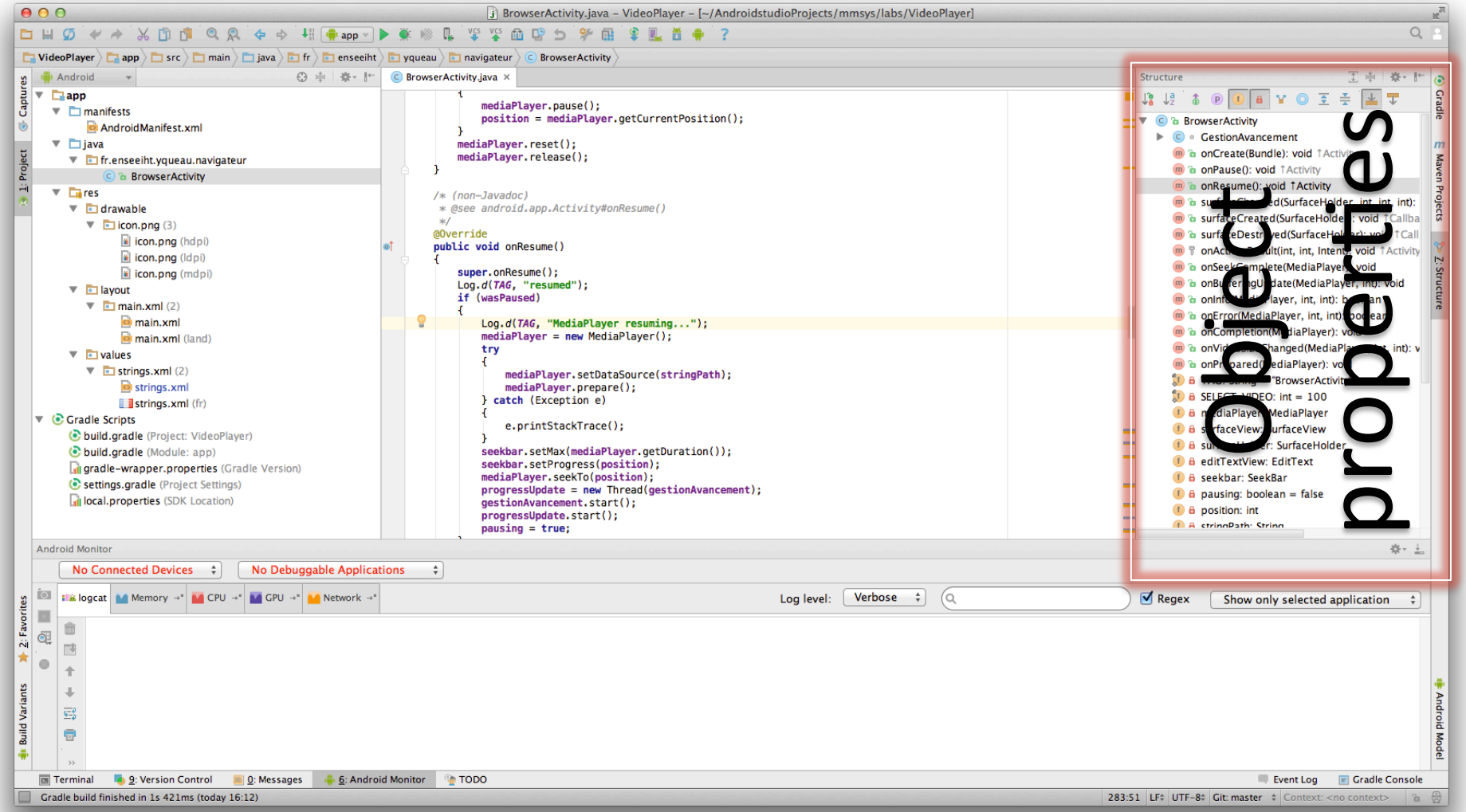
Android Studio



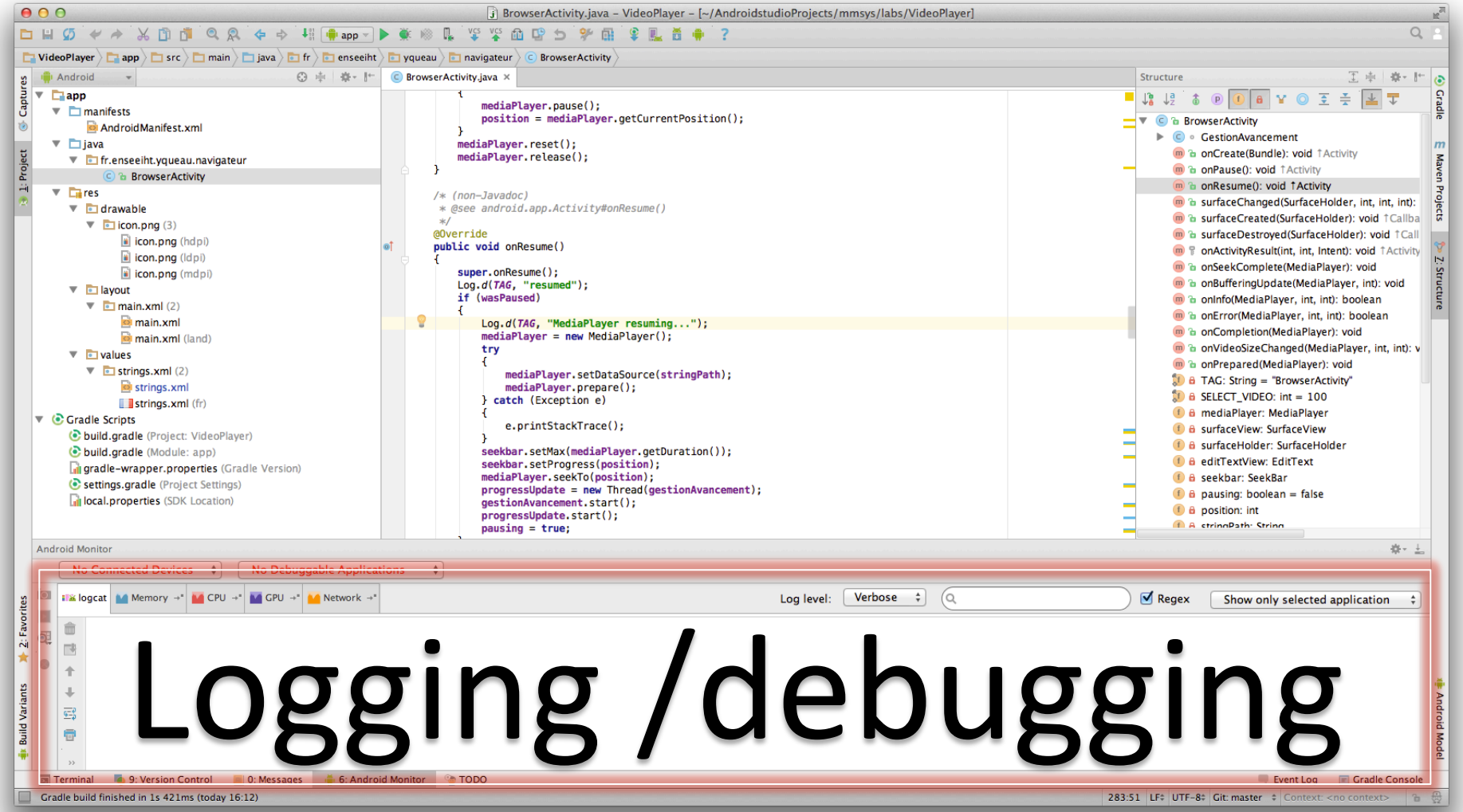
Android Studio



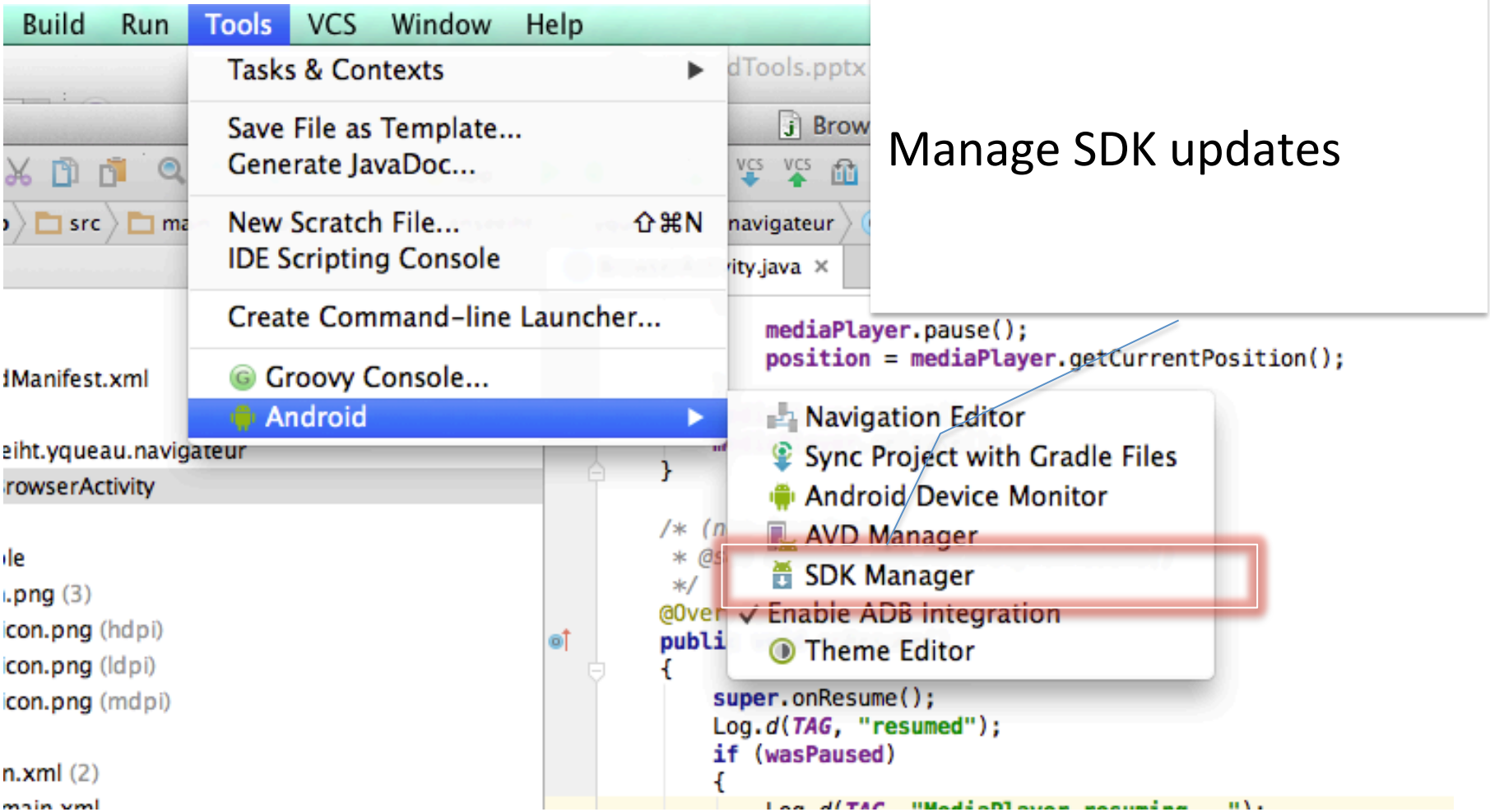
Android Studio



Android Studio

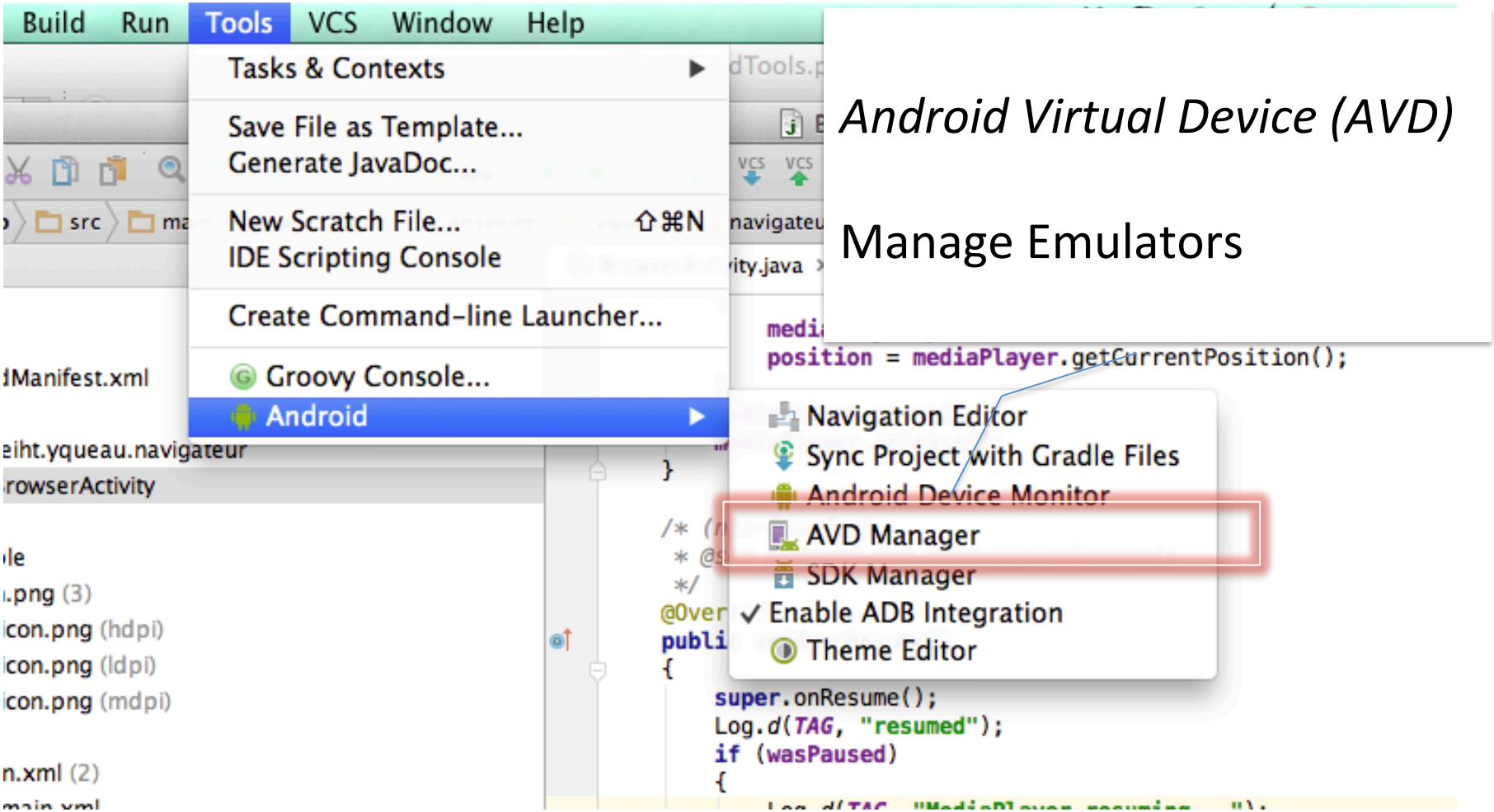


Android Studio

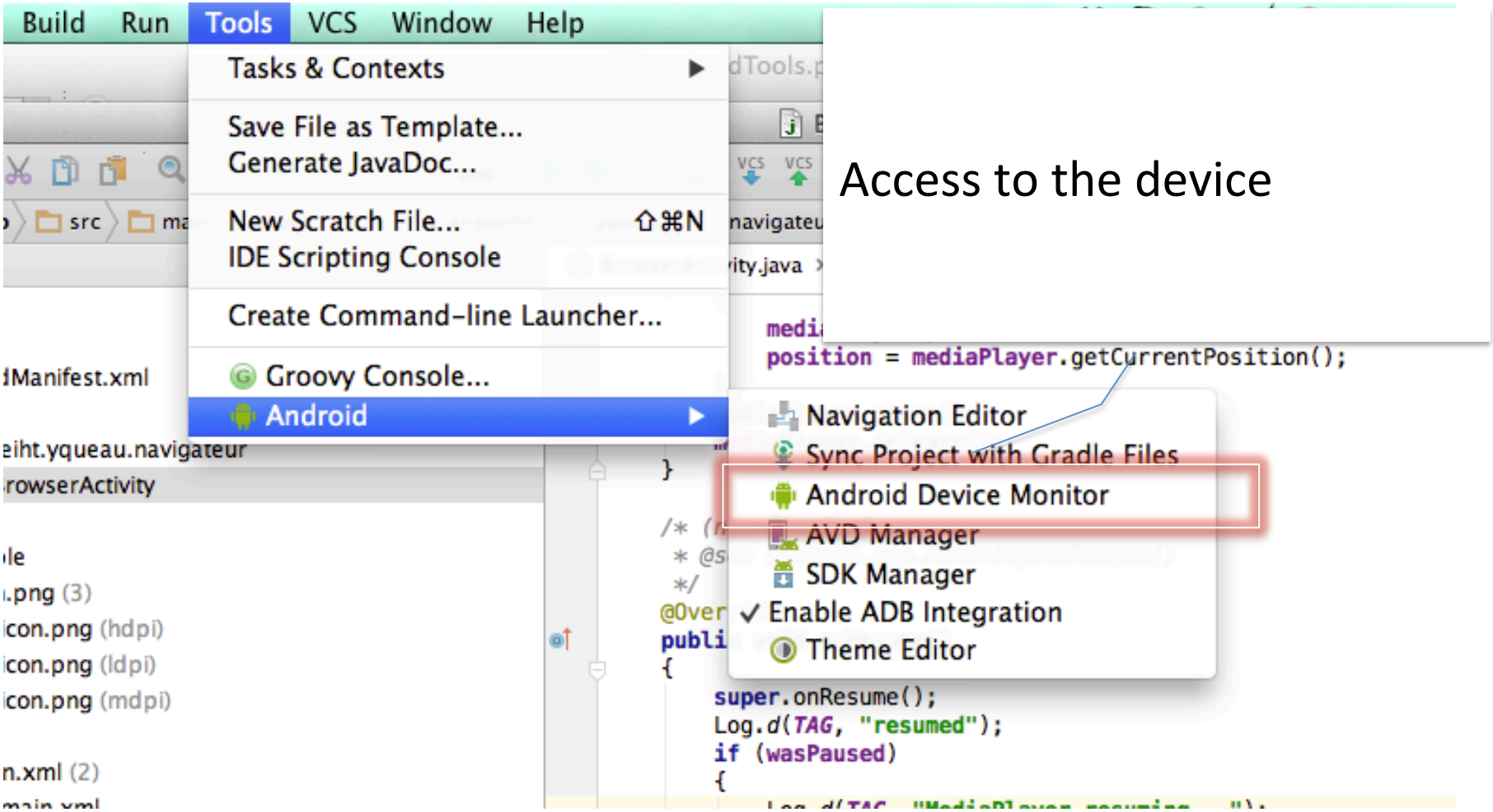


Manage SDK updates

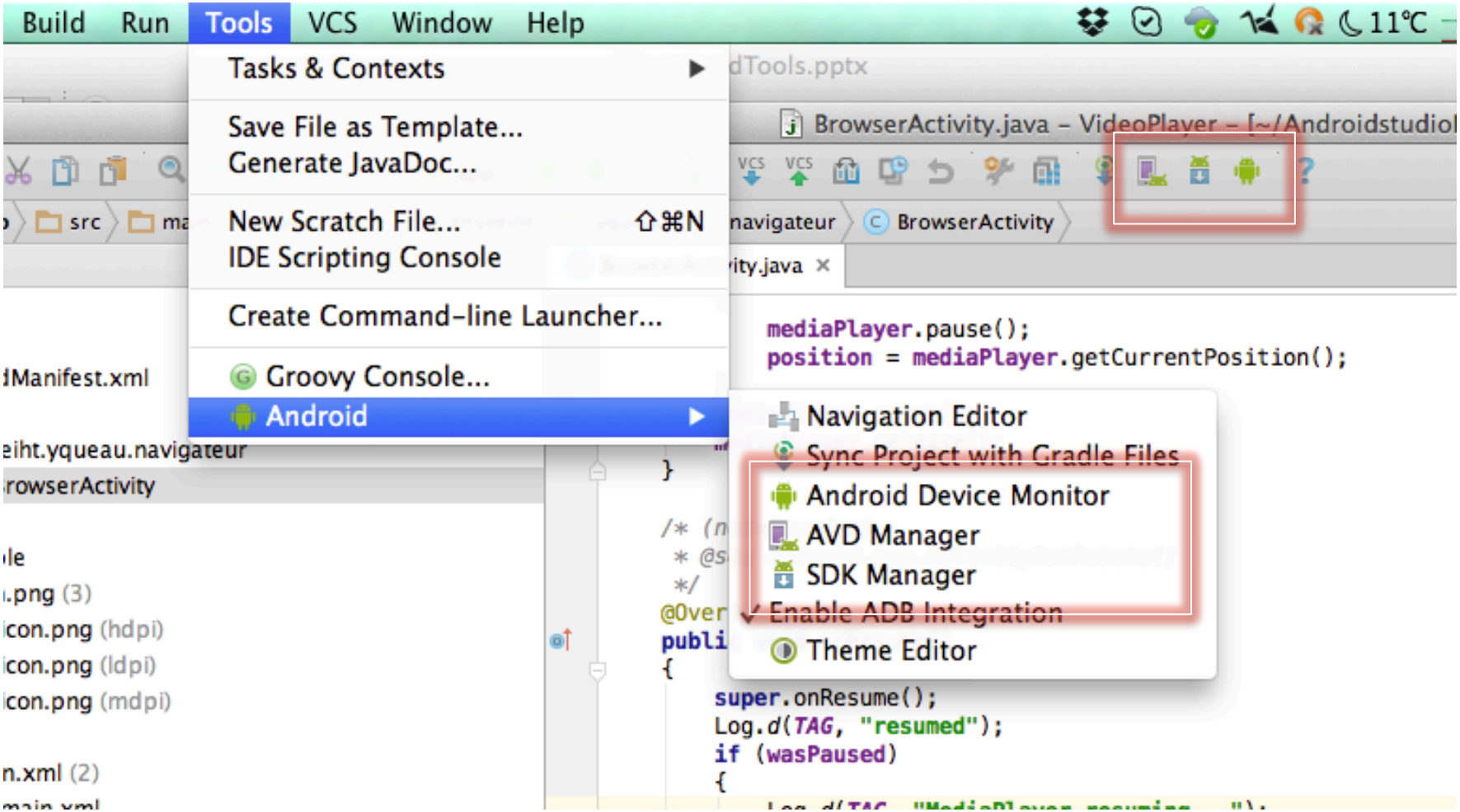
Android Studio



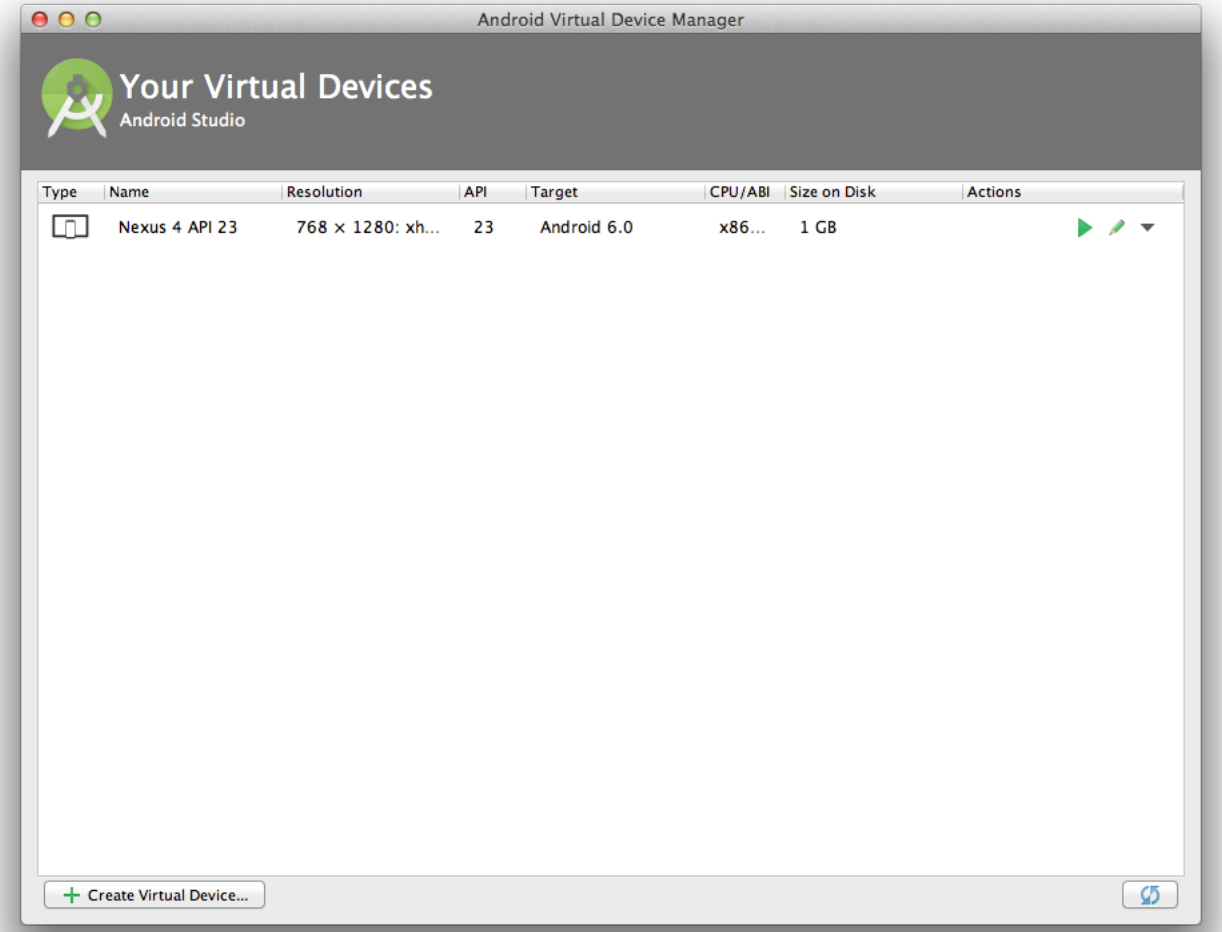
Android Studio



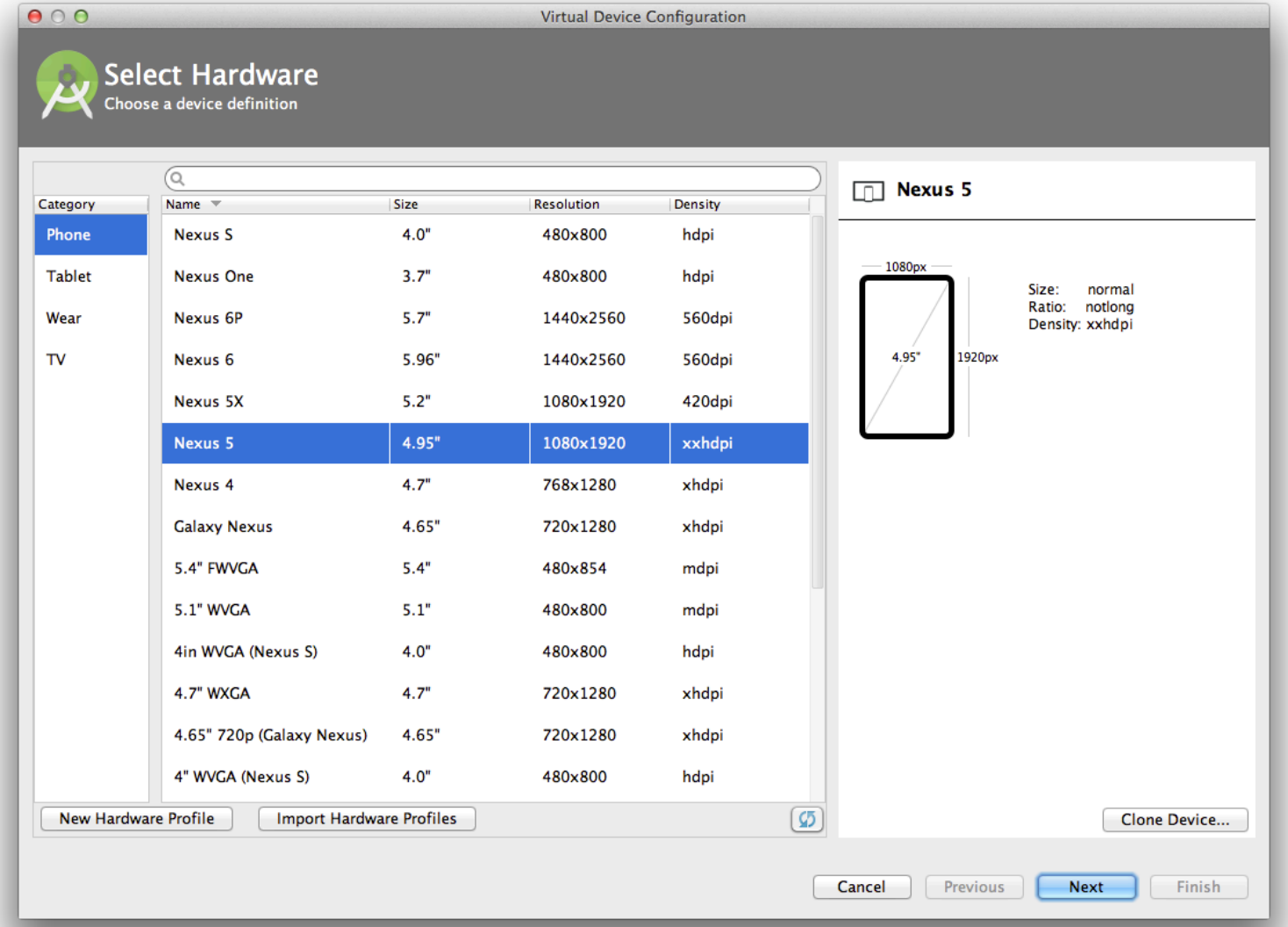
Android Studio



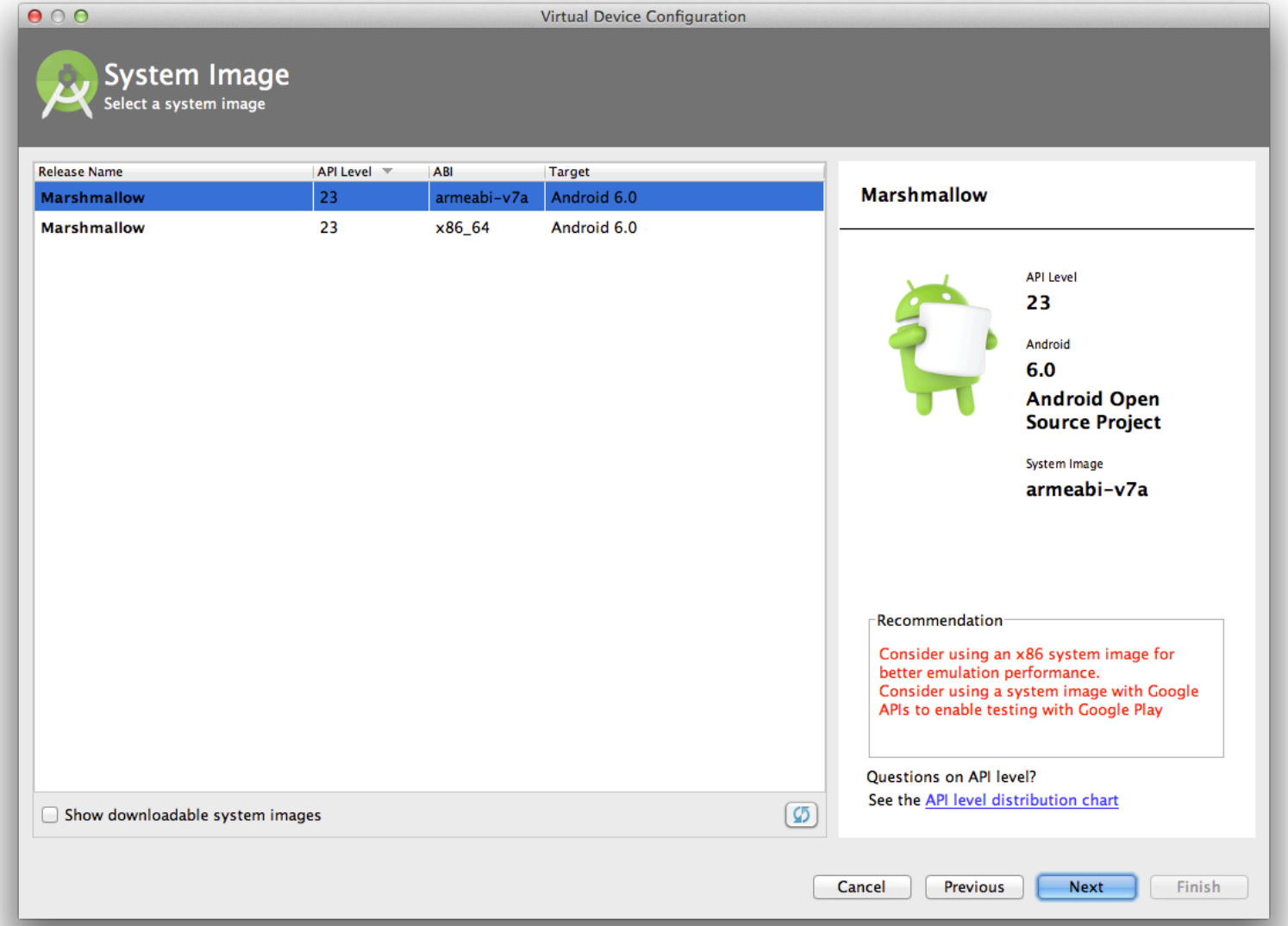
Android Studio – AVD Manager



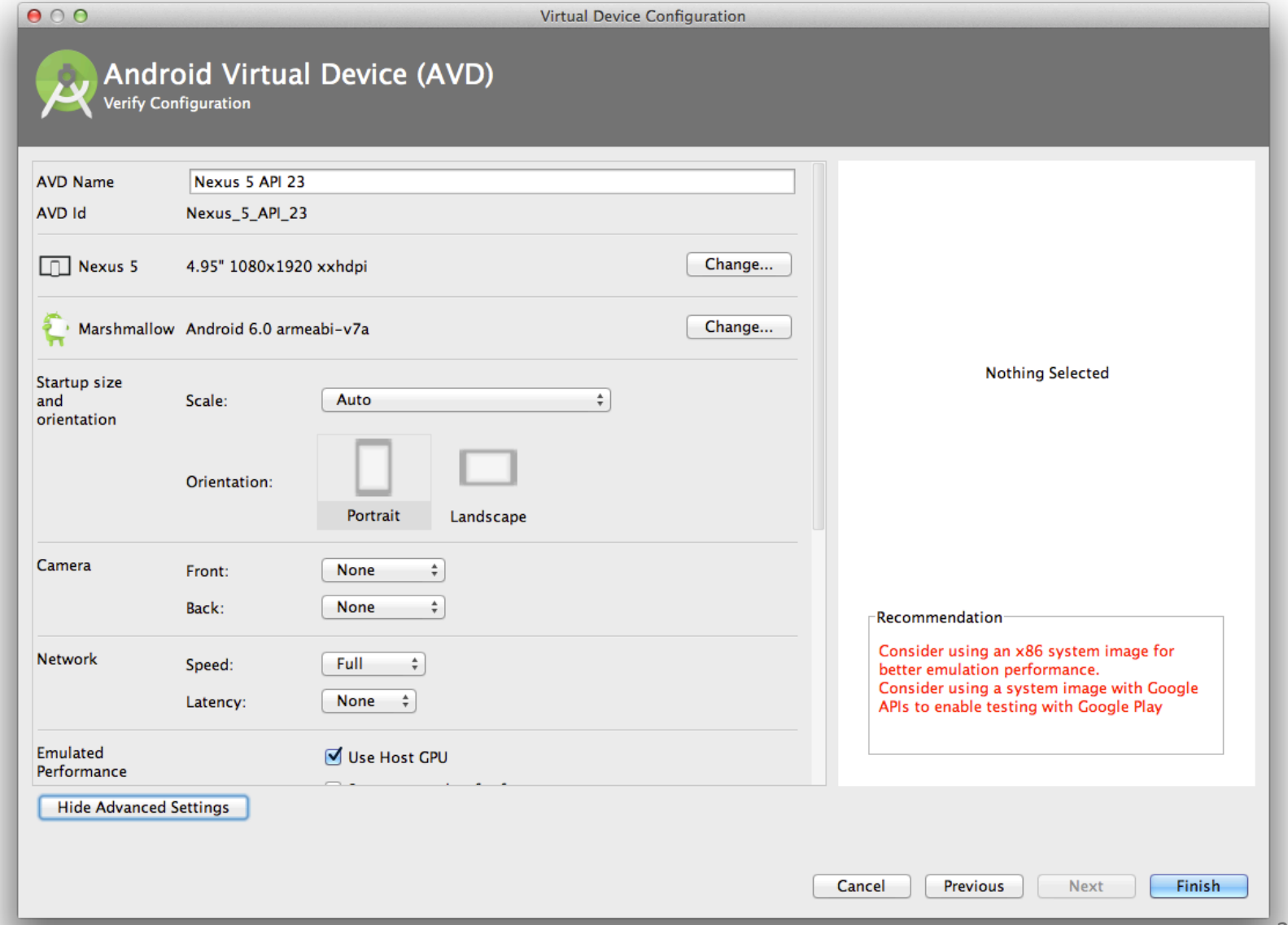
Android Studio – AVD



Android Studio – AVD

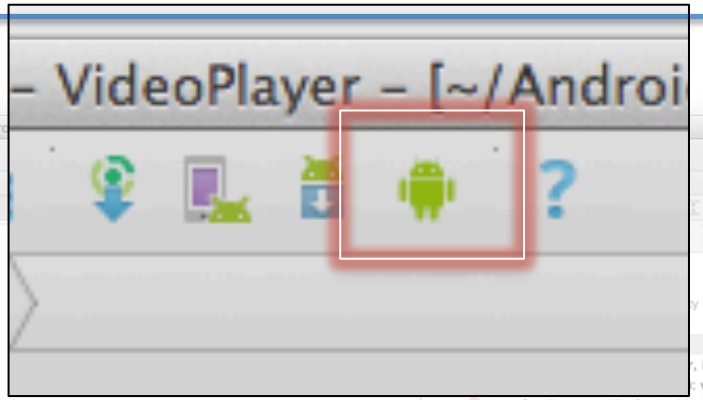


Android Studio – AVD



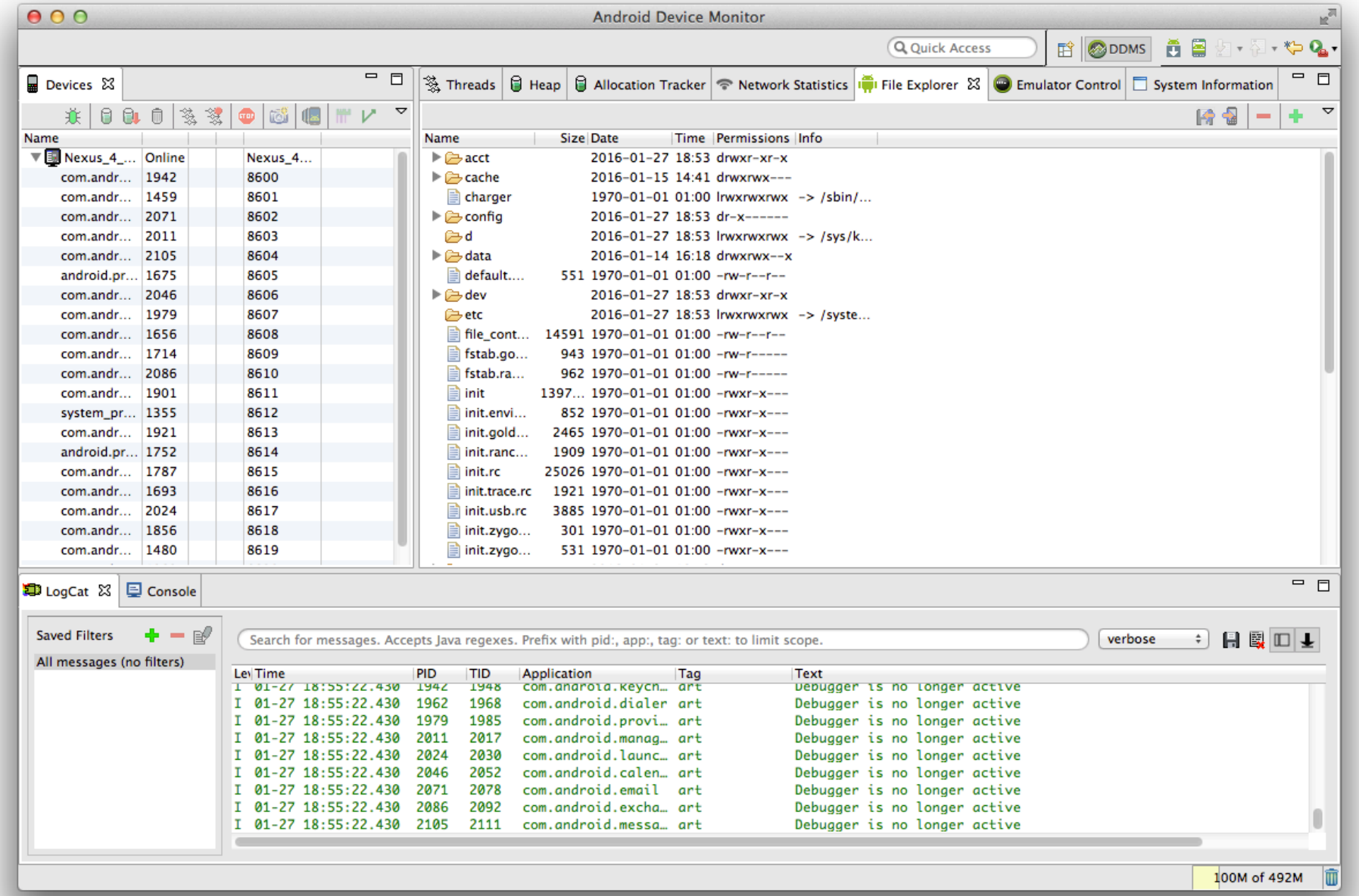
Android Studio – AVD

Android Device Monitor

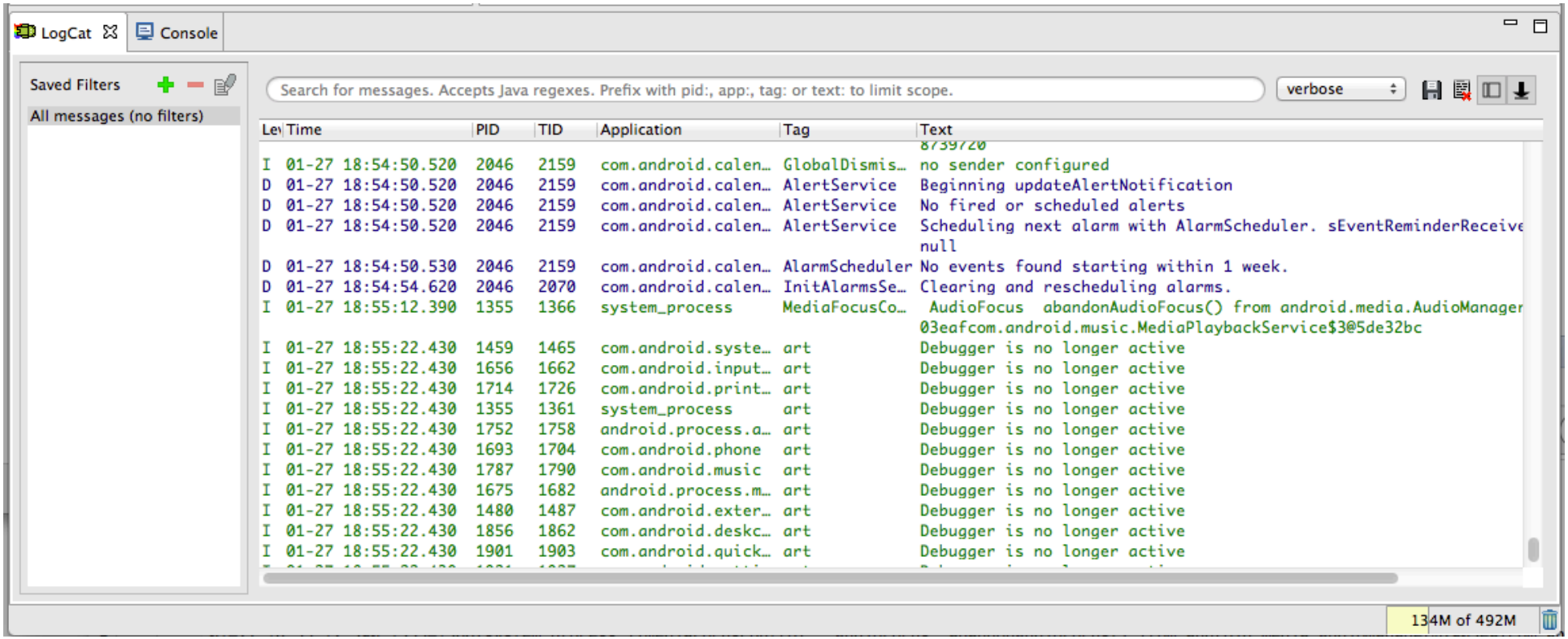


- debugging tools
- screen capture on the device,
- thread and heap information on the device,
- logcat,
- process, and radio state information,
- incoming call and SMS spoofing, location data spoofing, and more

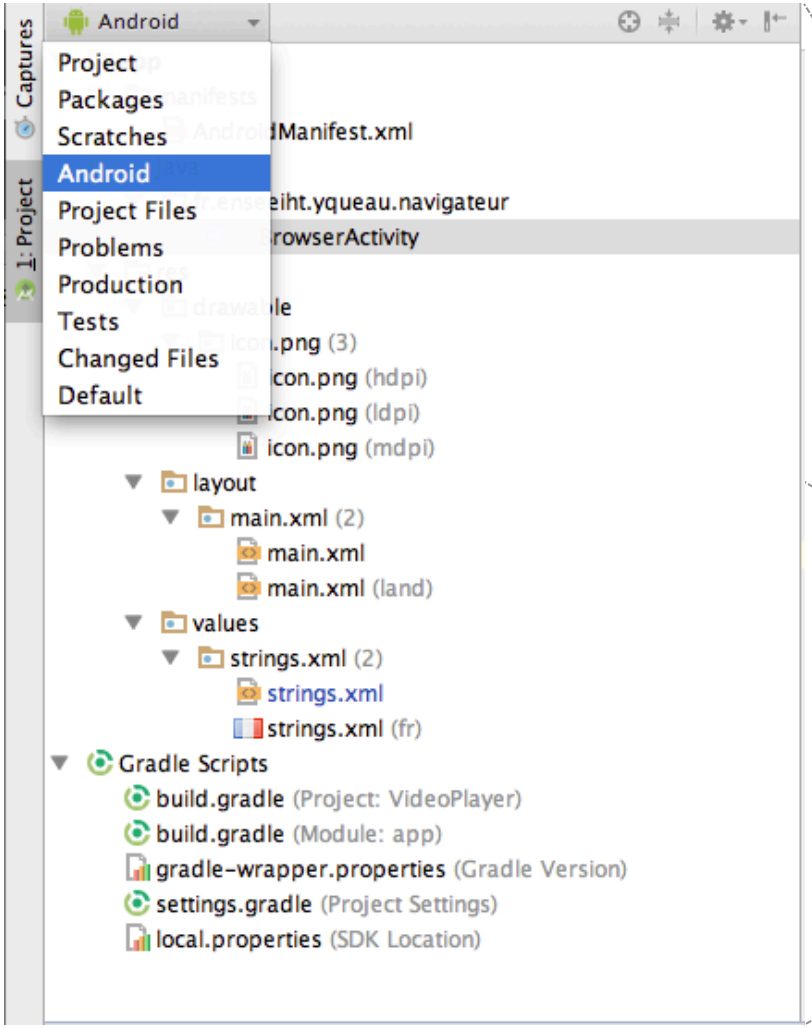
Android Device Monitor



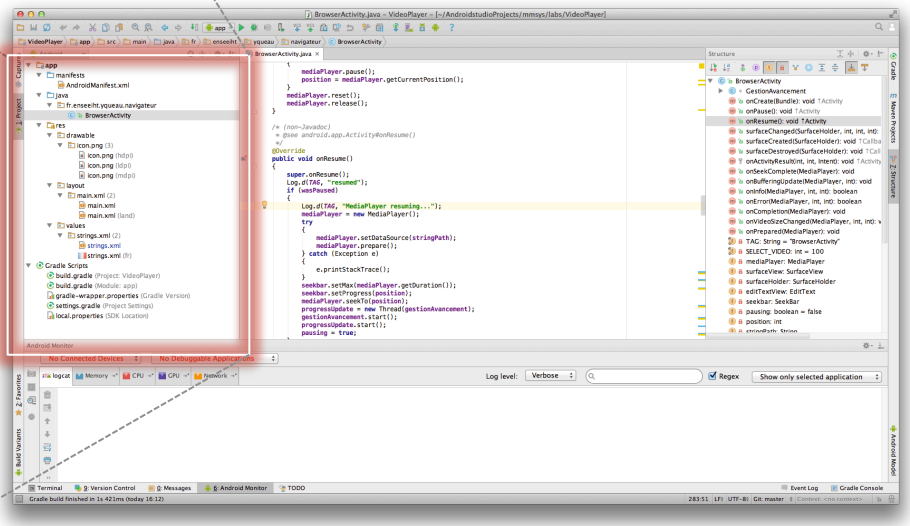
LogCat


LogCat
Console
Saved Filters
All messages (no filters)
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.
verbose
Le Time PID TID Application Tag Text
I 01-27 18:54:50.520 2046 2159 com.android.calen… GlobalDismis… no sender configured
D 01-27 18:54:50.520 2046 2159 com.android.calen… AlertService Beginning updateAlertNotification
D 01-27 18:54:50.520 2046 2159 com.android.calen… AlertService No fired or scheduled alerts
D 01-27 18:54:50.520 2046 2159 com.android.calen… AlertService Scheduling next alarm with AlarmScheduler. sEventReminderReceive
null
D 01-27 18:54:50.530 2046 2159 com.android.calen… AlarmScheduler No events found starting within 1 week.
D 01-27 18:54:54.620 2046 2070 com.android.calen… InitAlarmsSe… Clearing and rescheduling alarms.
I 01-27 18:55:12.390 1355 1366 system_process MediaFocusCo… AudioFocus abandonAudioFocus() from android.media.AudioManager
03eafcom.android.music.MediaPlaybackService$3@5de32bc
I 01-27 18:55:22.430 1459 1465 com.android.syste… art Debugger is no longer active
I 01-27 18:55:22.430 1656 1662 com.android.input… art Debugger is no longer active
I 01-27 18:55:22.430 1714 1726 com.android.print… art Debugger is no longer active
I 01-27 18:55:22.430 1355 1361 system_process art Debugger is no longer active
I 01-27 18:55:22.430 1752 1758 android.process.a… art Debugger is no longer active
I 01-27 18:55:22.430 1693 1704 com.android.phone art Debugger is no longer active
I 01-27 18:55:22.430 1787 1790 com.android.music art Debugger is no longer active
I 01-27 18:55:22.430 1675 1682 android.process.m… art Debugger is no longer active
I 01-27 18:55:22.430 1480 1487 com.android.exter… art Debugger is no longer active
I 01-27 18:55:22.430 1856 1862 com.android.deskc… art Debugger is no longer active
I 01-27 18:55:22.430 1901 1903 com.android.quick… art Debugger is no longer active
134M of 492M

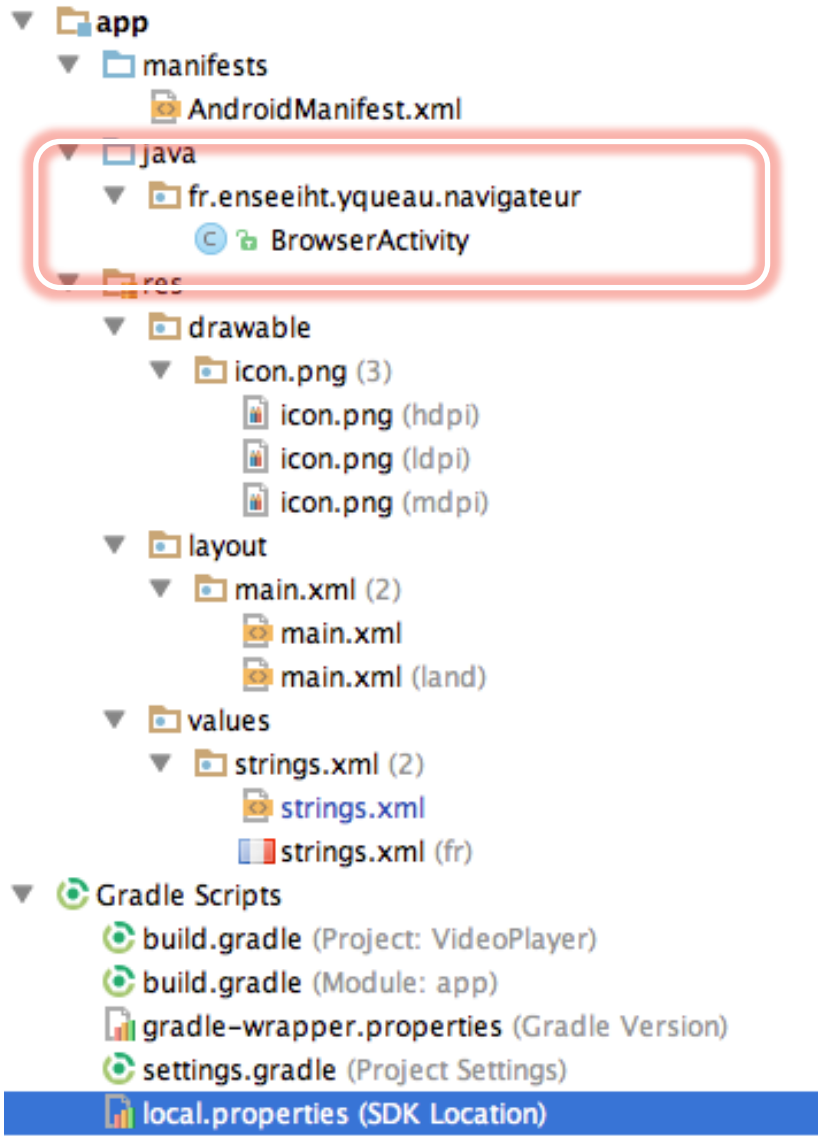
Android applications - Structure



- Android Studio has different "views" on the app structure
- Real structure may be different



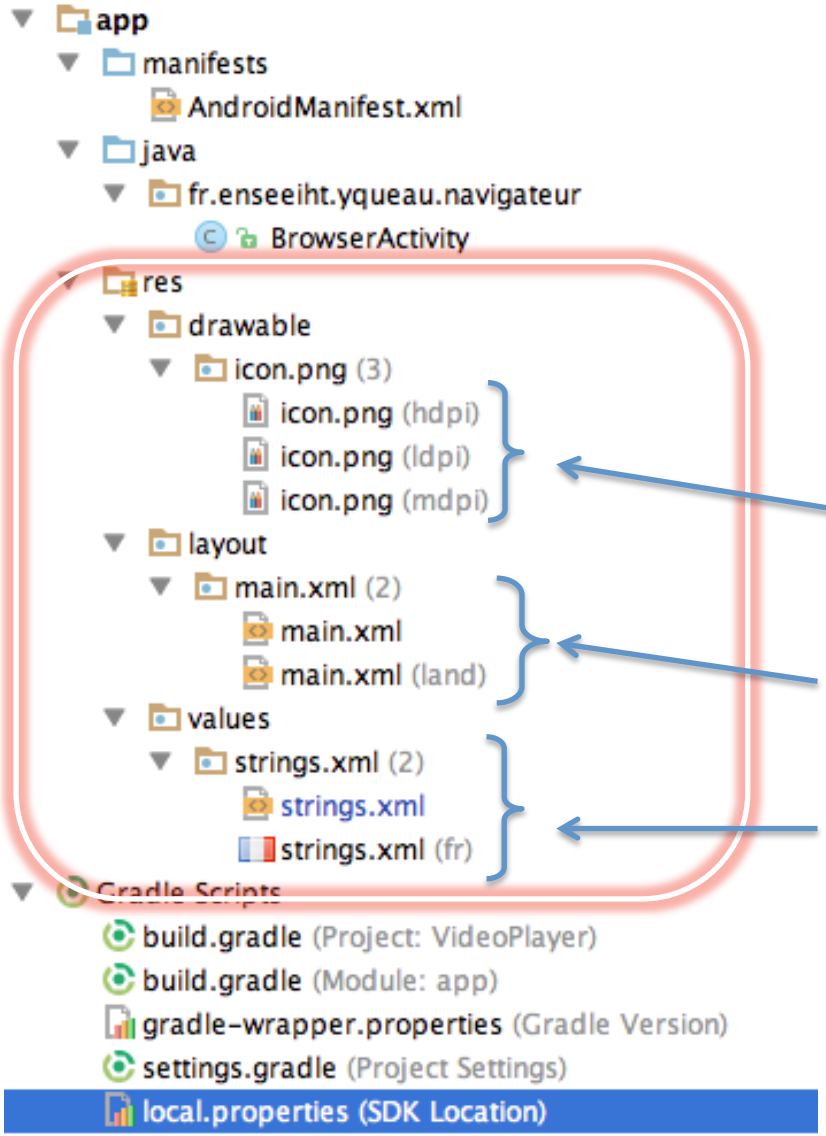
Android applications - Structure



Source code

- **java** folder
- Organize like any Java application
- Package(s)
- `.java` with sources

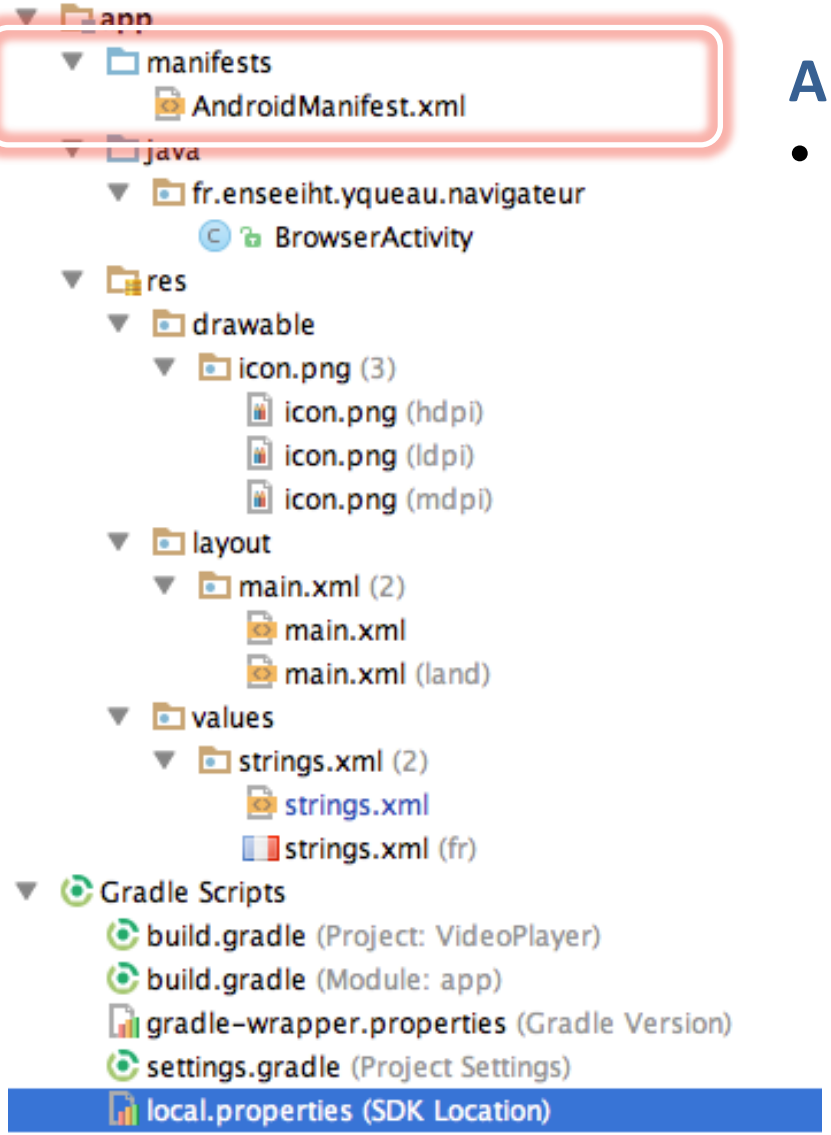
Android applications - Structure



Resource files

- All the GUI related files
  - Icons, text, layout GUI
- Support for multiple screen densities
- Support for multiple layout orientations
- Support for localisation

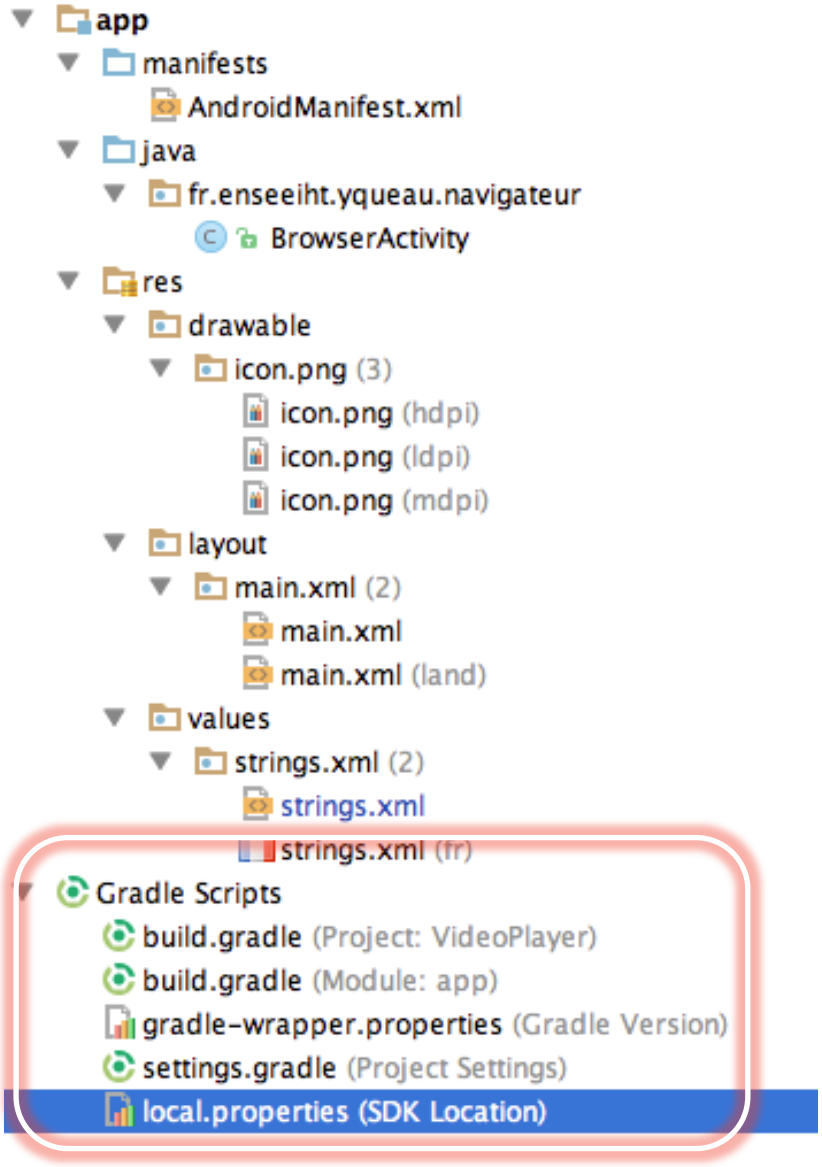
Android applications - Structure



Android Manifest File

- Contains information about the app
  - The name
  - The components (activities, services, broadcast receivers…)
  - Permissions needed to operate (access to sensors, to sdcard, to internet…)

Android applications - Structure



Gradle scripts

- Scripts and configuration file for the building system (Gradle)
- Dependencies
- Targeted version to build

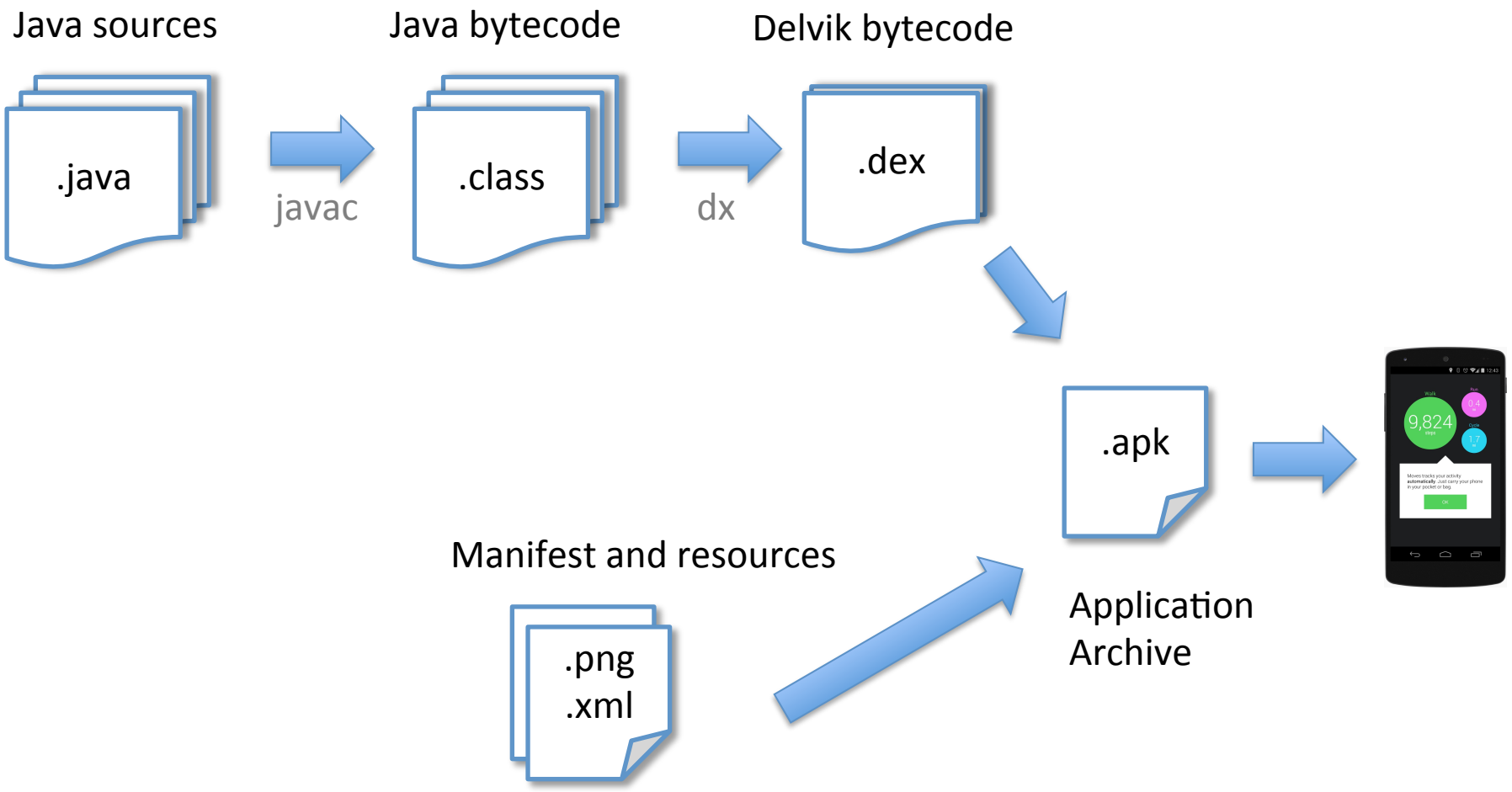
Plan

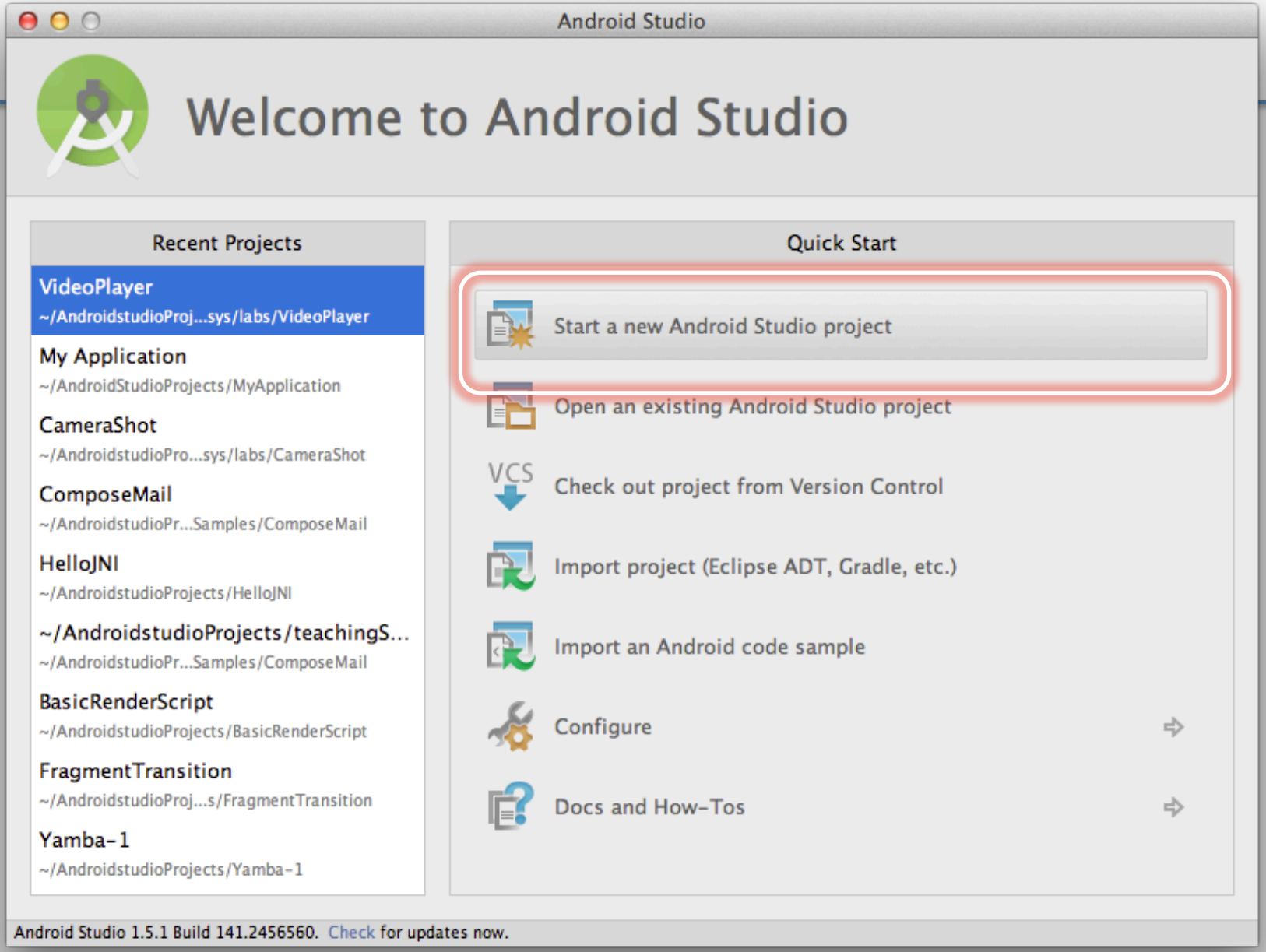
- Mobile devices
- Android overview
  - History
  - Architecture
  - Applications
- The tools
- **Android Applications**

Hello World! Application

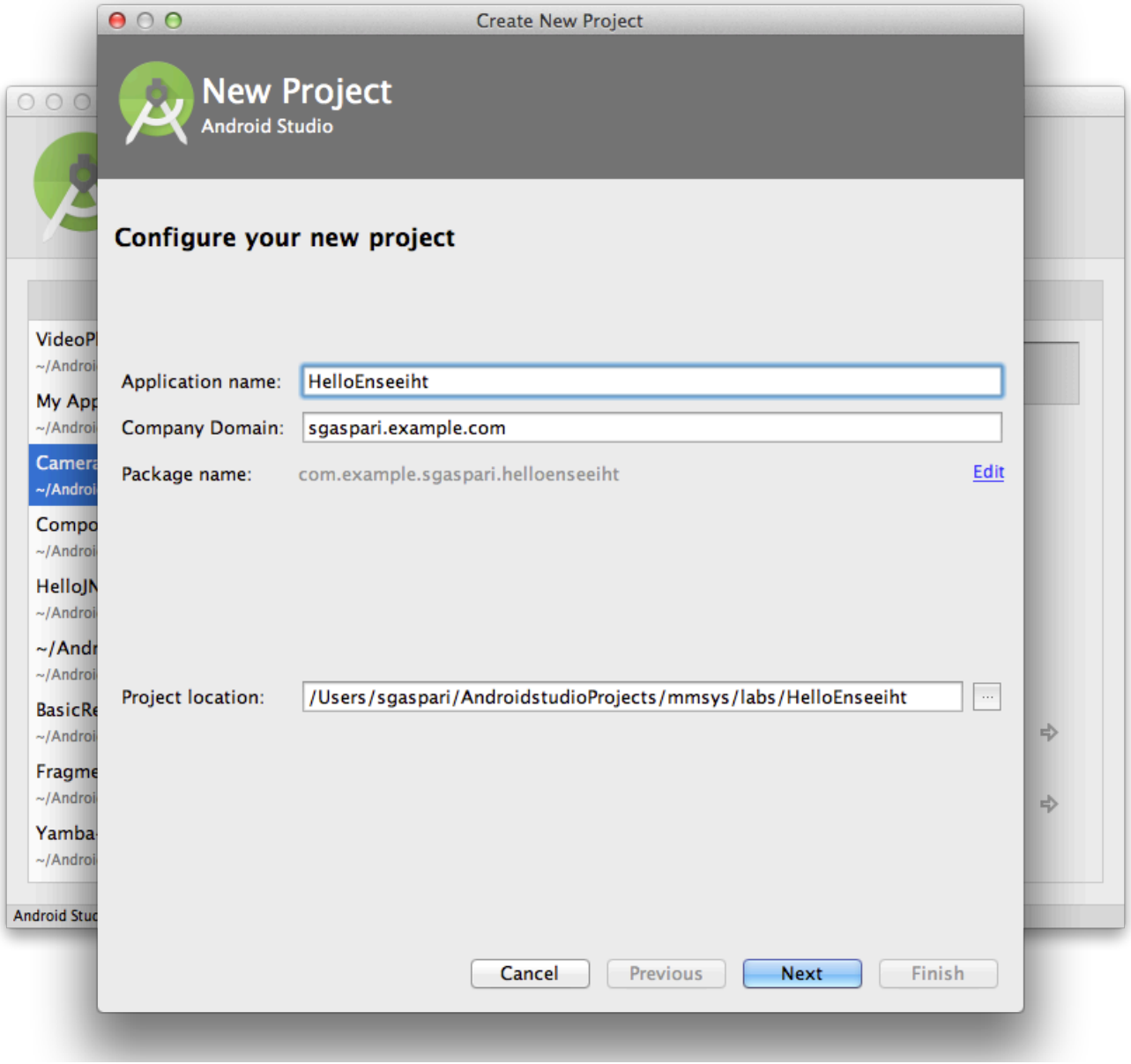
- Demo
- Activity class
  - Lifecycle
- The GUI
- Manifest file

Applications

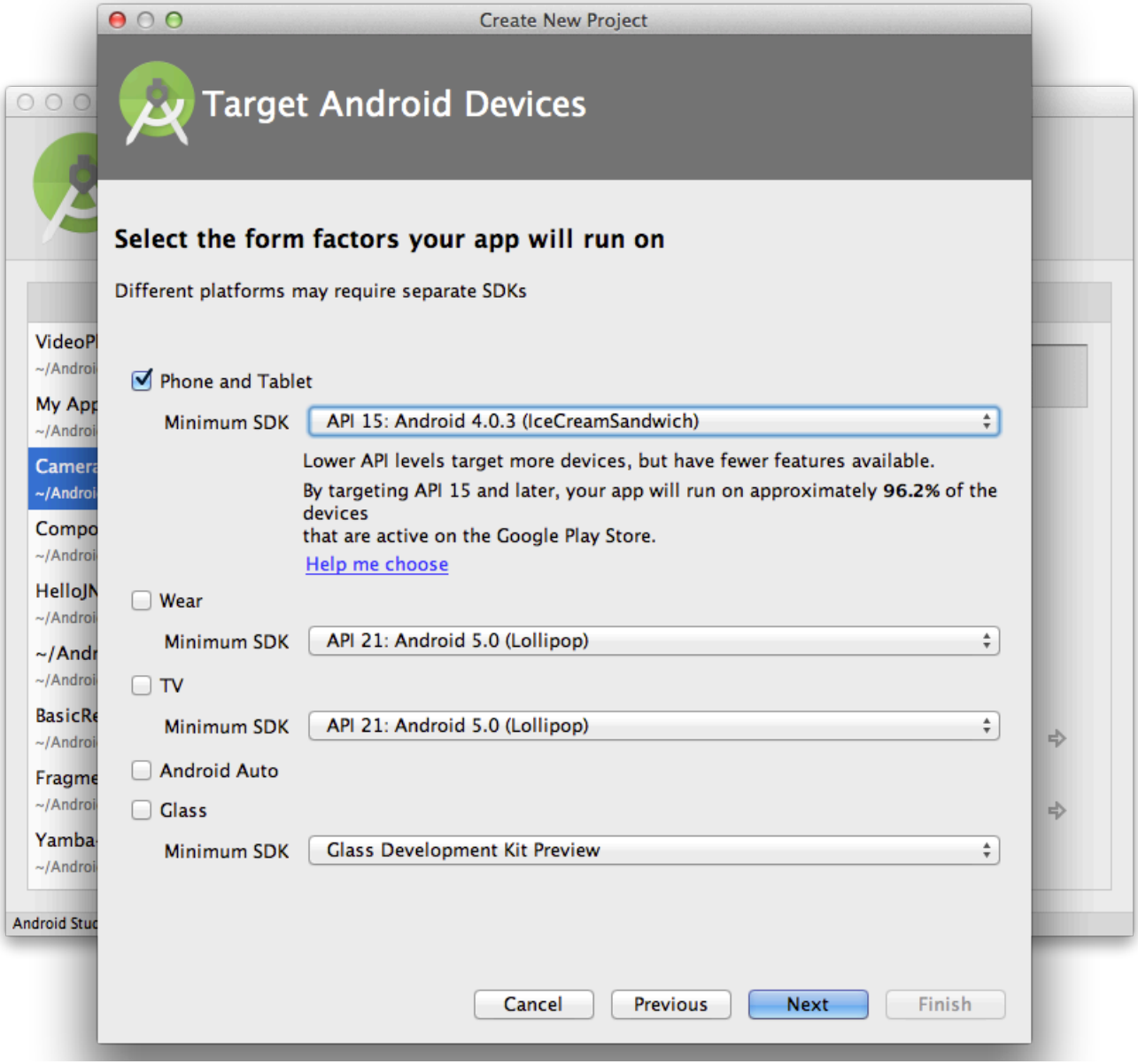



Android Studio
Welcome to Android Studio
Recent Projects
VideoPlayer
~/AndroidstudioProj...sys/labs/VideoPlayer
My Application
~/AndroidStudioProjects/MyApplication
CameraShot
~/AndroidstudioPro...sys/labs/CameraShot
ComposeMail
~/AndroidstudioPr...Samples/ComposeMail
HelloJNI
~/AndroidstudioProjects/HelloJNI
~/AndroidstudioProjects/teachingS...
~/AndroidstudioPr...Samples/ComposeMail
BasicRenderScript
~/AndroidstudioProjects/BasicRenderScript
FragmentTransition
~/AndroidstudioProj...s/FragmentTransition
Yamba-1
~/AndroidstudioProjects/Yamba-1
Quick Start
Start a new Android Studio project
Open an existing Android Studio project
VCS
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Docs and How-Tos
Android Studio 1.5.1 Build 141.2456560. Check for updates now.

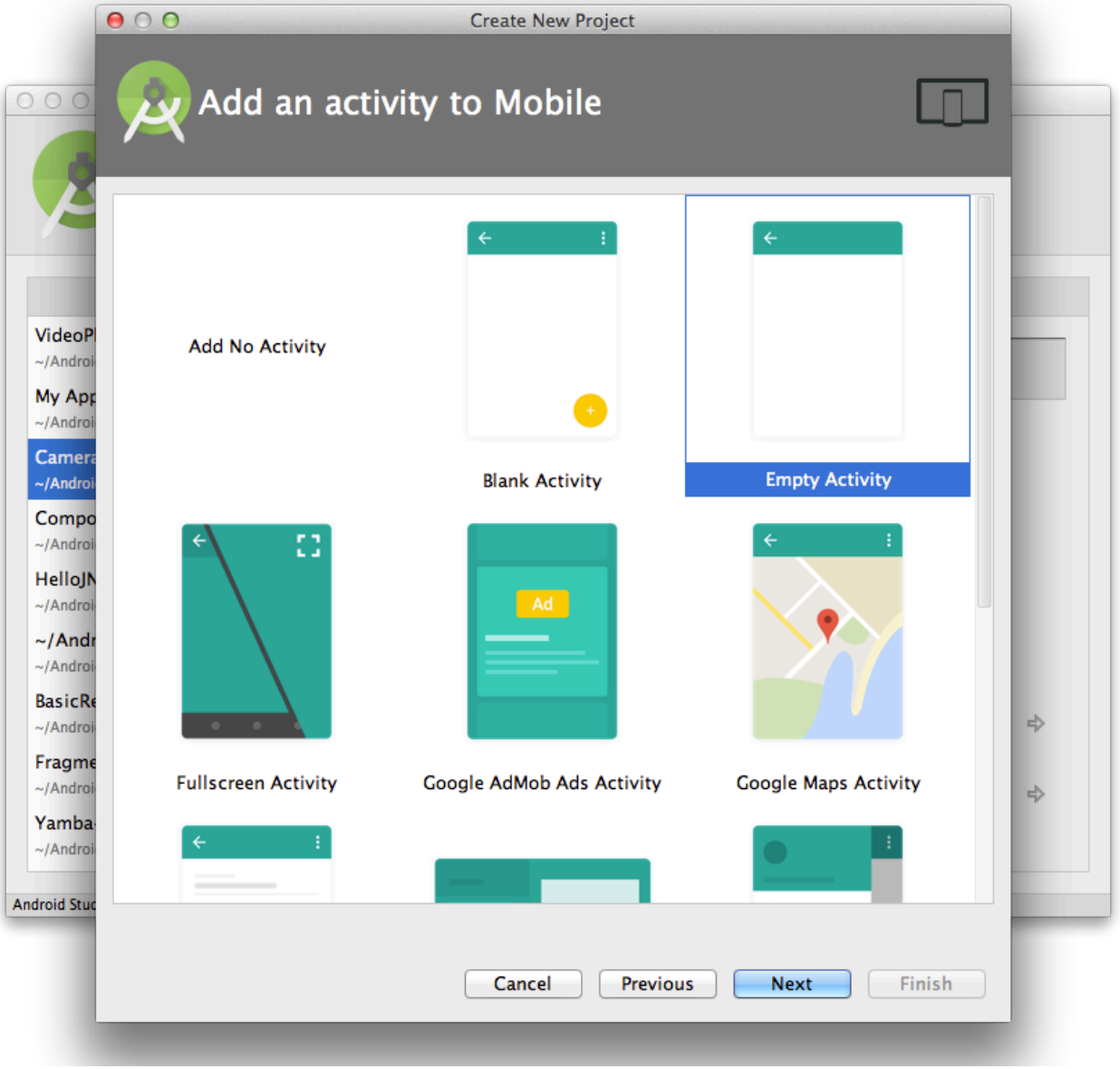
Hello World!


Create New Project
New Project
Android Studio
Configure your new project
Application name:
HelloEnseeiht
Company Domain:
sgaspari.example.com
Package name:
com.example.sgaspari.helloenseeiht
Edit
Project location:
/Users/sgaspari/AndroidstudioProjects/mmsys/labs/HelloEnseeiht
Cancel
Previous
Next
Finish

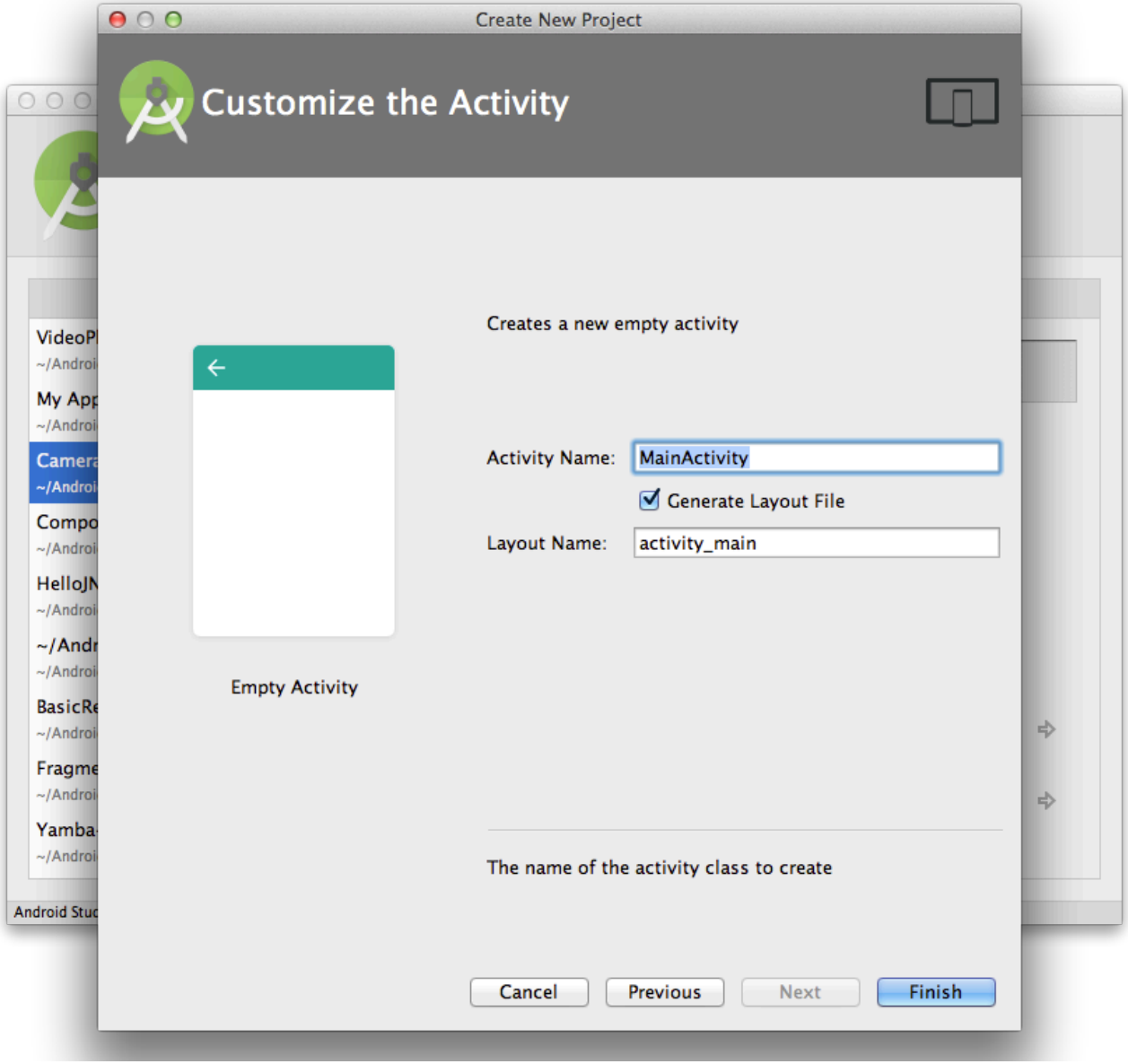
Hello World!


Create New Project
Target Android Devices
Select the form factors your app will run on
Different platforms may require separate SDKs
Phone and Tablet
Minimum SDK
API 15: Android 4.0.3 (IceCreamSandwich)
Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately 96.2% of the devices
that are active on the Google Play Store.
Help me choose
Wear
Minimum SDK
API 21: Android 5.0 (Lollipop)
TV
Minimum SDK
API 21: Android 5.0 (Lollipop)
Android Auto
Glass
Minimum SDK
Glass Development Kit Preview
Cancel
Previous
Next
Finish

Hello World!


Create New Project
Add an activity to Mobile
Add No Activity
Blank Activity
Empty Activity
Fullscreen Activity
Google AdMob Ads Activity
Google Maps Activity
Cancel
Previous
Next
Finish

Hello World!


Create New Project
Customize the Activity
Creates a new empty activity
Activity Name:
MainActivity
Generate Layout File
Layout Name:
activity_main
Empty Activity
The name of the activity class to create
Cancel
Previous
Next
Finish

Hello World!

```
package com.example.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity
{
    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
```

Hello World!

Hello World!

```
public class MainActivity extends Activity
{
    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );
    }
```

Hello World! Application

- Demo
- **Activity class**
  - **Lifecycle**
- The GUI
- Manifest file

Activity lifecycle

3 main states for an activity according to the current Task

- **Resumed**
  - foreground and ready to interact ("running")

http://developer.android.com/guide/components/activities.html#Lifecycle

Activity lifecycle

3 main states for an activity according to the current Task

- **Resumed**
  - foreground and ready to interact ("running")
- **Paused**
  - Visible but with another activity in foreground
    - i.e. under a dialog box or another activity is about to taking focus
  - It can be killed by the system in extremely low memory situations.

http://developer.android.com/guide/components/activities.html#Lifecycle

Activity lifecycle

3 main states for an activity according to the current Task

- **Resumed**
  - foreground and ready to interact ("running")
- **Paused**
  - Visible but with another activity in foreground
    - i.e. under a dialog box or another activity is about to taking focus
  - It can be killed by the system in extremely low memory situations.
- **Stopped**
  - Not visible, another activity is in foreground
  - It is still in memory but…
  - …it can be killed by the system when memory is needed elsewhere.

http://developer.android.com/guide/components/activities.html#Lifecycle

Activity lifecycle

Activity class provides callbacks for each state transition

- The callbacks can be overridden to perform custom operations

http://developer.android.com/guide/components/activities.html#Lifecycle

Activity lifecycle

Activity class provides callbacks for each state transition

- The callbacks can be overridden to perform custom operations

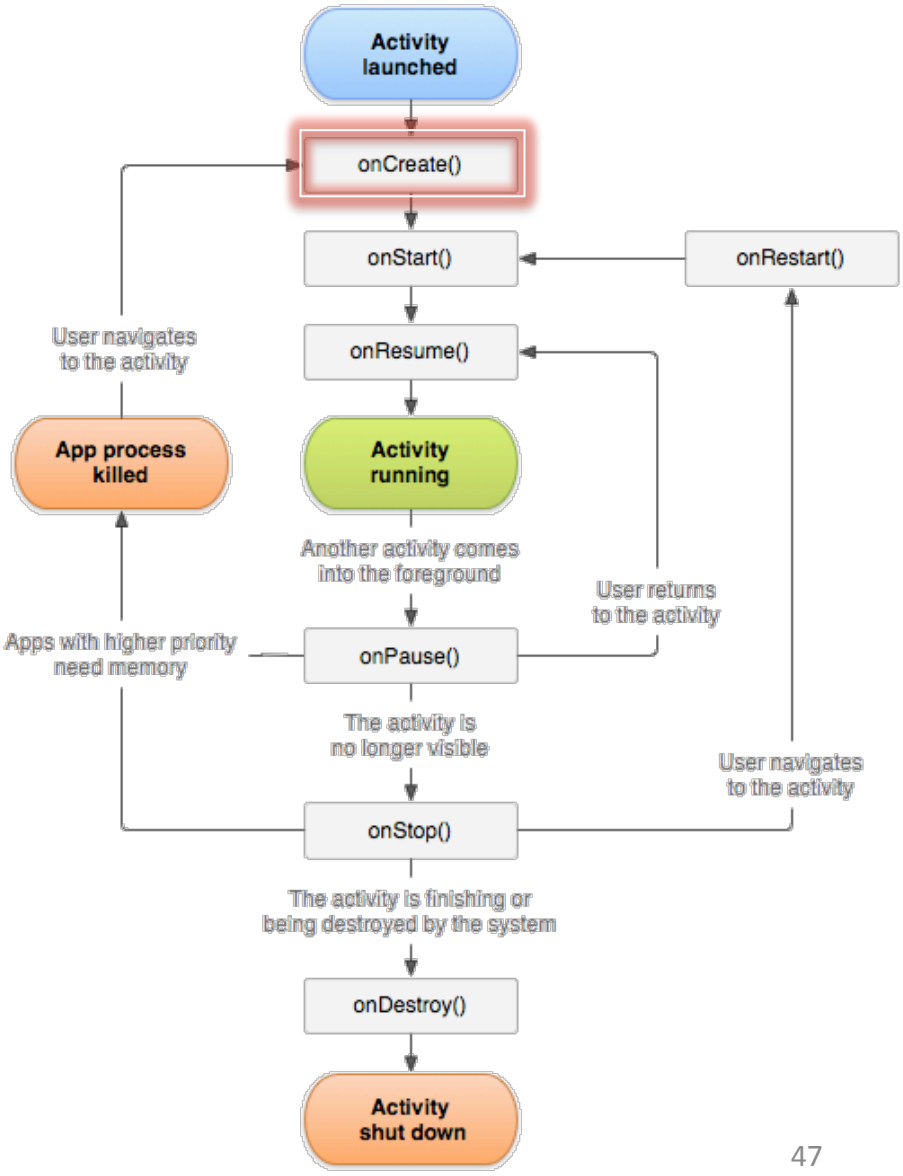
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

http://developer.android.com/guide/components/activities.html#Lifecycle

Activity Lifecycle callback methods

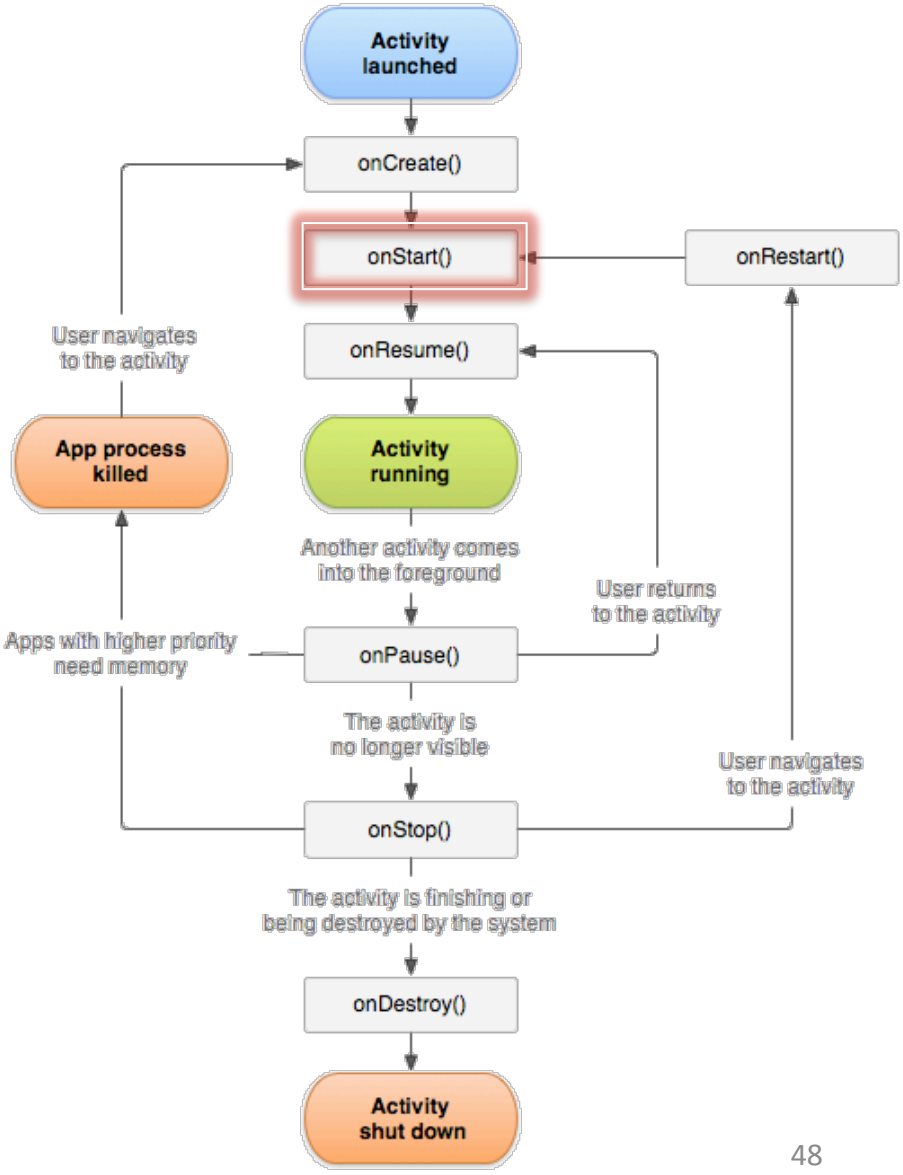
- **`onCreate()`** – called to initialize an Activity when it is first created

  - Startup operations
  - Ex. Load the GUI



Activity Lifecycle callback methods

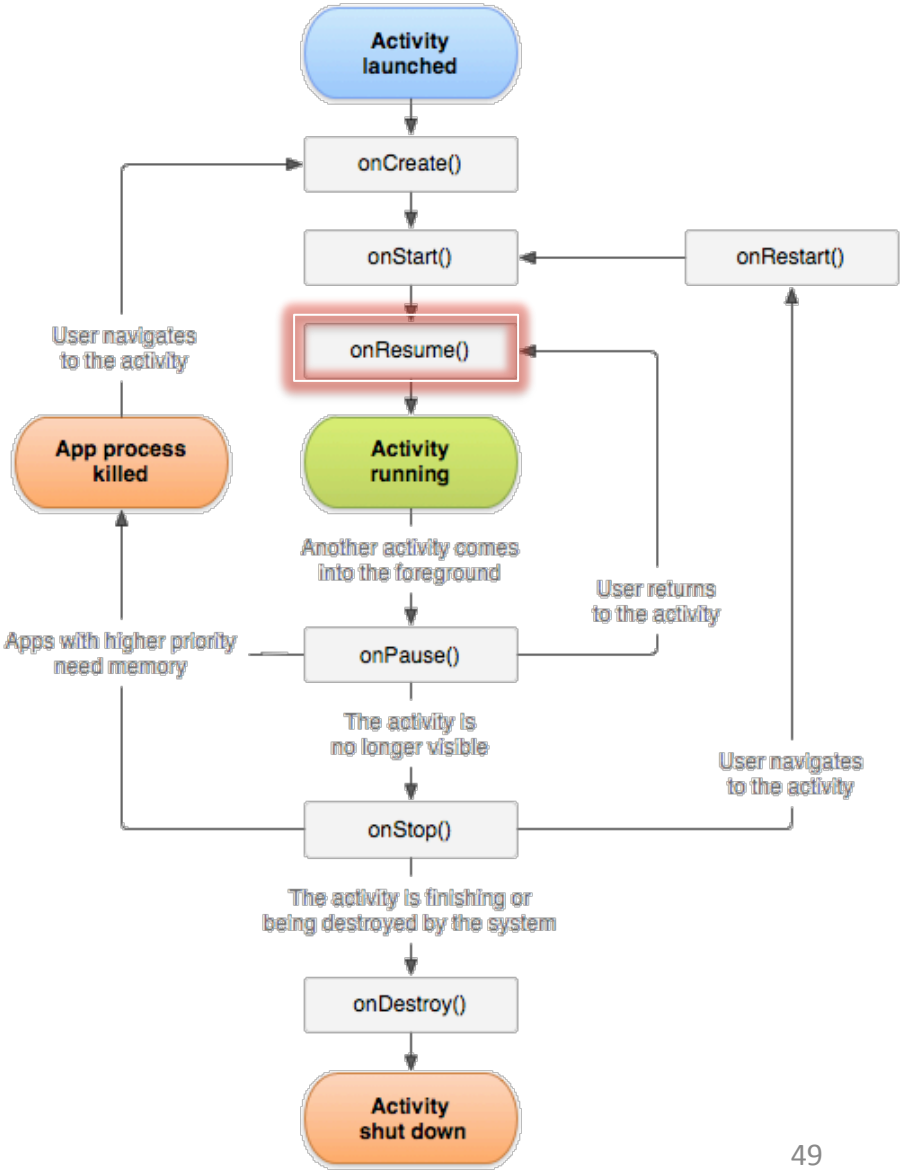
- `onCreate()` – called to initialize an Activity when it is first created
- `onStart()` – called when Activity is becoming visible to the user



Activity Lifecycle callback methods

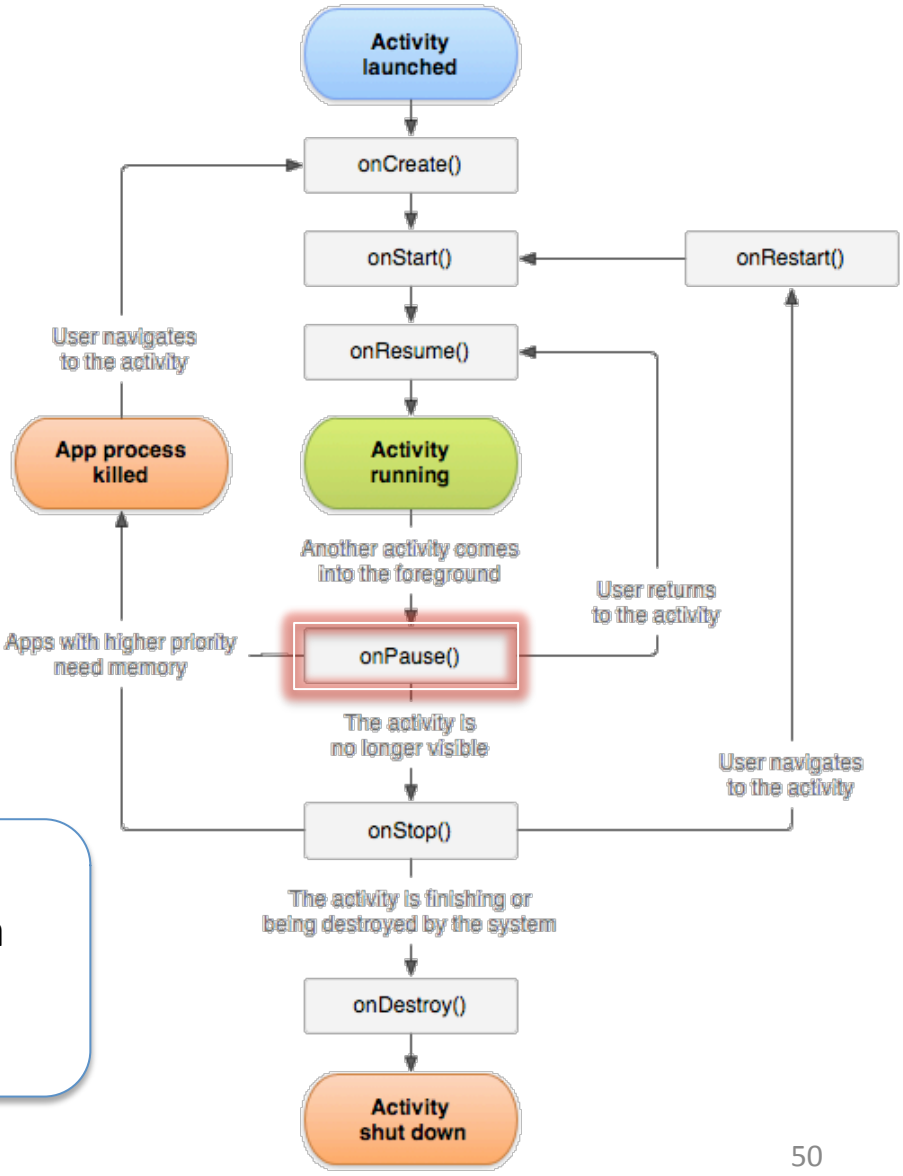
- `onCreate()` – called to initialize an Activity when it is first created
- `onStart()` – called when Activity is becoming visible to the user
- **`onResume()`** – called when user returns to an Activity from another

initialize components like sensors, camera etc



Activity Lifecycle callback methods

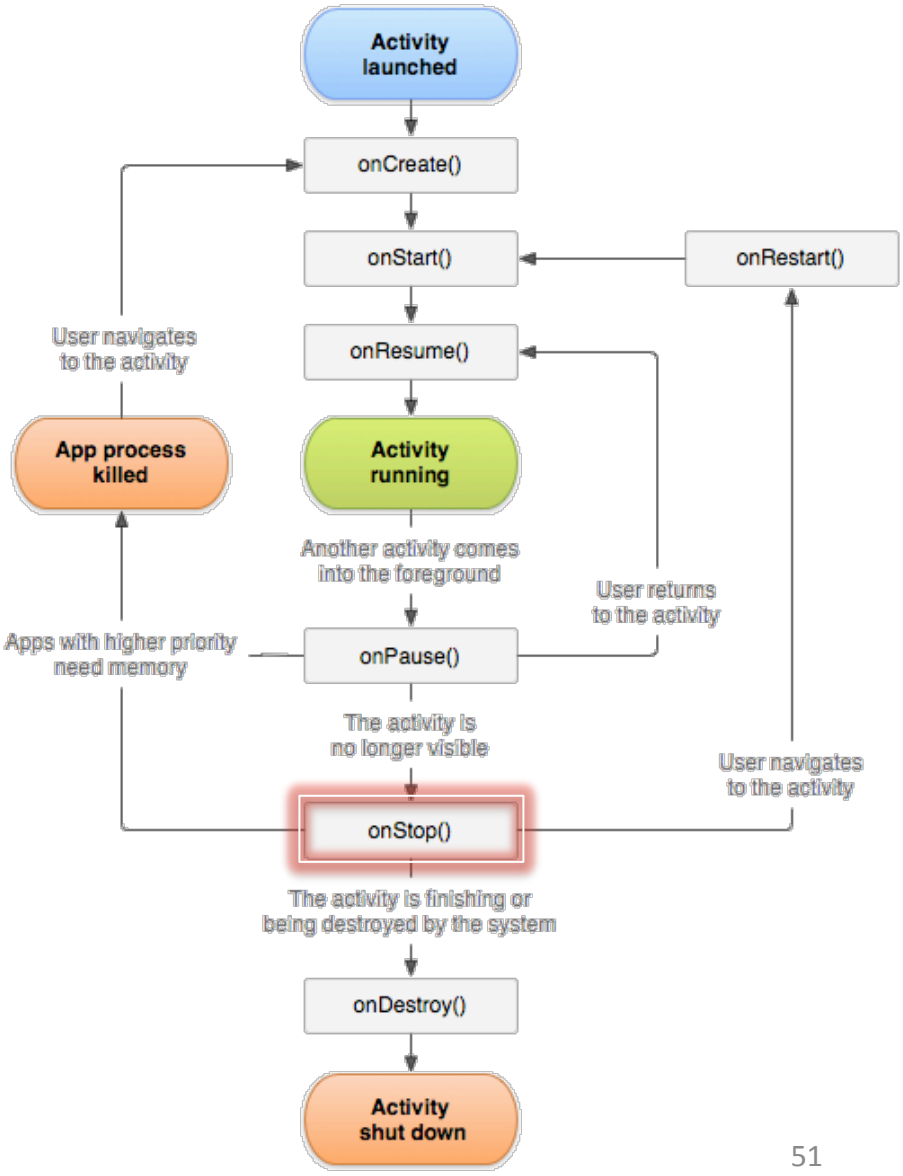
- `onCreate()` – called to initialize an Activity when it is first created
- `onStart()` – called when Activity is becoming visible to the user
- `onResume()` – called when user returns to an Activity from another
- **`onPause()`** – called when user leaves an Activity that's still visible in background



- Dismiss all the resources that consume computational power (sensors, camera etc)
- Save data if needed

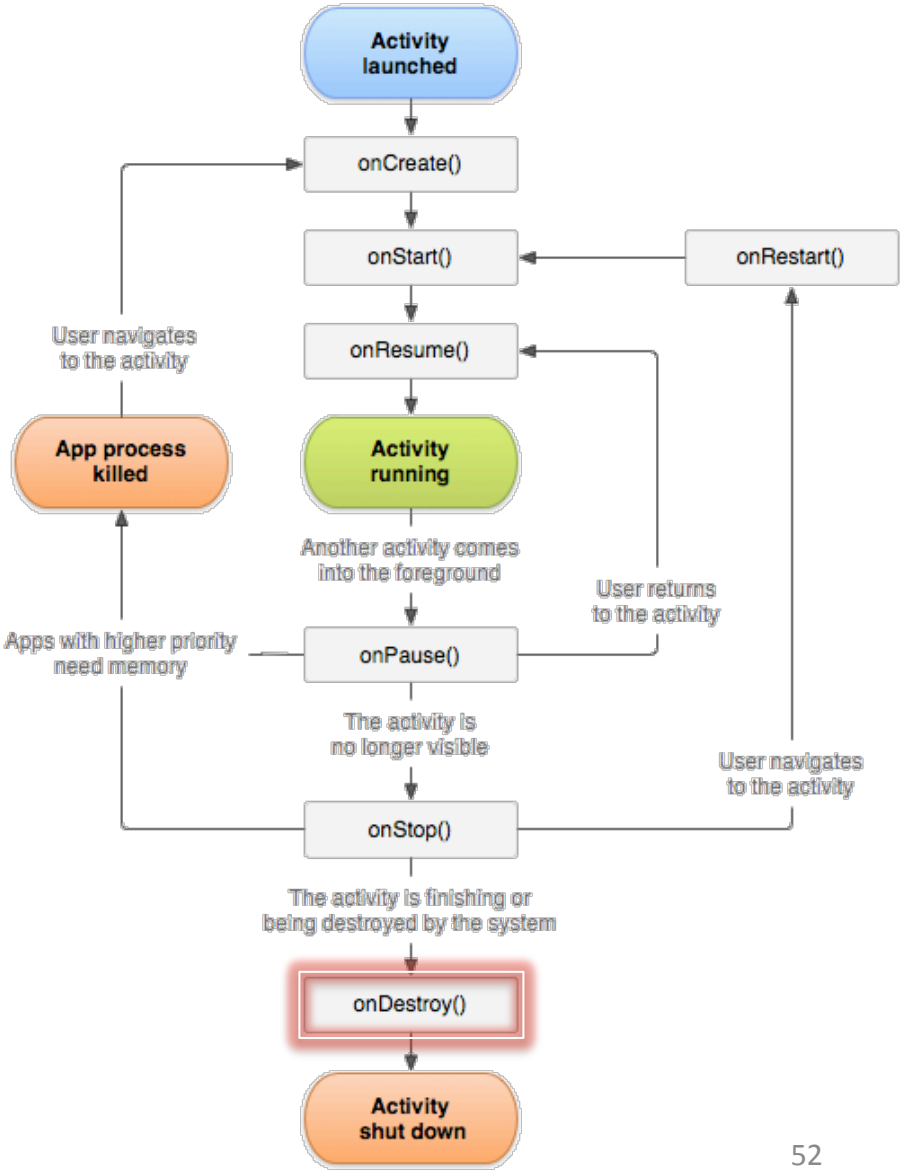
Activity Lifecycle callback methods

- **`onCreate()`** – called to initialize an Activity when it is first created
- **`onStart()`** – called when Activity is becoming visible to the user
- **`onResume()`** – called when user returns to an Activity from another
- **`onPause()`** – called when user leaves an Activity that's still visible in background
- **`onStop()`** – called when user leaves an Activity for another

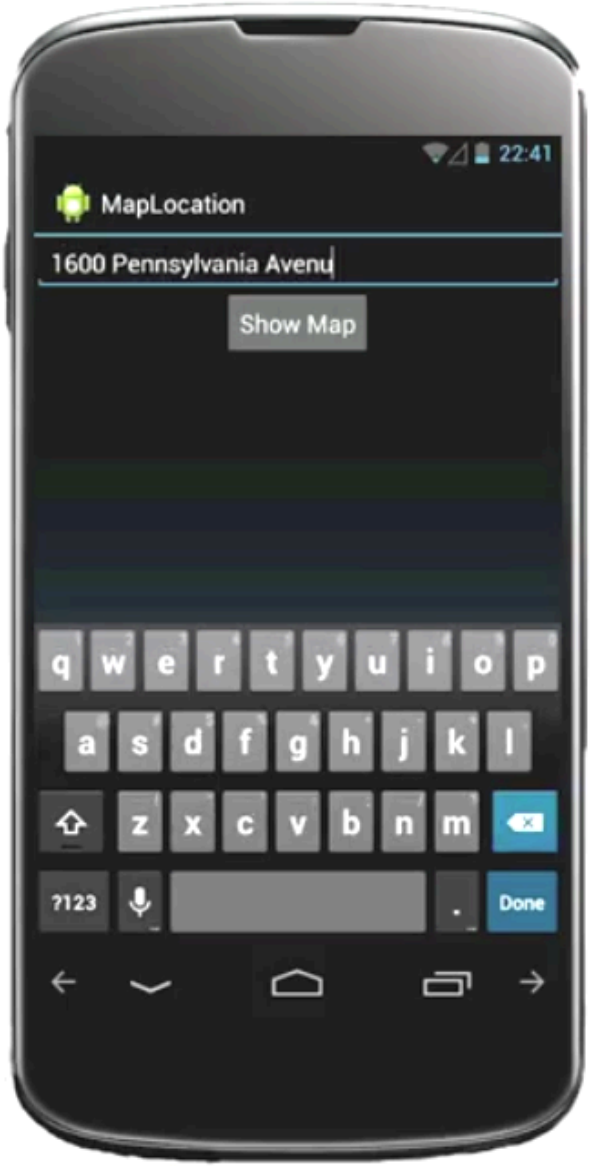


Activity Lifecycle callback methods

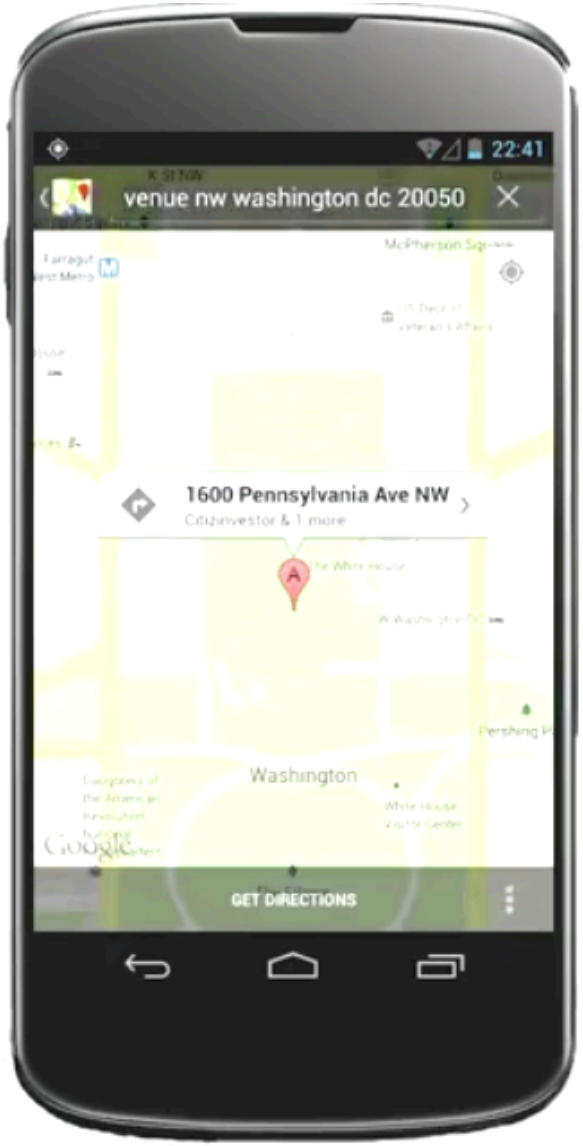
- **`onCreate()`** – called to initialize an Activity when it is first created
- **`onStart()`** – called when Activity is becoming visible to the user
- **`onResume()`** – called when user returns to an Activity from another
- **`onPause()`** – called when user leaves an Activity that's still visible in background
- **`onStop()`** – called when user leaves an Activity for another
- **`onDestroy()`** – called when Activity is being released & needs to clean up its allocated resources



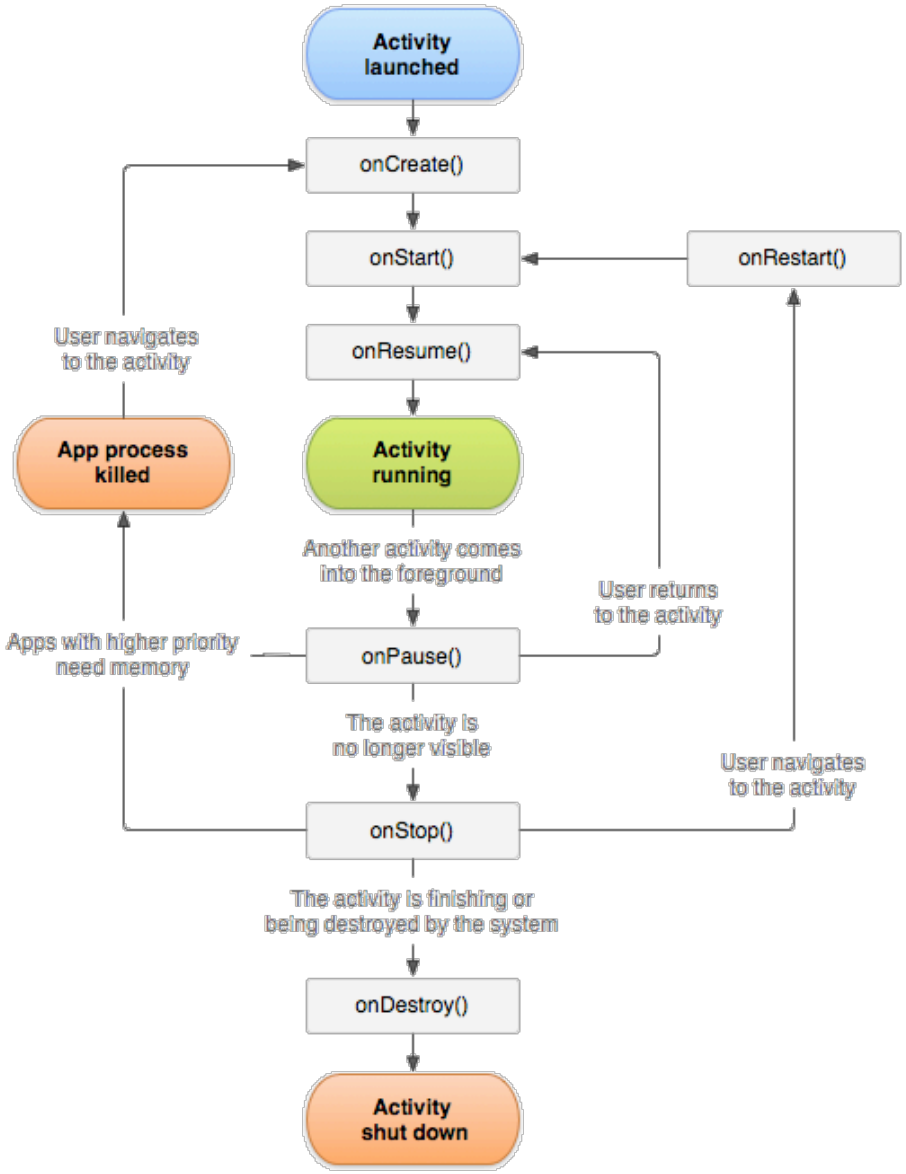
Example



Main application launch an **intent** to show a location on a map

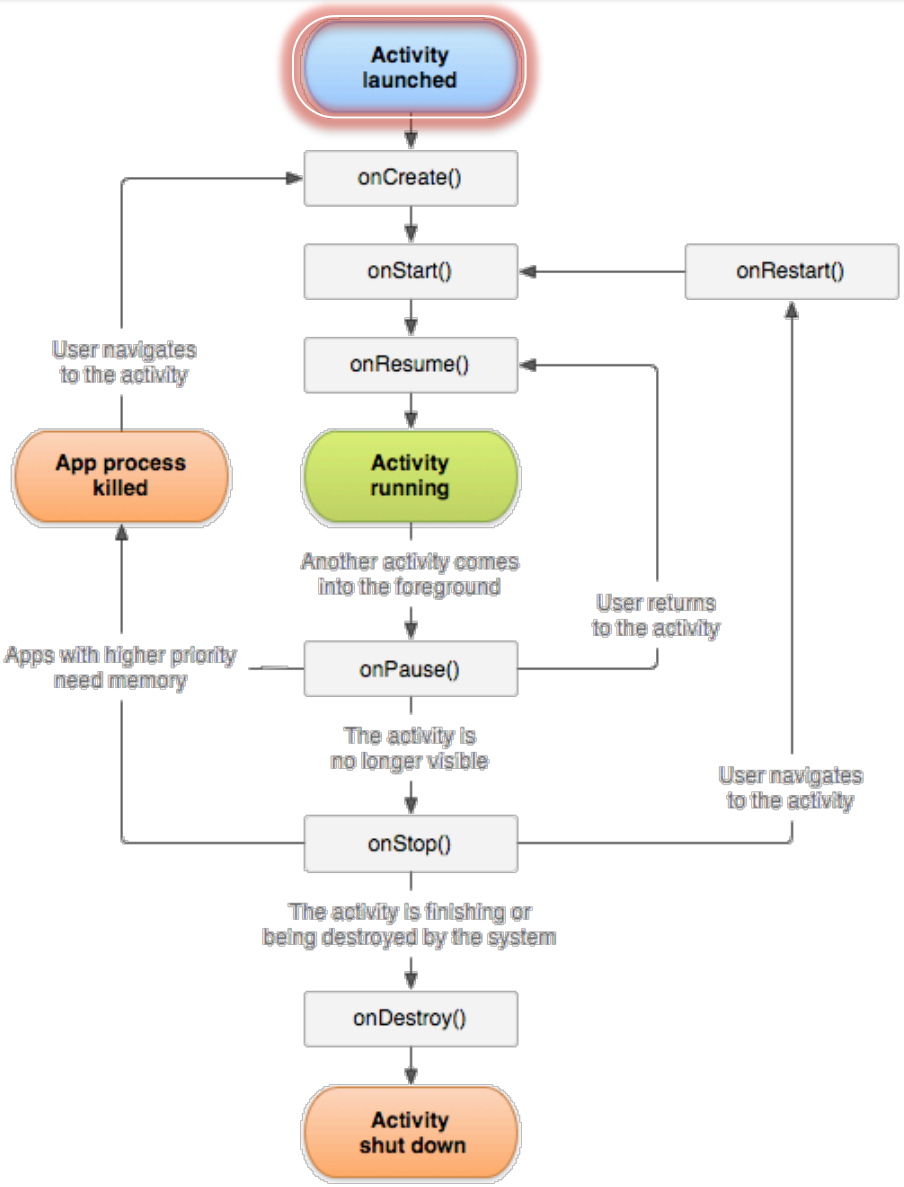


Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

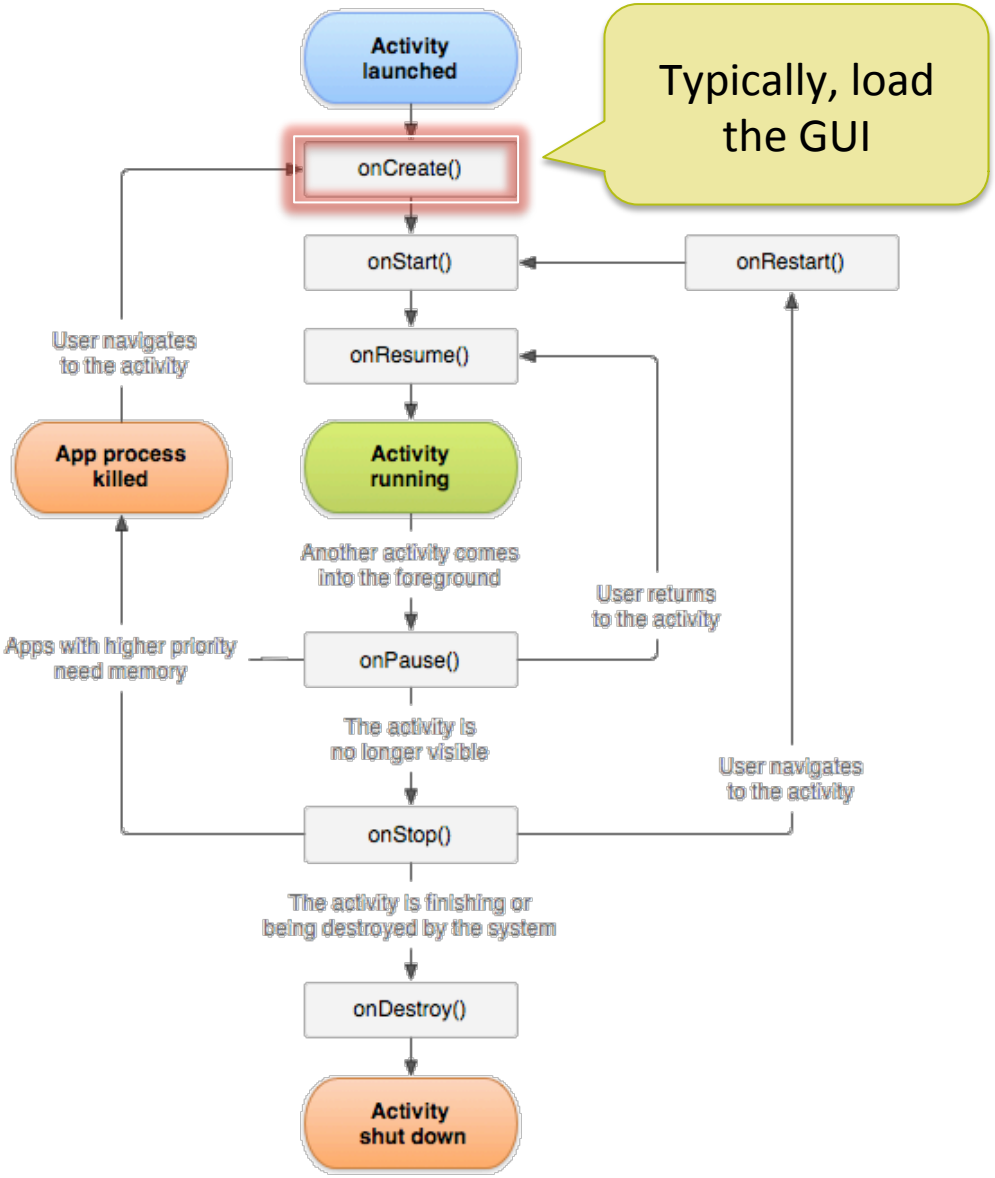

22:41
APPS
WIDGETS
AirDroid
Amazon Kindle
Browser
Calculator
Calendar
Camera
Chrome
Clock
Downloads
Email
ES File Explorer
Gallery
Gin Rummy Free
Gmail
Google
Google Settings
Google+
Jota Text Editor
Local
Map Location
MapLocation
Maps
Messaging
Messenger
MoboPlayer

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

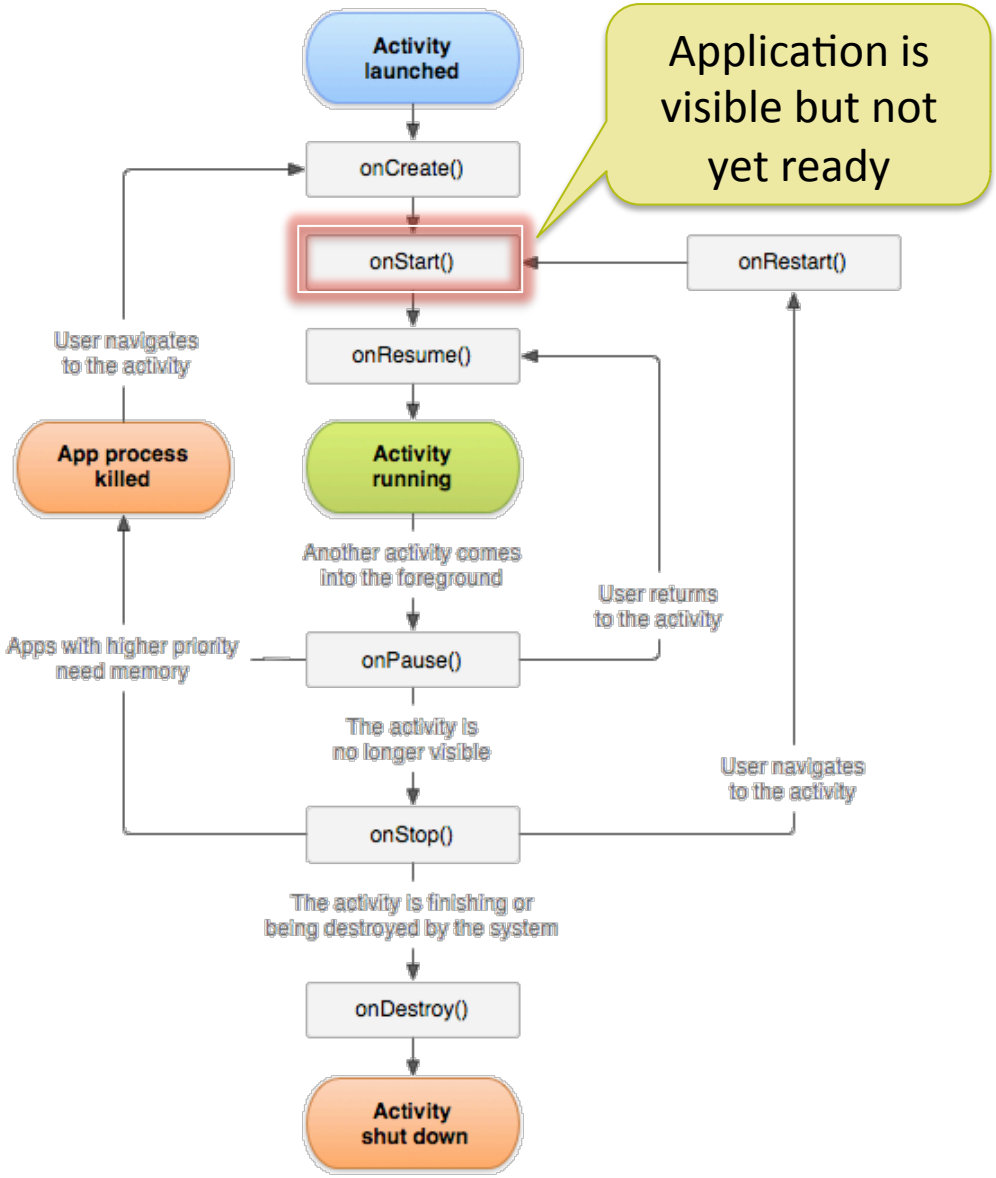

22:41
APPS
WIDGETS
AirDroid
Amazon Kindle
Browser
Calculator
Calendar
Camera
Chrome
Clock
Downloads
Email
ES File Explorer
Gallery
Gin Rummy Free
Gmail
Google
Google Settings
Google+
Jota Text Editor
Local
Map Location
Maps
Messaging
Messenger
MoboPlayer

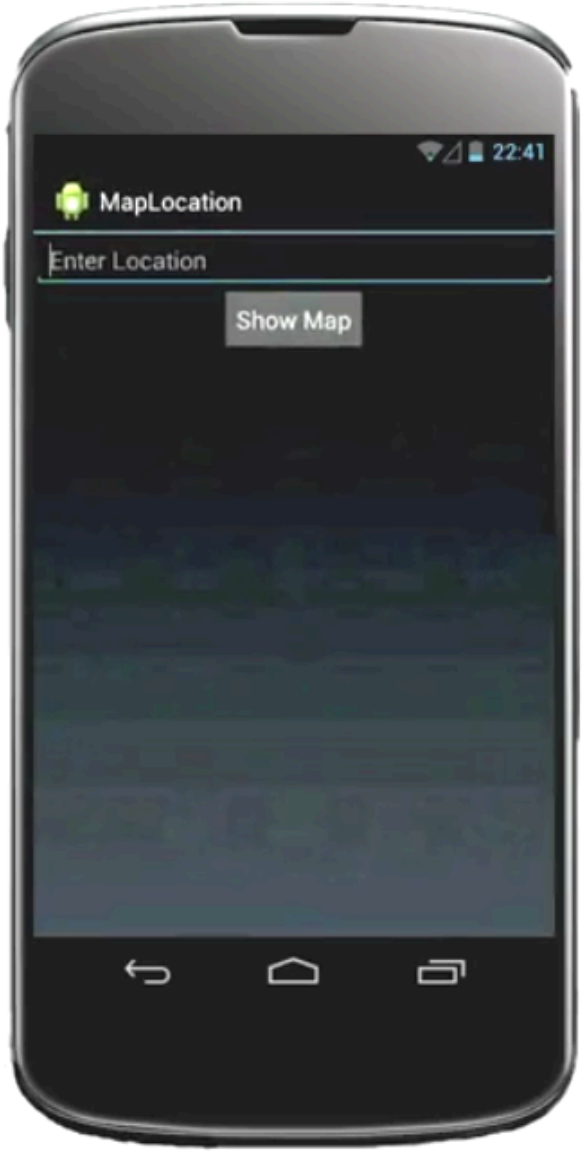
Example


Activity launched
onCreate()
Typically, load the GUI
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

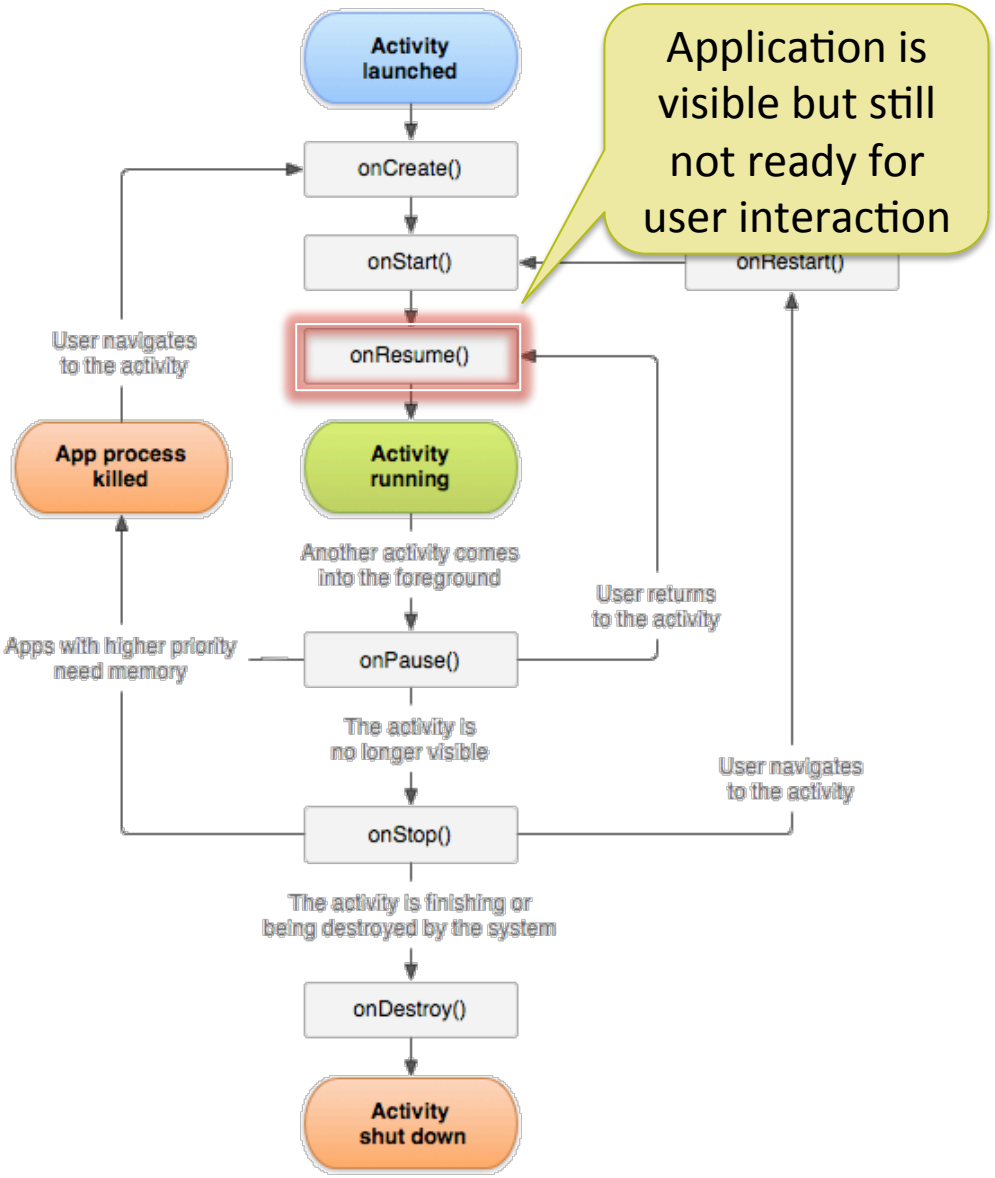

22:41
APPS
WIDGETS
AirDroid
Amazon Kindle
Browser
Calculator
Calendar
MapLocation
Enter Location
Camera
Chrome
Clock
Downloads
Email
Show Map
ES File Explorer
Gallery
Gin Rummy Free
Gmail
Google
Google Settings
Google+
Jota Text Editor
Local
Map Location
MapLocation
Maps
Messaging
Messenger
MoboPlayer

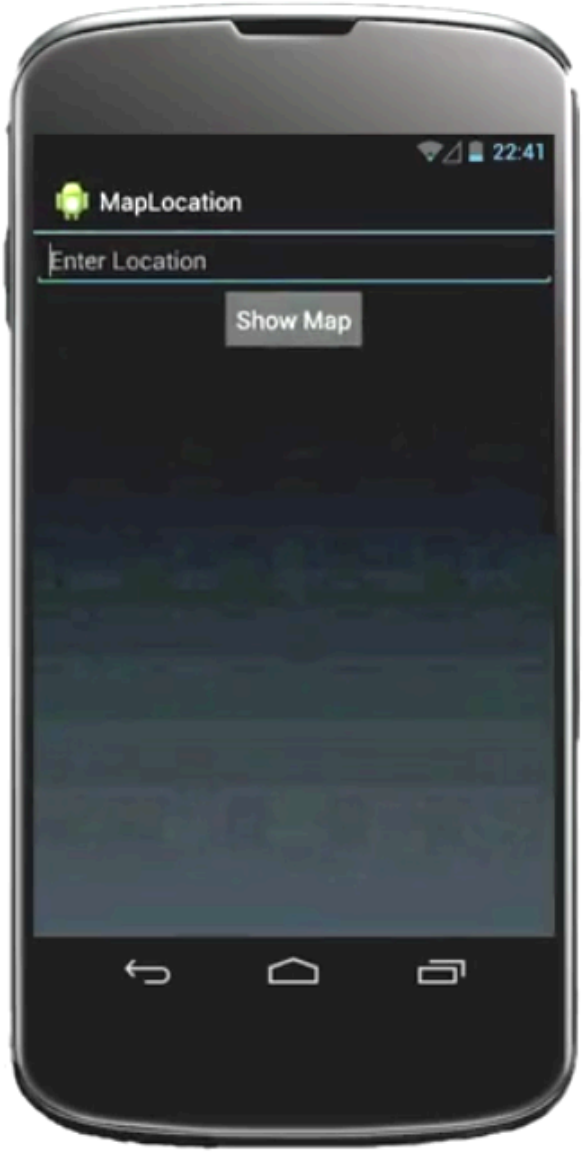
Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down
Application is visible but not yet ready

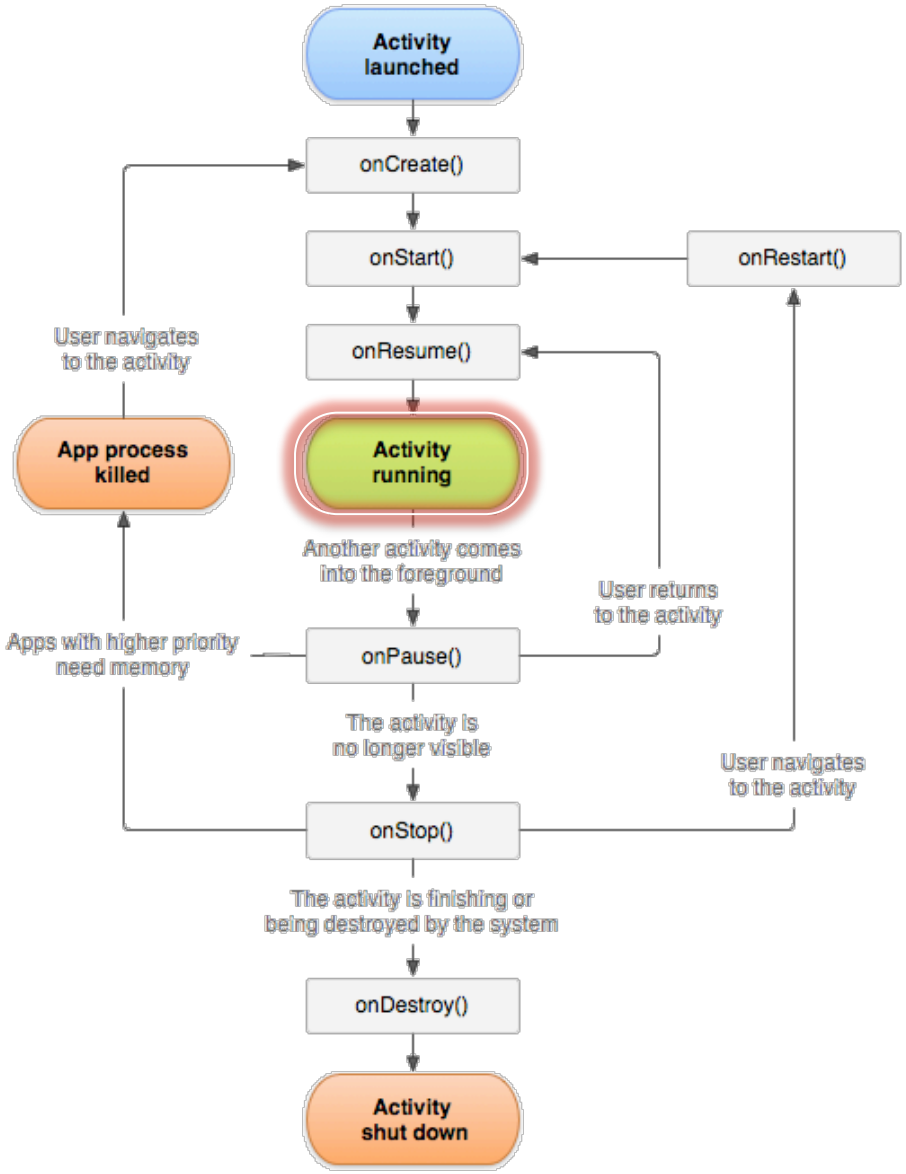

22:41
MapLocation
Enter Location
Show Map

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down
Application is visible but still not ready for user interaction

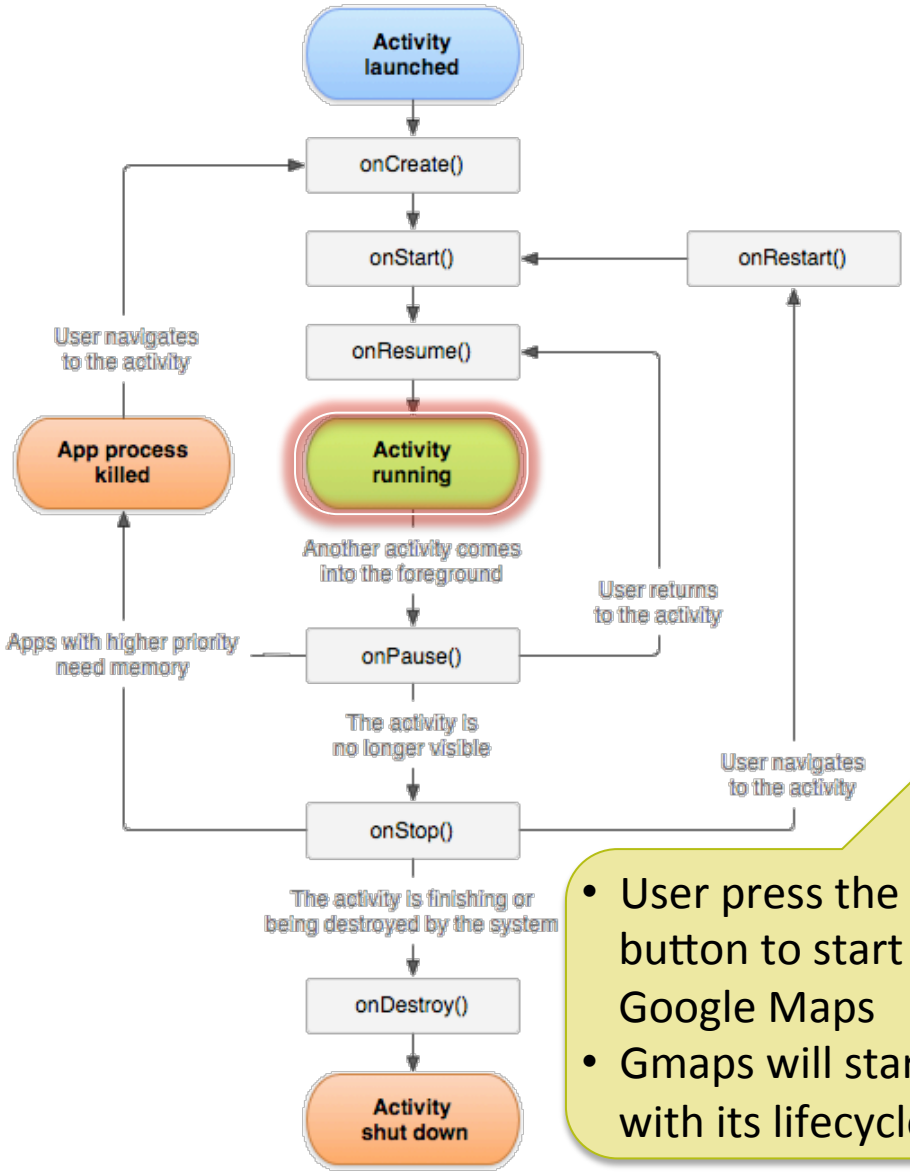

22:41
MapLocation
Enter Location
Show Map

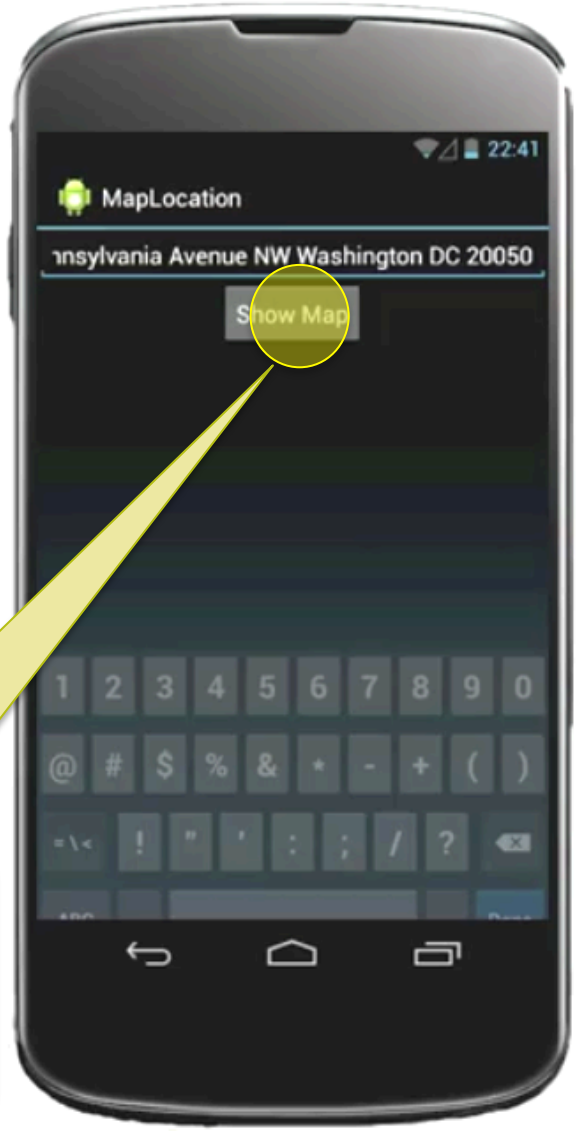
Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down


22:41
MapLocation
1600 Pennsylvania Avenu
Show Map
Done

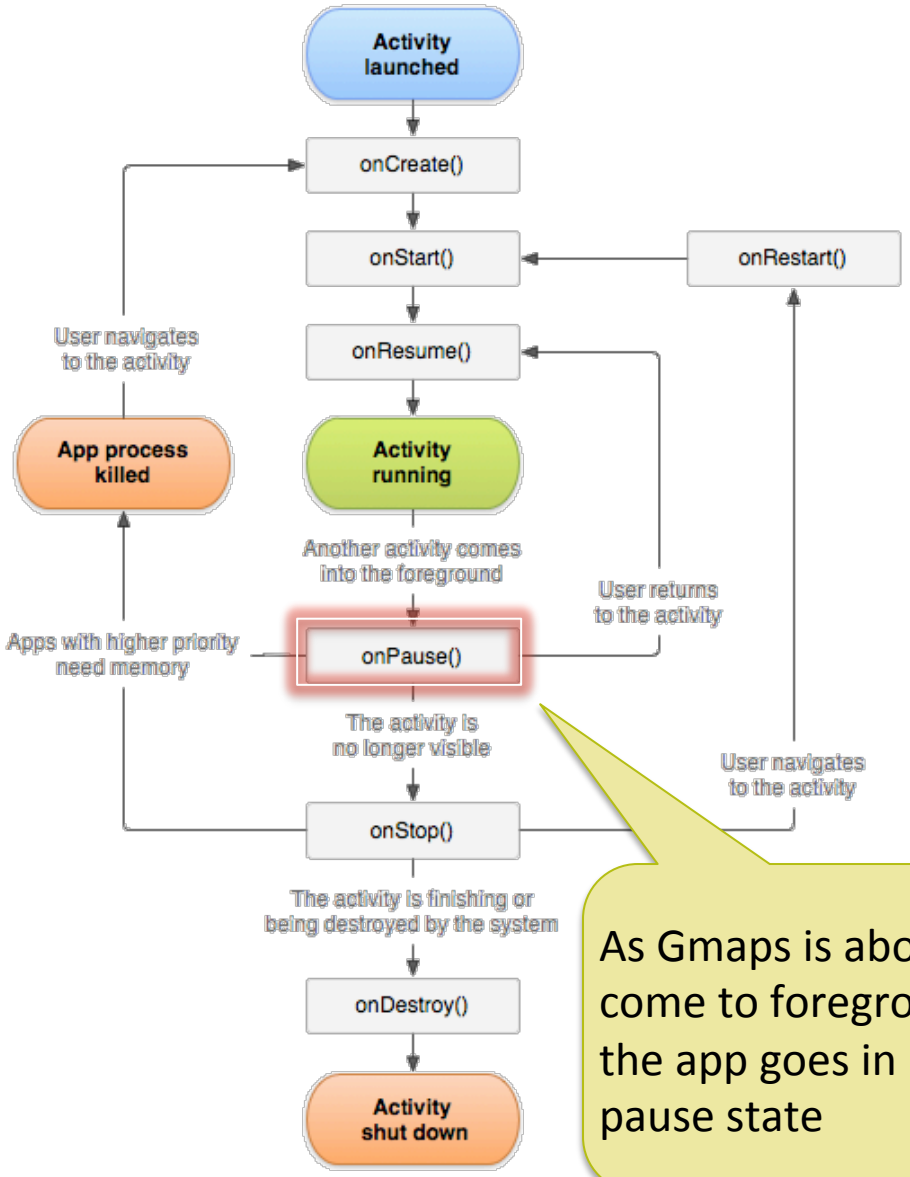
Example



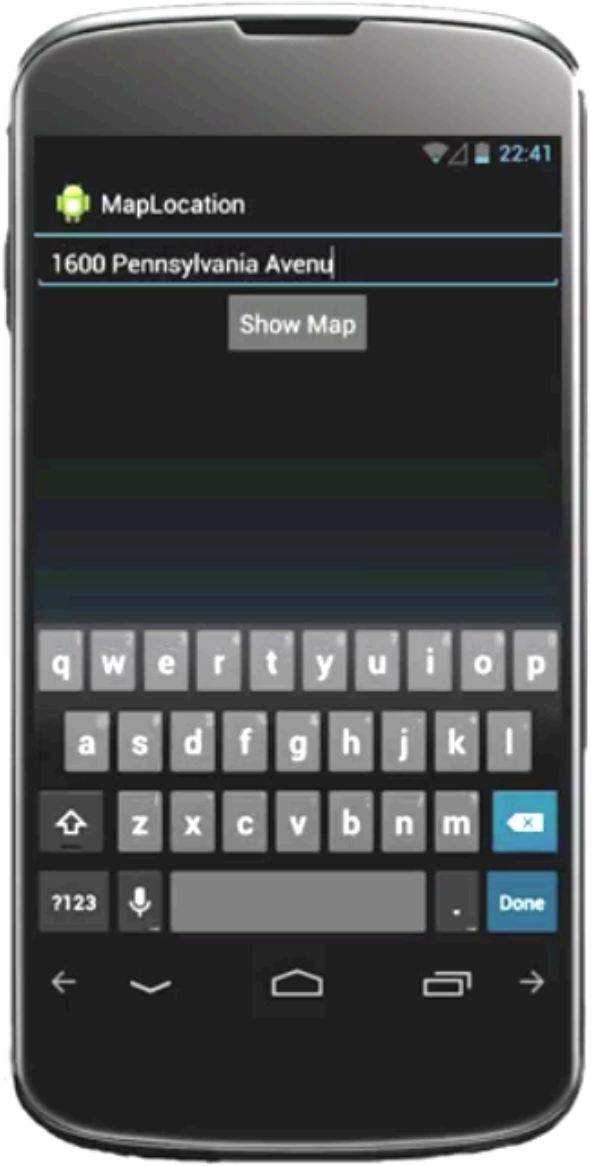


- User press the button to start Google Maps
- Gmaps will start with its lifecycle

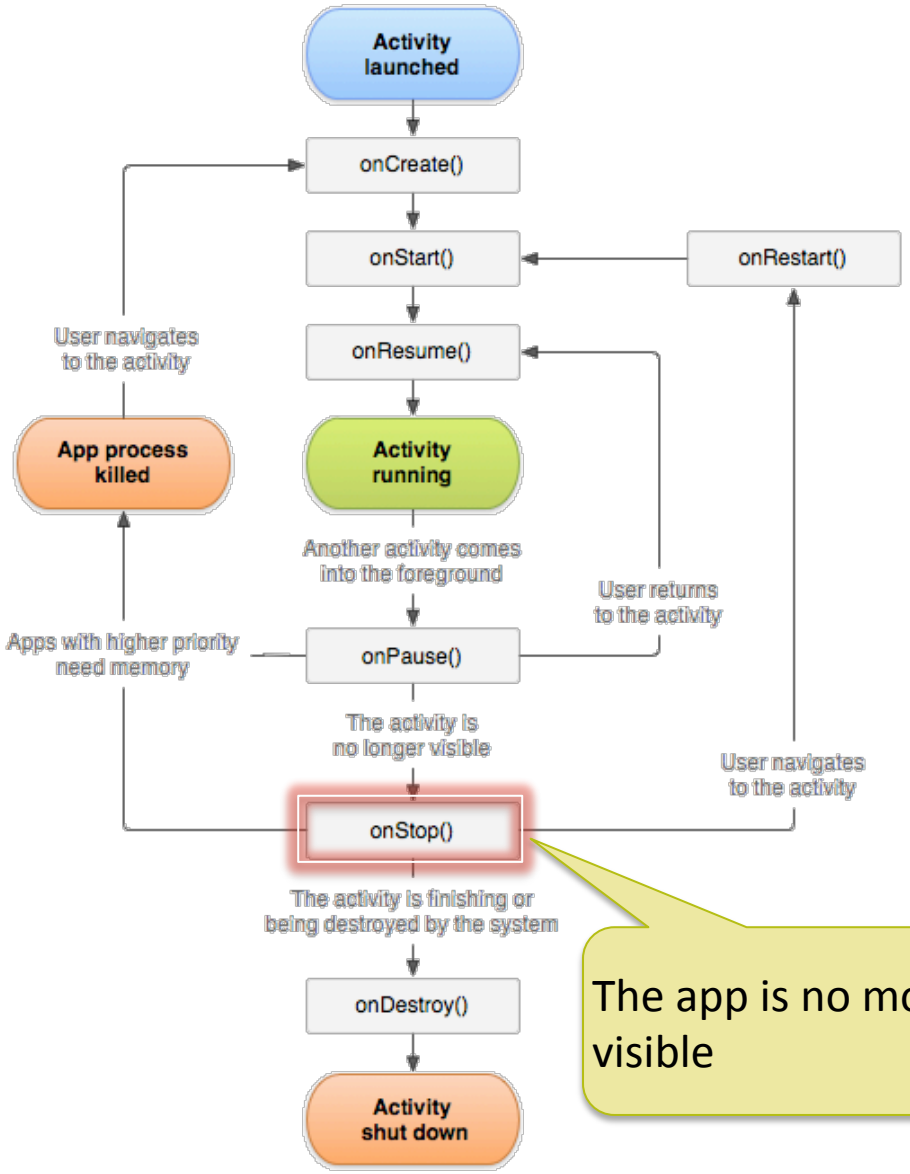
Example



As Gmaps is about to come to foreground the app goes in pause state

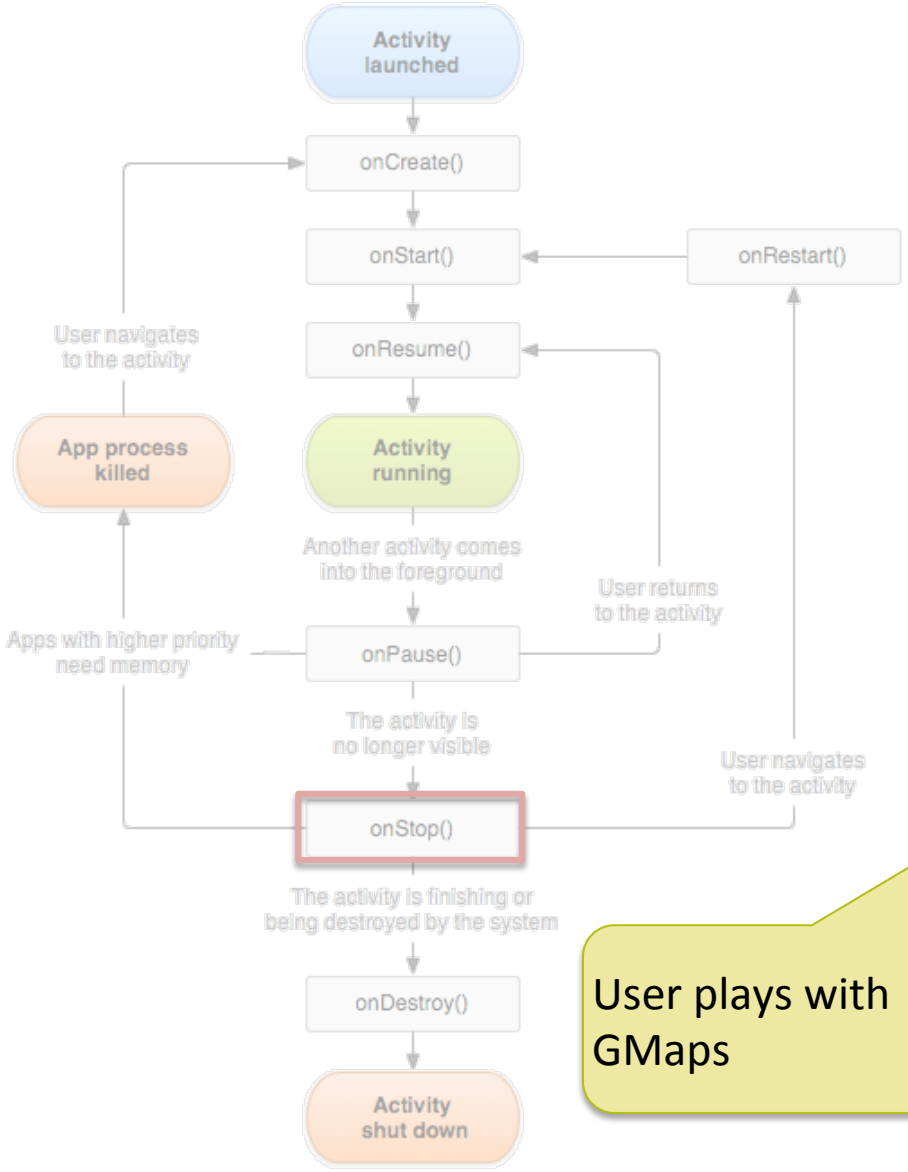


Example

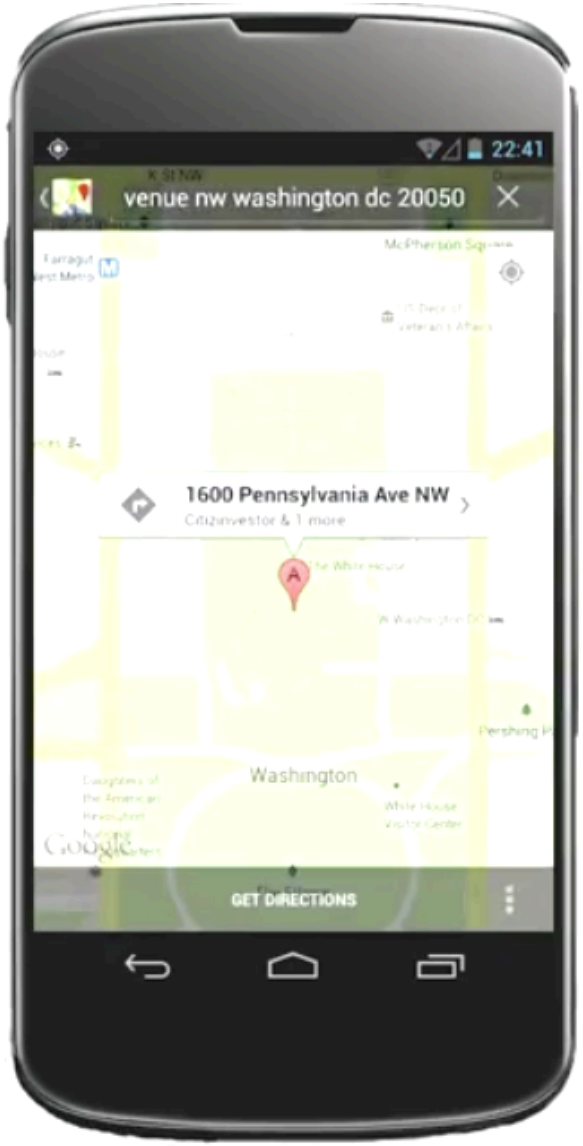

Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down
The app is no more visible


22:41
Maps

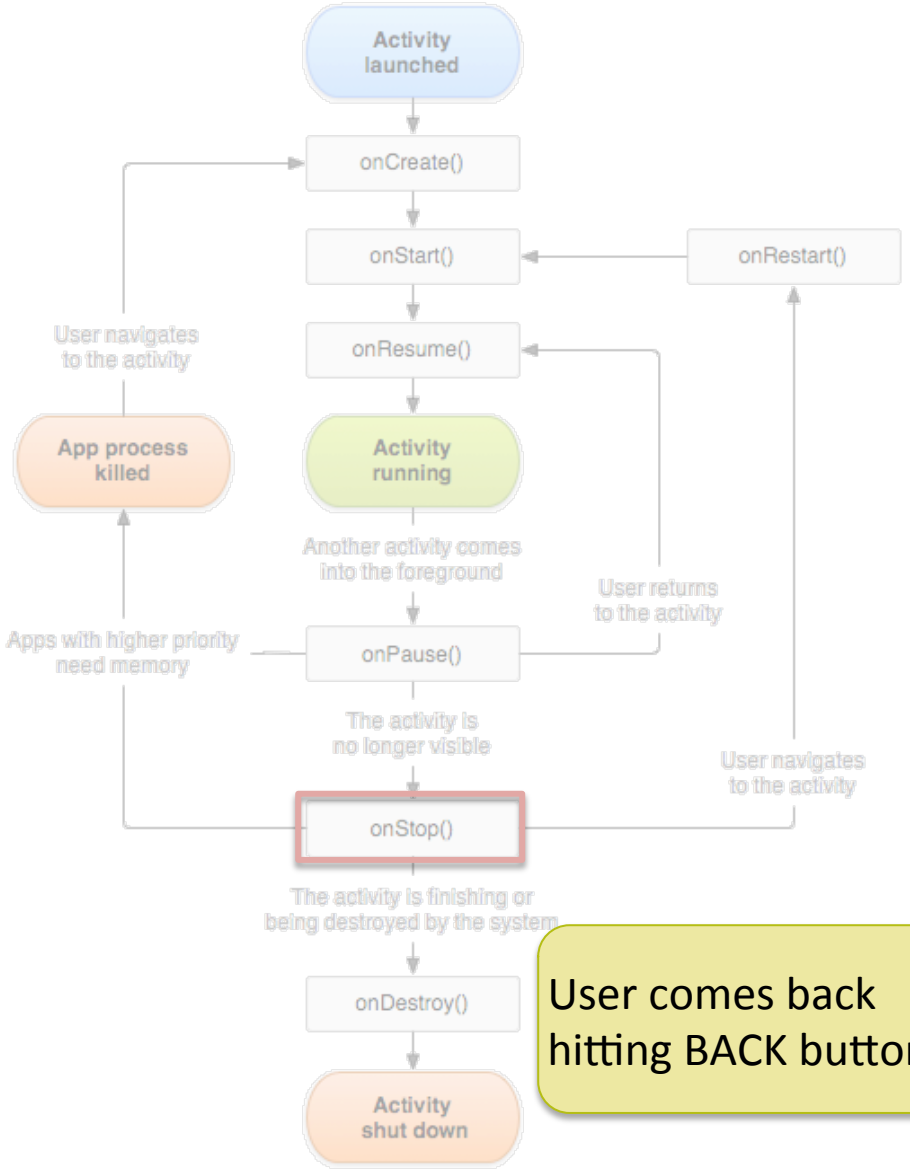
Example



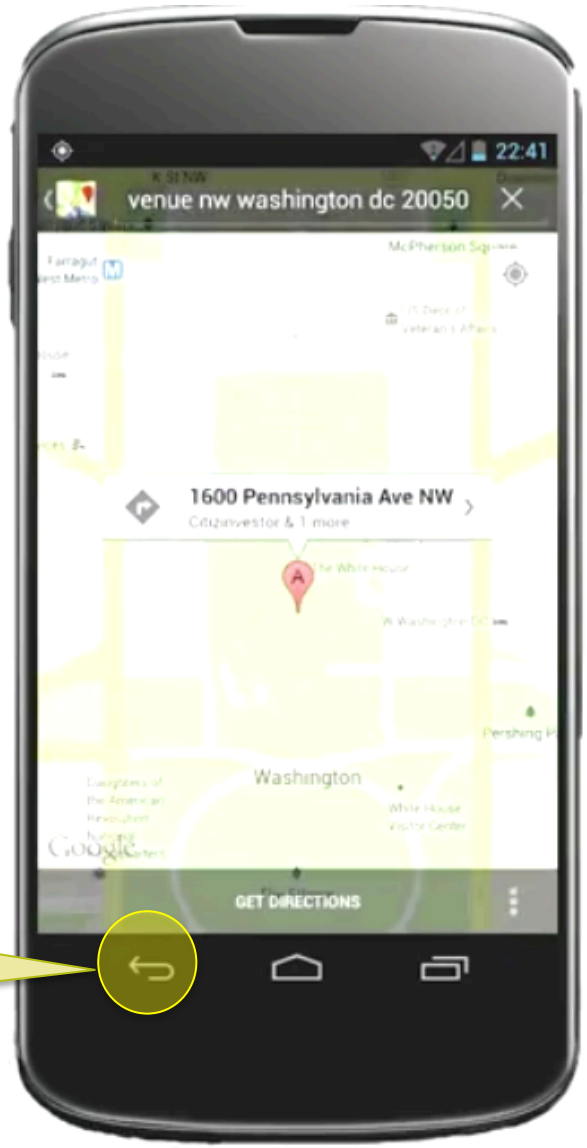
User plays with GMaps



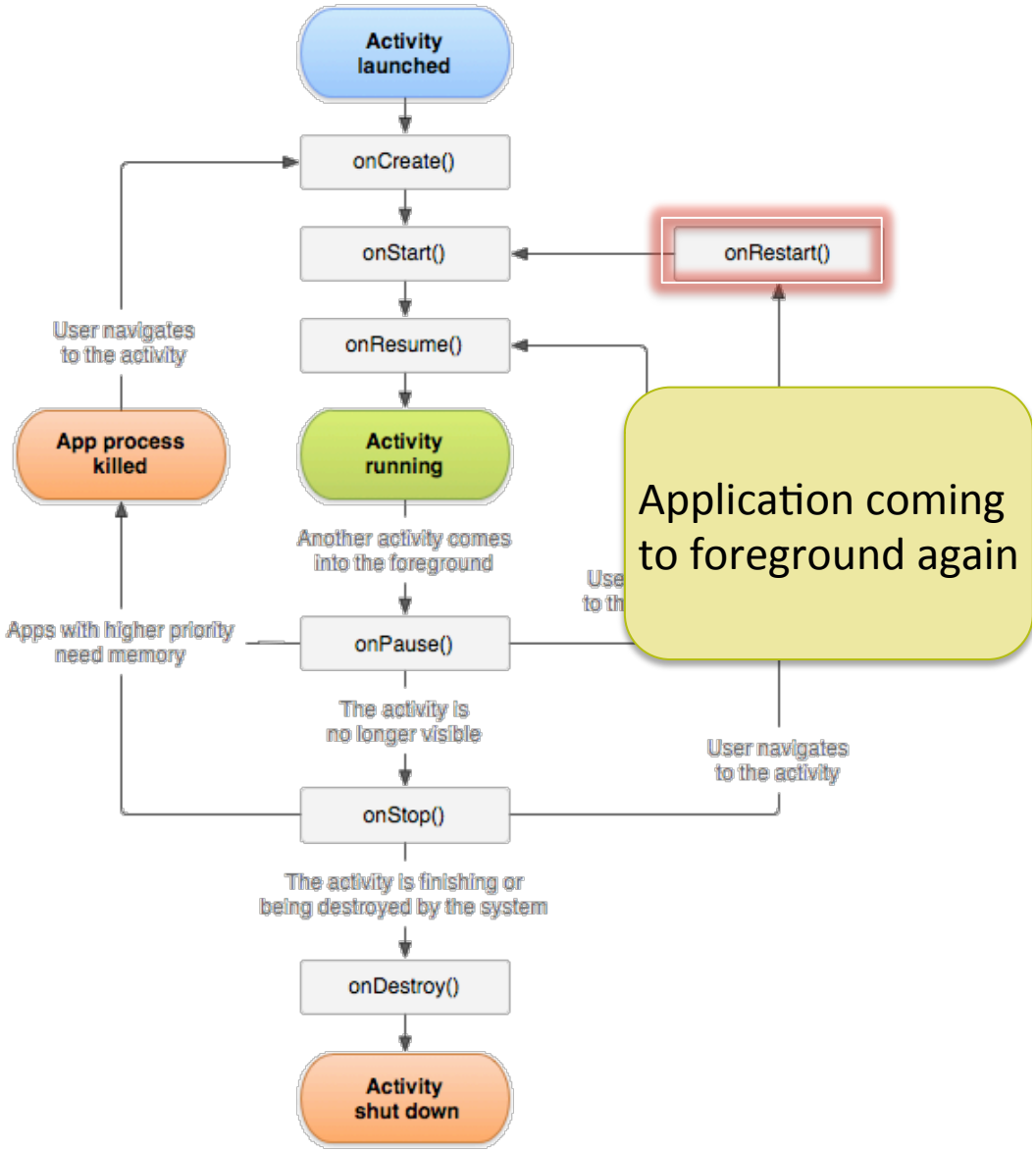
Example



User comes back hitting BACK button

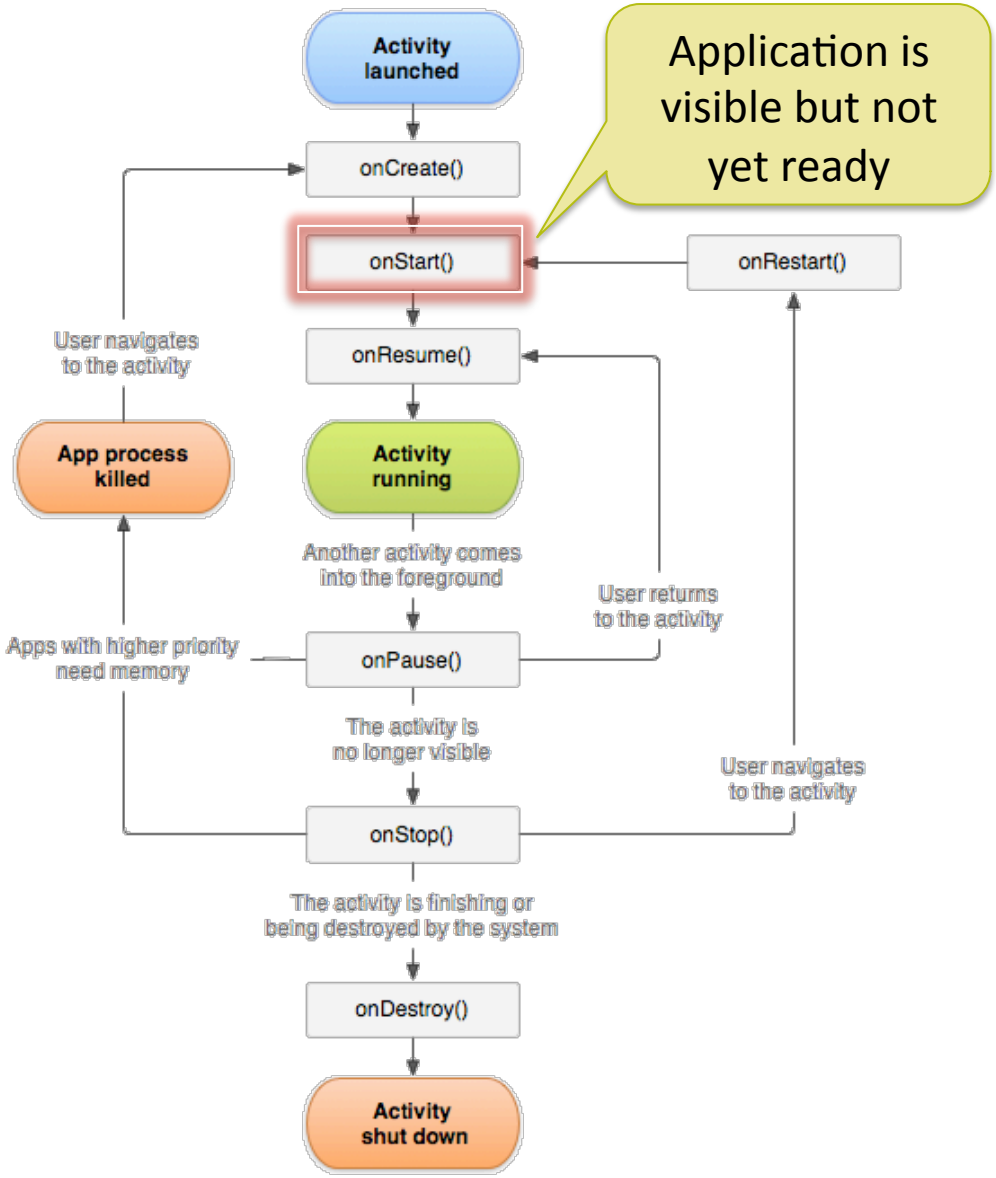


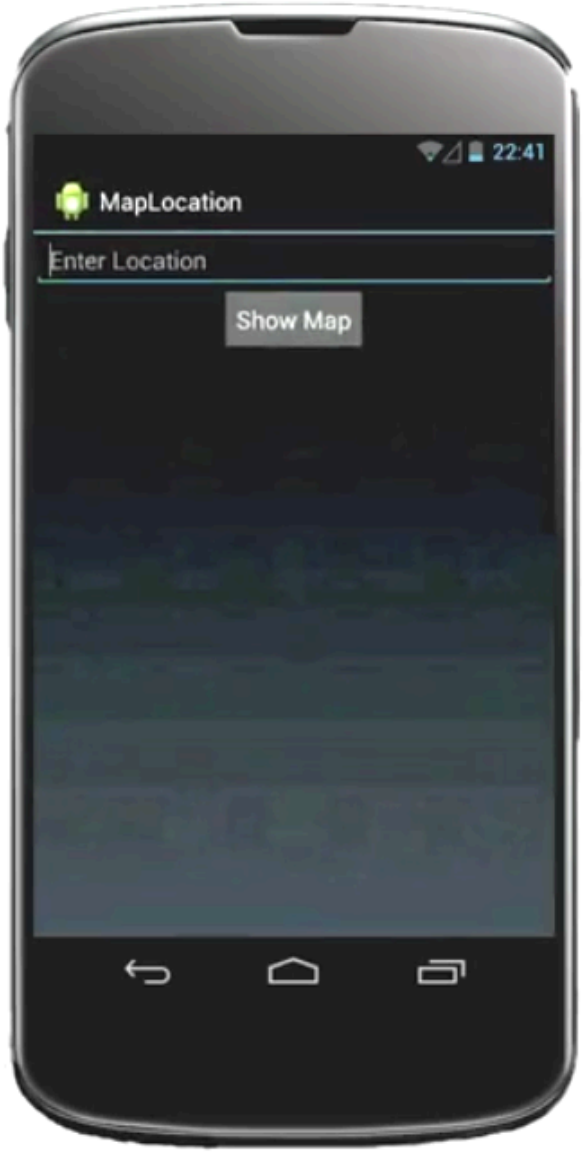
Example



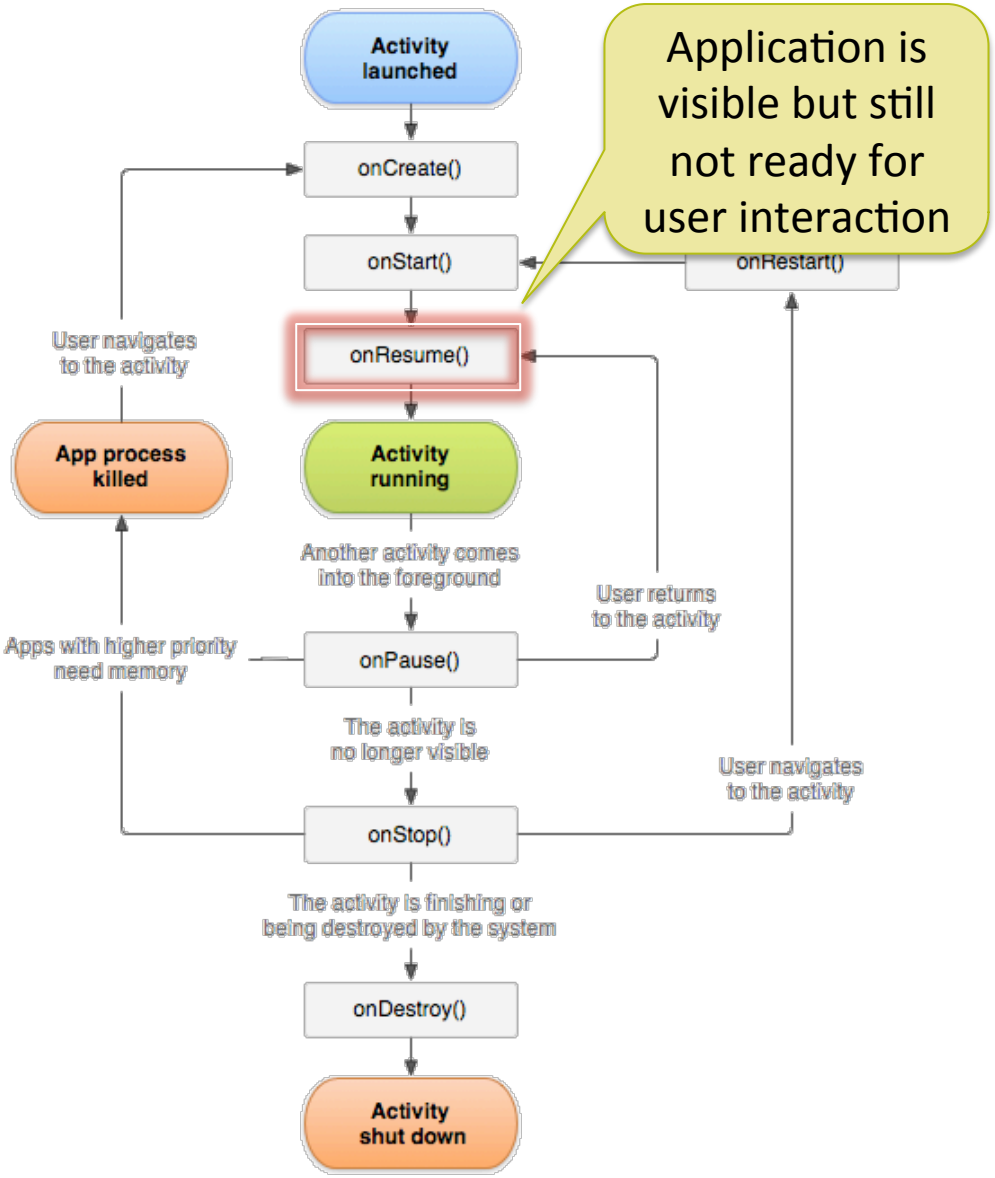
Application coming to foreground again

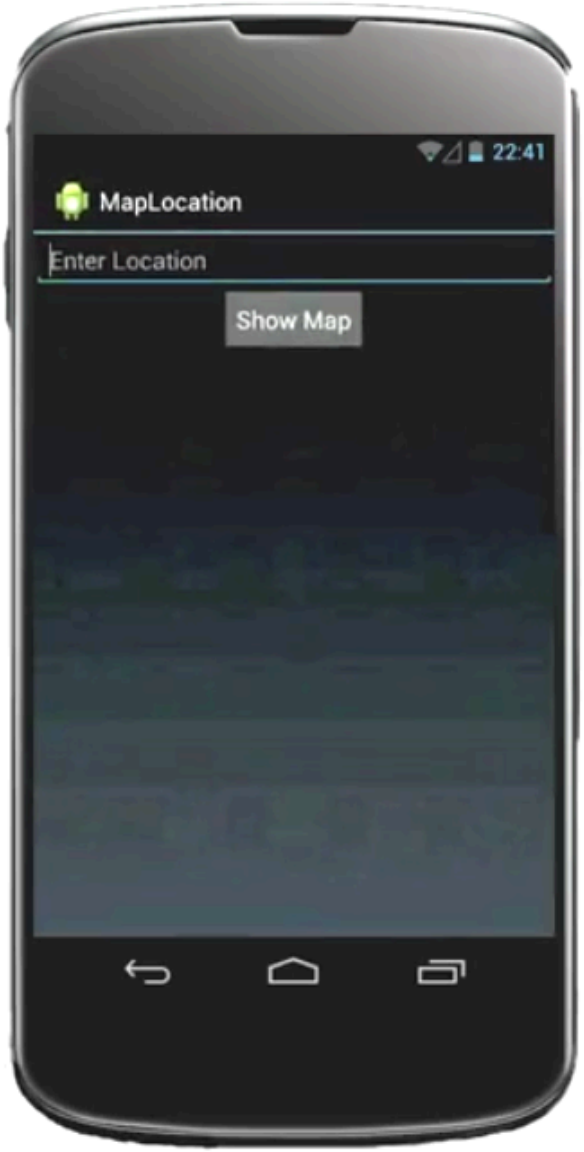
Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down
Application is visible but not yet ready

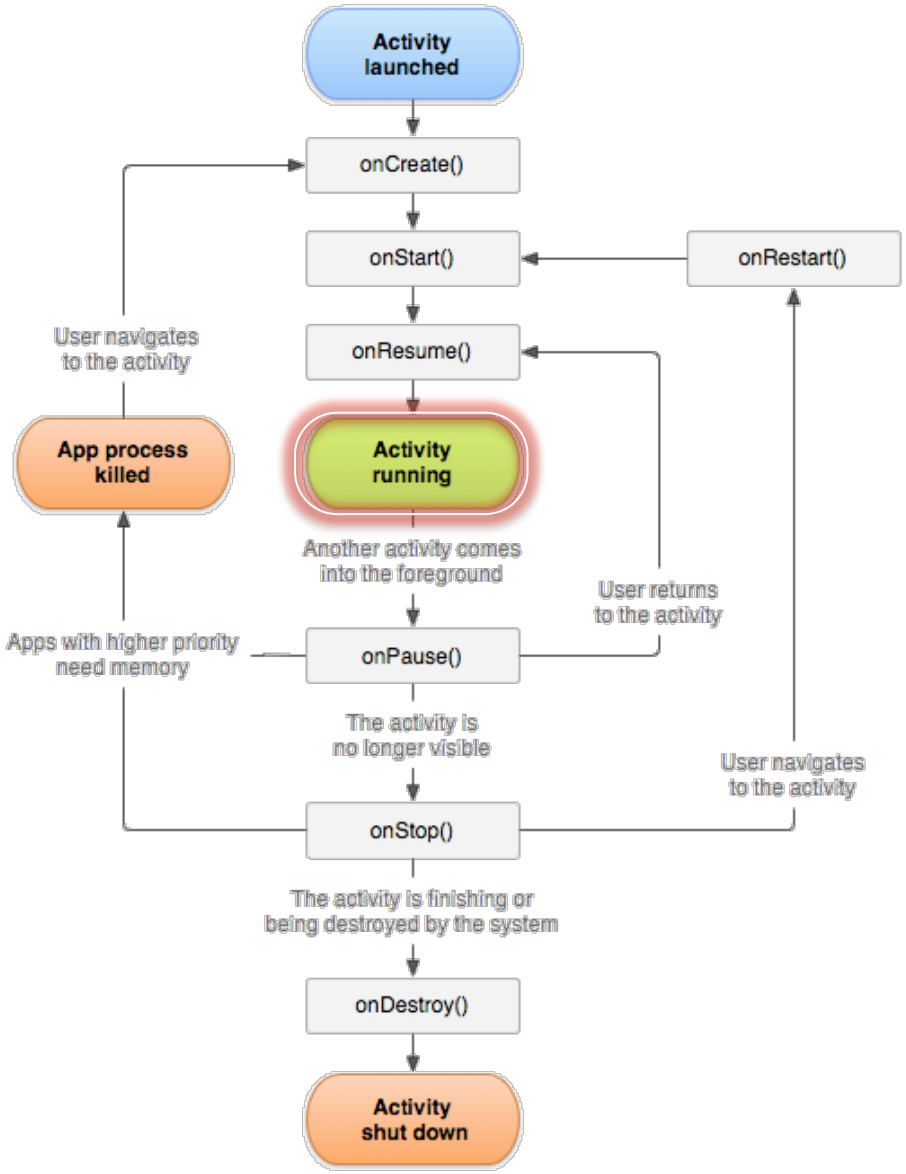

22:41
MapLocation
Enter Location
Show Map

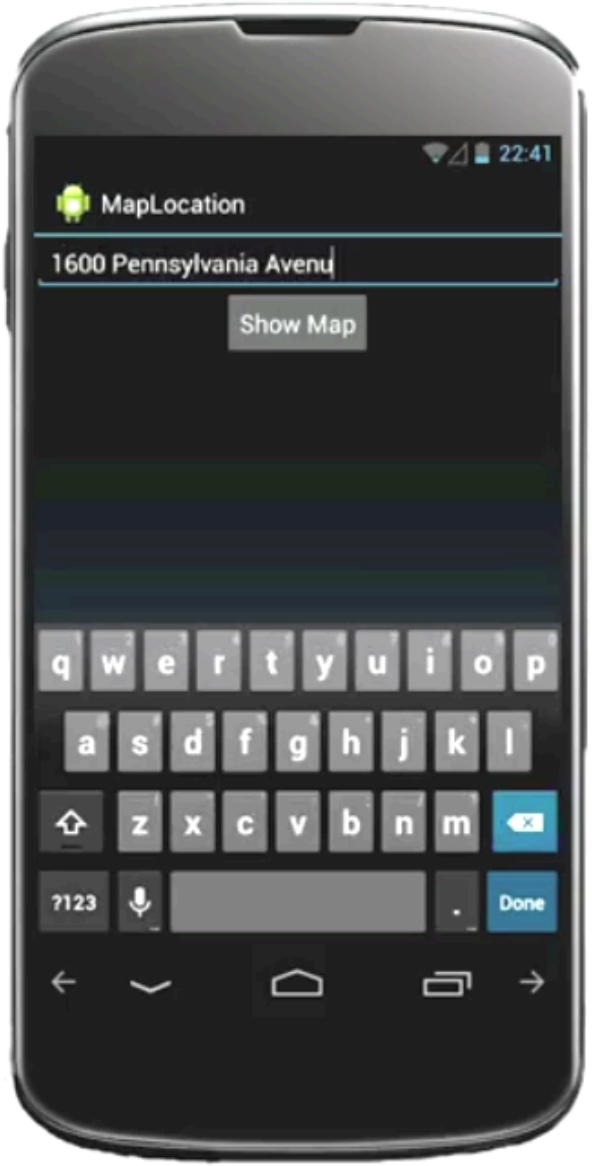
Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down
Application is visible but still not ready for user interaction

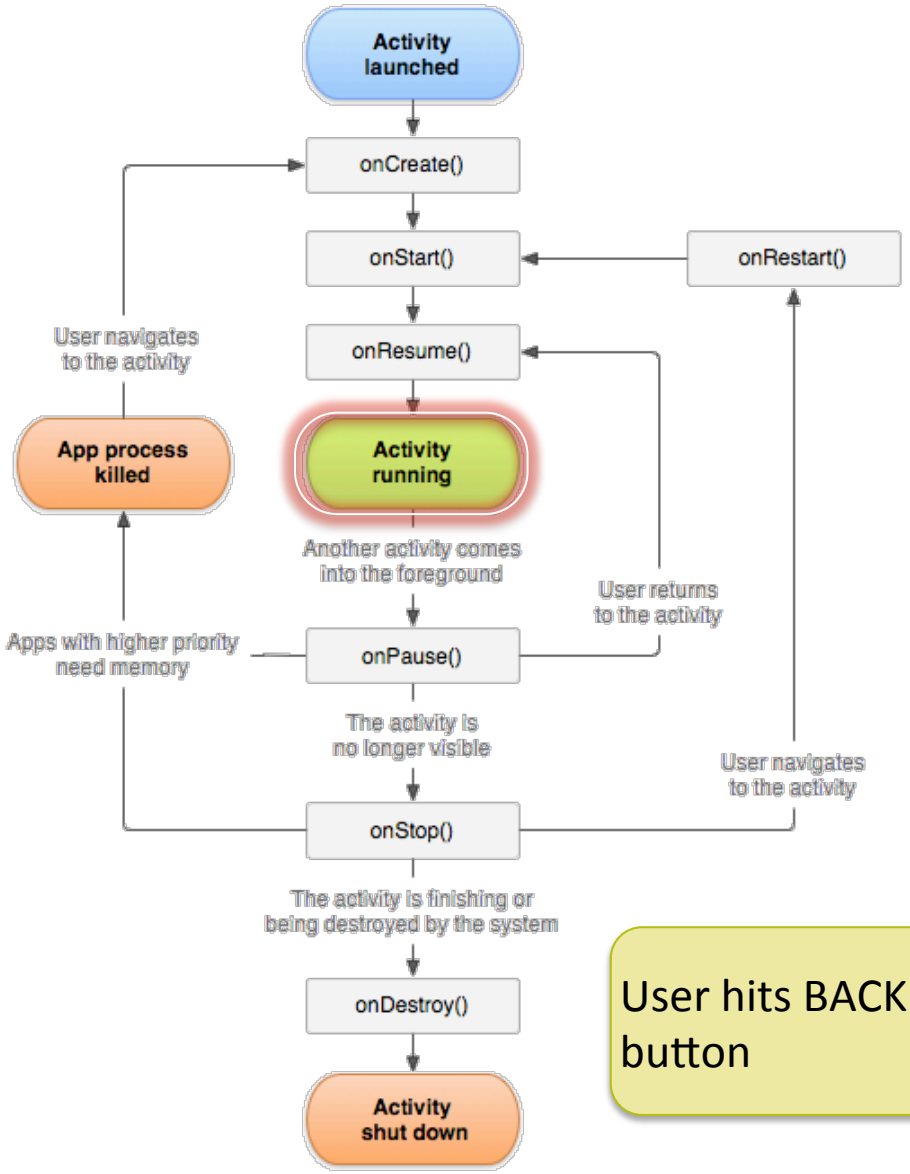

22:41
MapLocation
Enter Location
Show Map

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
App process killed
User navigates to the activity
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

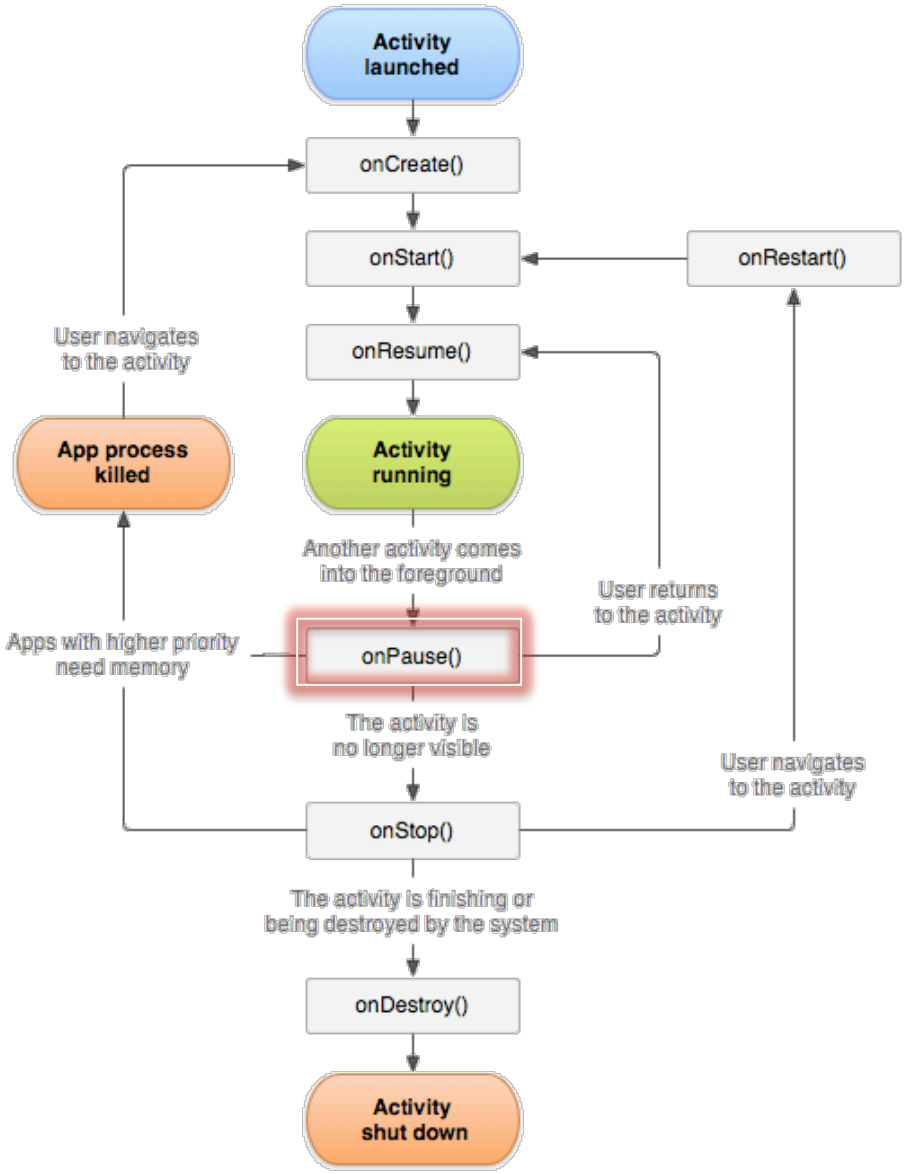

22:41
MapLocation
1600 Pennsylvania Avenu
Show Map
Done

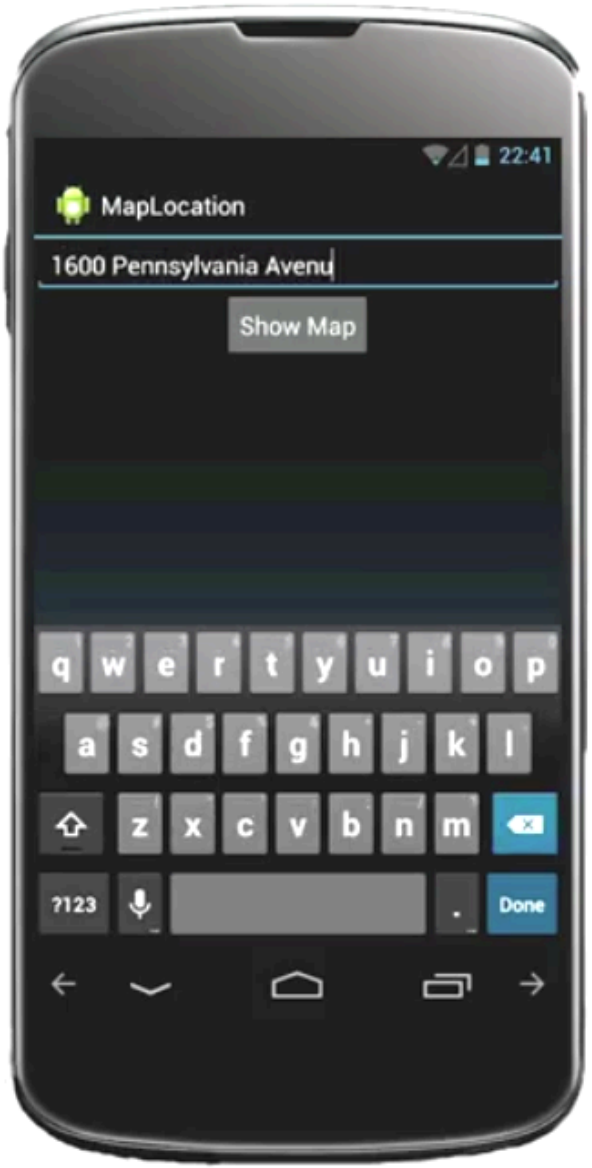
Example



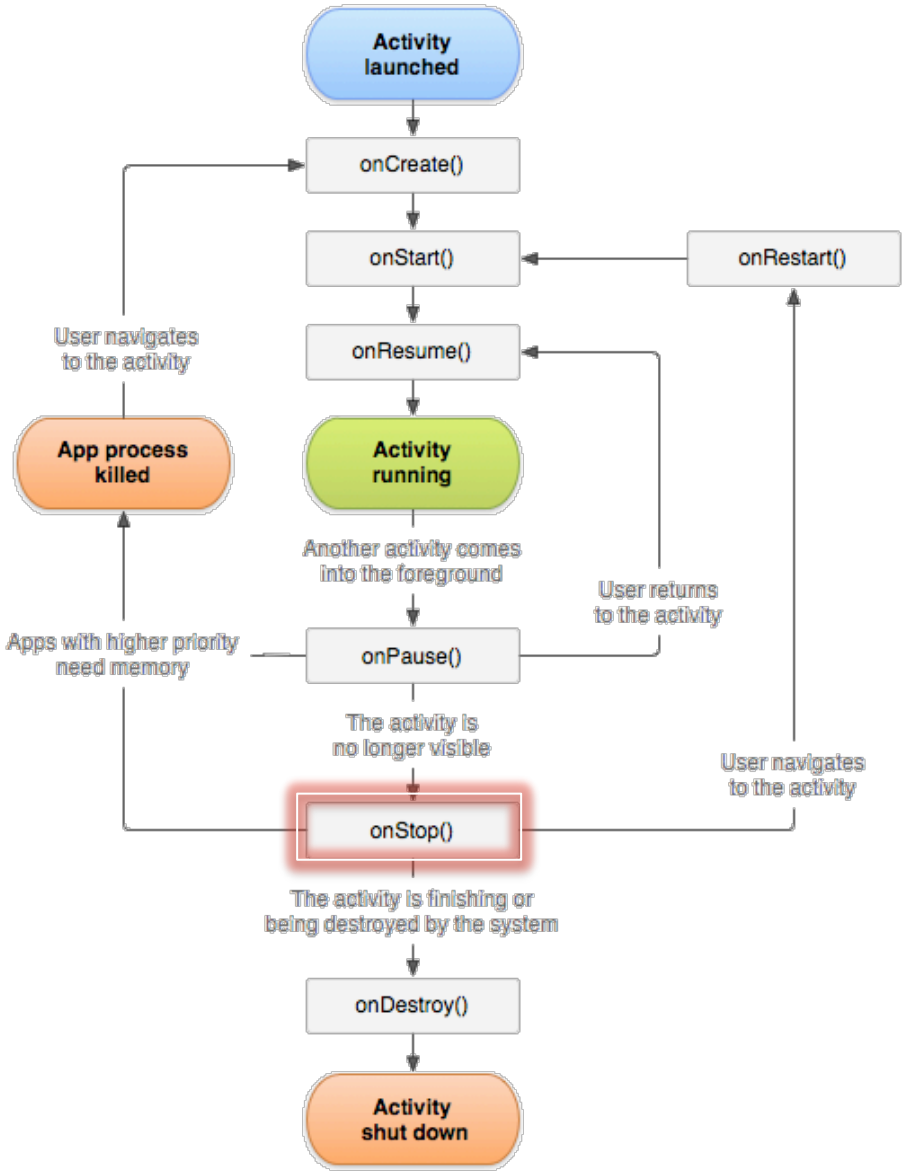
User hits BACK button

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
User navigates to the activity
App process killed
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

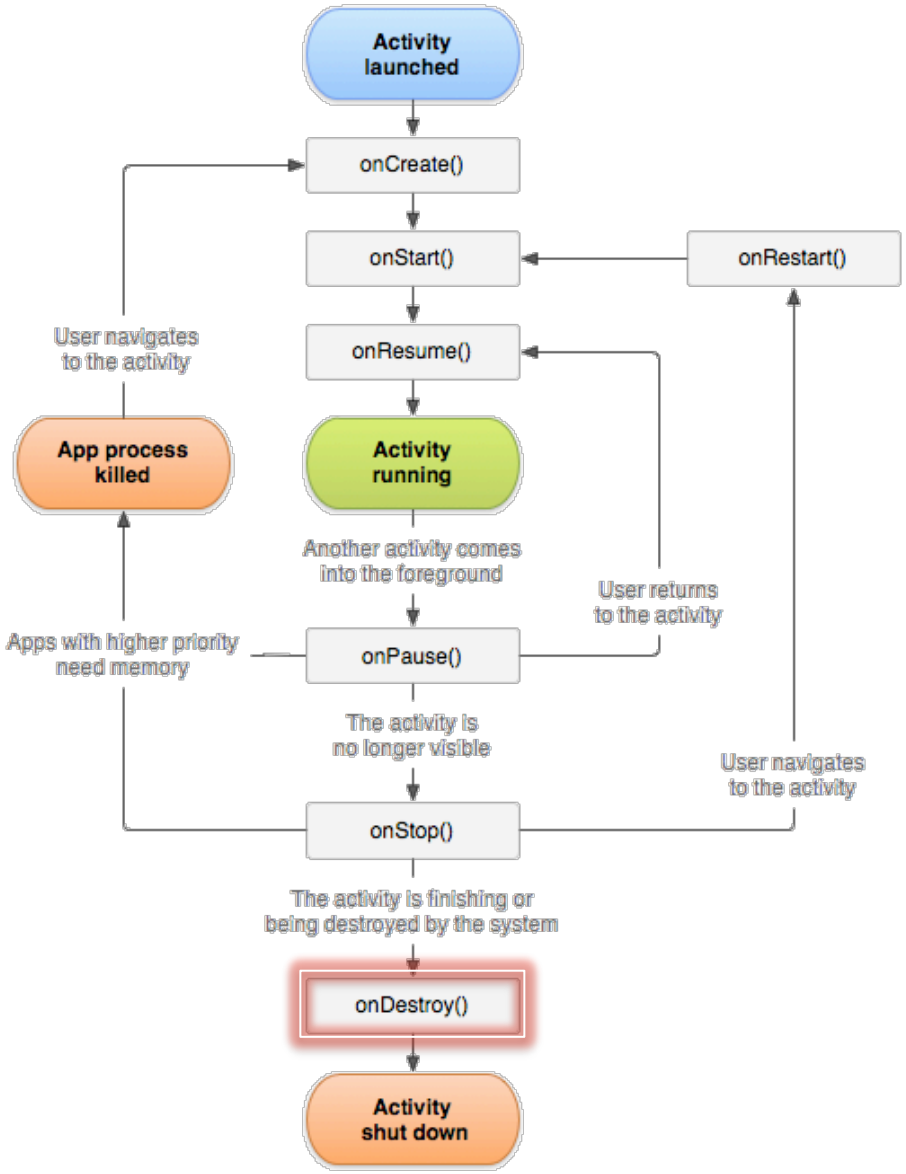

22:41
MapLocation
1600 Pennsylvania Avenu
Show Map
Done

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down


22:41
MapLocation
1600 Pennsylvania Avenu
Show Map
Done

Example


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down


22:41
APPS
WIDGETS
AirDroid
Amazon Kindle
Browser
Calculator
Calendar
MapLocation
Enter Location
Camera
Chrome
Clock
Downloads
Email
Show Map
ES File Explorer
Gallery
Gin Rummy Free
Gmail
Google
Google Settings
Google+
Jota Text Editor
Local
Map Location
MapLocation
Maps
Messaging
Messenger
MoboPlayer

Activity lifecycle

- Android attempts to keep application process alive for as long as possible.
- It will kill less important processes before killing more important processes

Order of importance

1. The **foreground activity** (the activity at the top of the screen)

   Only as a last resort (ie application hanging)

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

Activity lifecycle

- Android attempts to keep application process alive for as long as possible.
- It will kill less important processes before killing more important processes

Order of importance

1. The **foreground activity** (the activity at the top of the screen)
   Only as a last resort (ie application hanging)
2. A **visible activity** (eg. sitting behind a foreground dialog)
   Not killed unless that is required to keep the foreground activity running.

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

Activity lifecycle

- Android attempts to keep application process alive for as long as possible.
- It will kill less important processes before killing more important processes

Order of importance

1. The **foreground activity** (the activity at the top of the screen)
   Only as a last resort (ie application hanging)
2. A **visible activity** (eg. sitting behind a foreground dialog)
   Not killed unless that is required to keep the foreground activity running.
3. A **background activity** (not visible to the user and paused)
   Safely killed whenever memory is needed

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

Activity lifecycle

- Android attempts to keep application process alive for as long as possible.
- It will kill less important processes before killing more important processes

Order of importance

1. The **foreground activity** (the activity at the top of the screen)

   Only as a last resort (ie application hanging)

2. A **visible activity** (eg. sitting behind a foreground dialog)

   Not killed unless that is required to keep the foreground activity running.

3. A **background activity** (not visible to the user and paused)

   Safely killed whenever memory is needed

4. An **empty process** (no activities, services or Broadcast Receivers)

   Always killed → all background operation must always be associated to services

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

Hello World! Application

- Demo
- Activity class
  - Lifecycle
- **The GUI**
- Manifest file

Hello World!

```
public class MainActivity extends Activity
{
    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );
    }
```

Load the GUI elements from the resource class

The application resources

- Main idea: keep all the elements of the visual presentation of the app **separated from the code**
- Example:
  - Graphical Interface
  - Text of each component (buttons, field texts etc)
  - Icons, images
  - Other images, video, pdf, webpages…
- Each element is added to the app with an unique integer ID
- The class `R.java` collects all the IDs

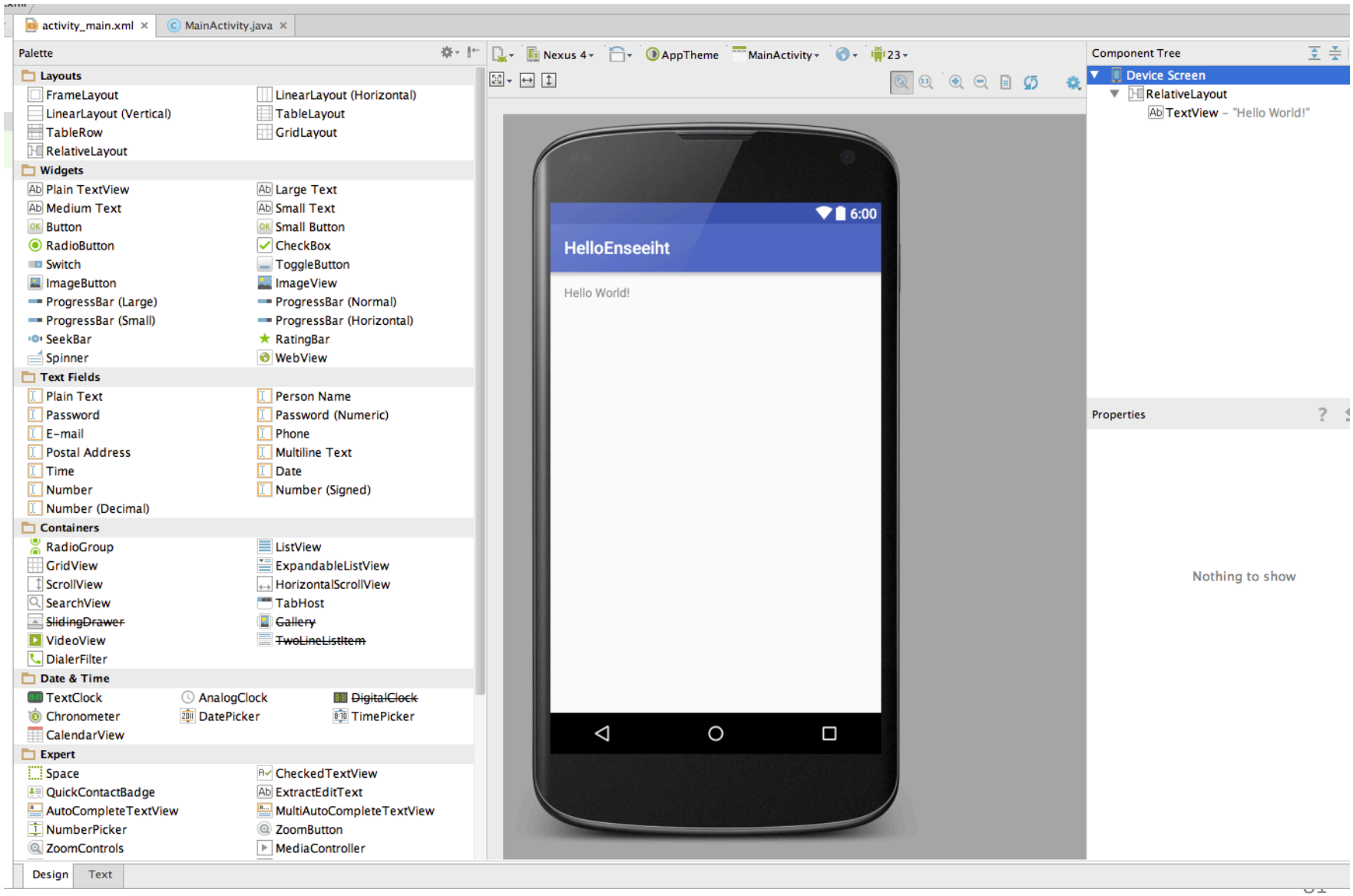
```
//Load the graphical layout activity_main.xml
setContentView( R.layout.activity_main );
```

Managing the GUI

- The most common way: **XML layout file**
  - an XML layout file saved in your application resources.
  - Keep the design of your user interface separated from the code that defines the activity's behavior.

- The hard way: **programmatically**
  - Build the hierarchy inside the code creating the objects from the relevant classess
  - Add custom widget and layout subclassing the class view .
  - Good luck with that!

GUI Editor

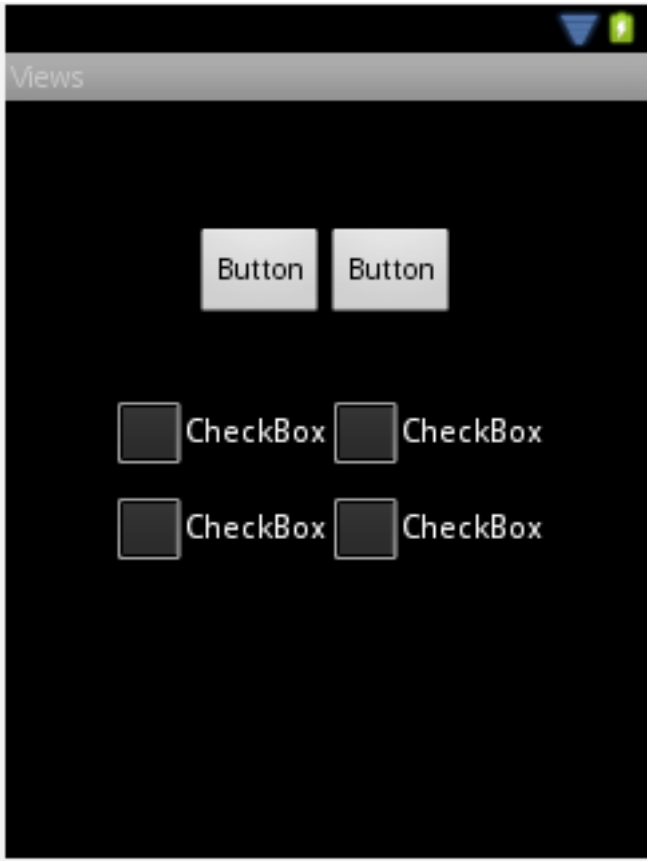

activity_main.xml
MainActivity.java
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal)
SeekBar
RatingBar
Spinner
WebView
Text Fields
Plain Text
Person Name
Password
Password (Numeric)
E-mail
Phone
Postal Address
Multiline Text
Time
Date
Number
Number (Signed)
Number (Decimal)
Containers
RadioGroup
ListView
GridView
ExpandableListView
ScrollView
HorizontalScrollView
SearchView
TabHost
SlidingDrawer
Gallery
VideoView
TwoLineListItem
DialerFilter
Date & Time
TextClock
AnalogClock
DigitalClock
Chronometer
DatePicker
TimePicker
CalendarView
Expert
Space
CheckedTextView
QuickContactBadge
ExtractEditText
AutoCompleteTextView
MultiAutoCompleteTextView
NumberPicker
ZoomButton
ZoomControls
MediaController
Nexus 4
AppTheme
MainActivity
23
6:00
HelloEnseeiht
Hello World!
Component Tree
Device Screen
RelativeLayout
TextView – "Hello World!"
Properties
Nothing to show
Design
Text

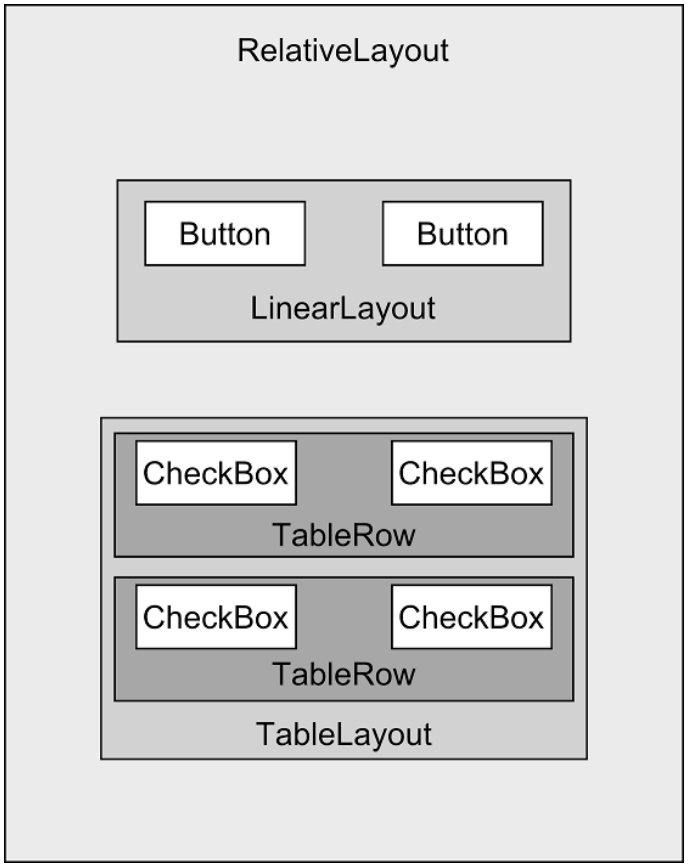
XML layout description

```
<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message" >
    </TextView>

</RelativeLayout>
```

activity_main.xml

The user interface

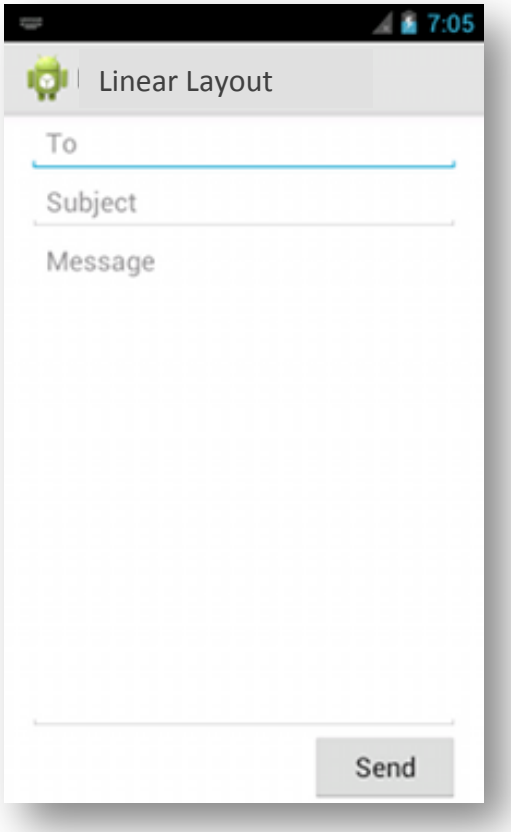




Linear Layout

- LinearLayout
- aligns all children in a single direction, vertically or horizontally.
- android:orientation="[vertical|horizontal]"

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
      android:orientation="vertical" >
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



XML layout description

```
<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message" >
    </TextView>

</RelativeLayout>
```

Unique ID of the element

Element content from `string.xml`

activity_main.xml

Hello World! Application

- Demo
- Activity class
  - Lifecycle
- The GUI
- **Manifest file**

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.enseeiht.HelloEnseeiht"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />
    <uses-permission />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.helloworld.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

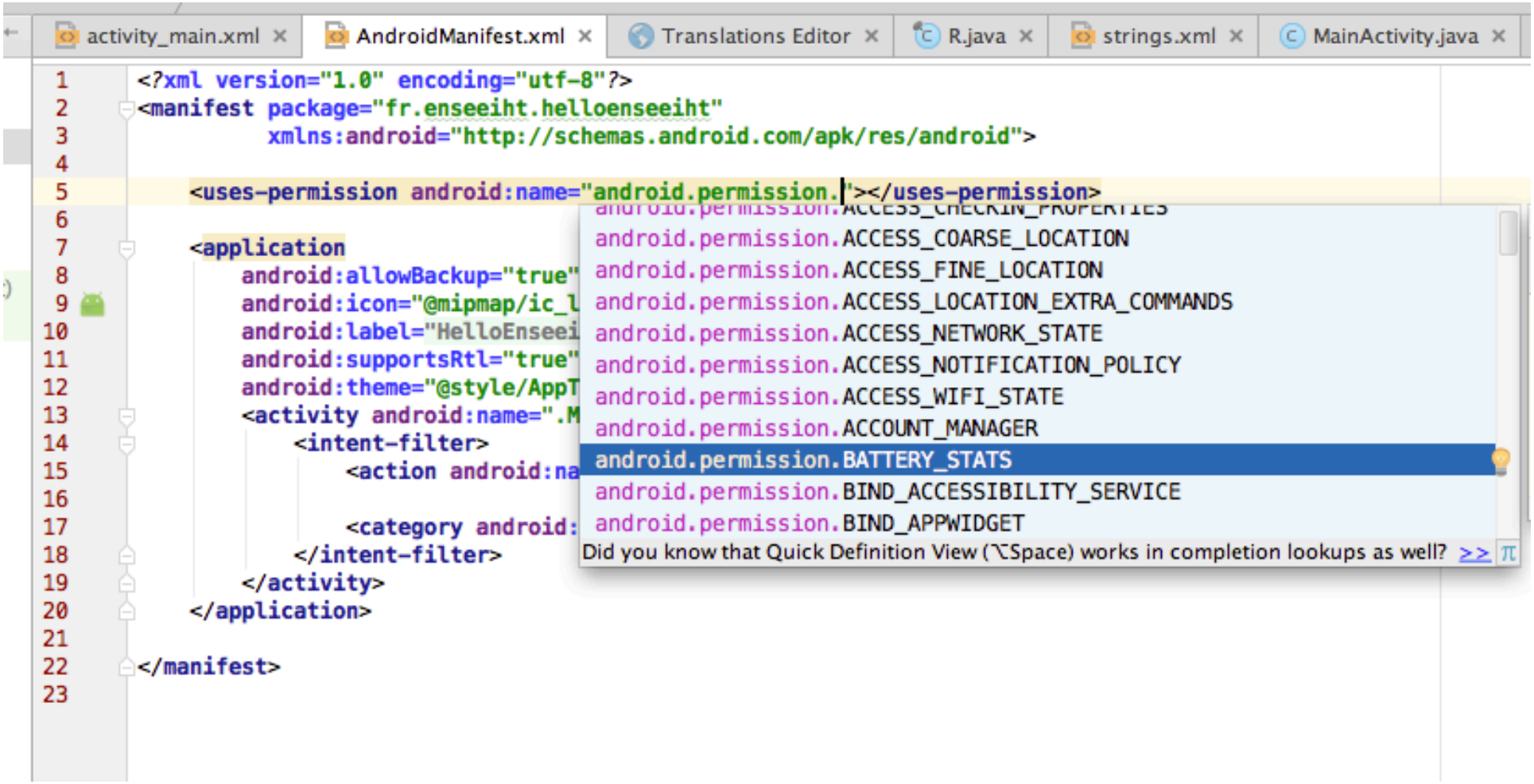
Android Manifest – Declaring permissions

- A list of tags `<uses-permission>` with all the permissions needed by the application
- Permissions allow to access to data or to specific hardware of the device
  - Eg contacts, messages, camera, flash light…
- If the permission is not listed there will be a run-time error when attempting to access the element

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

Example of runtime error due to missing permission

Android Manifest – Declaring permissions



Some examples of permissions

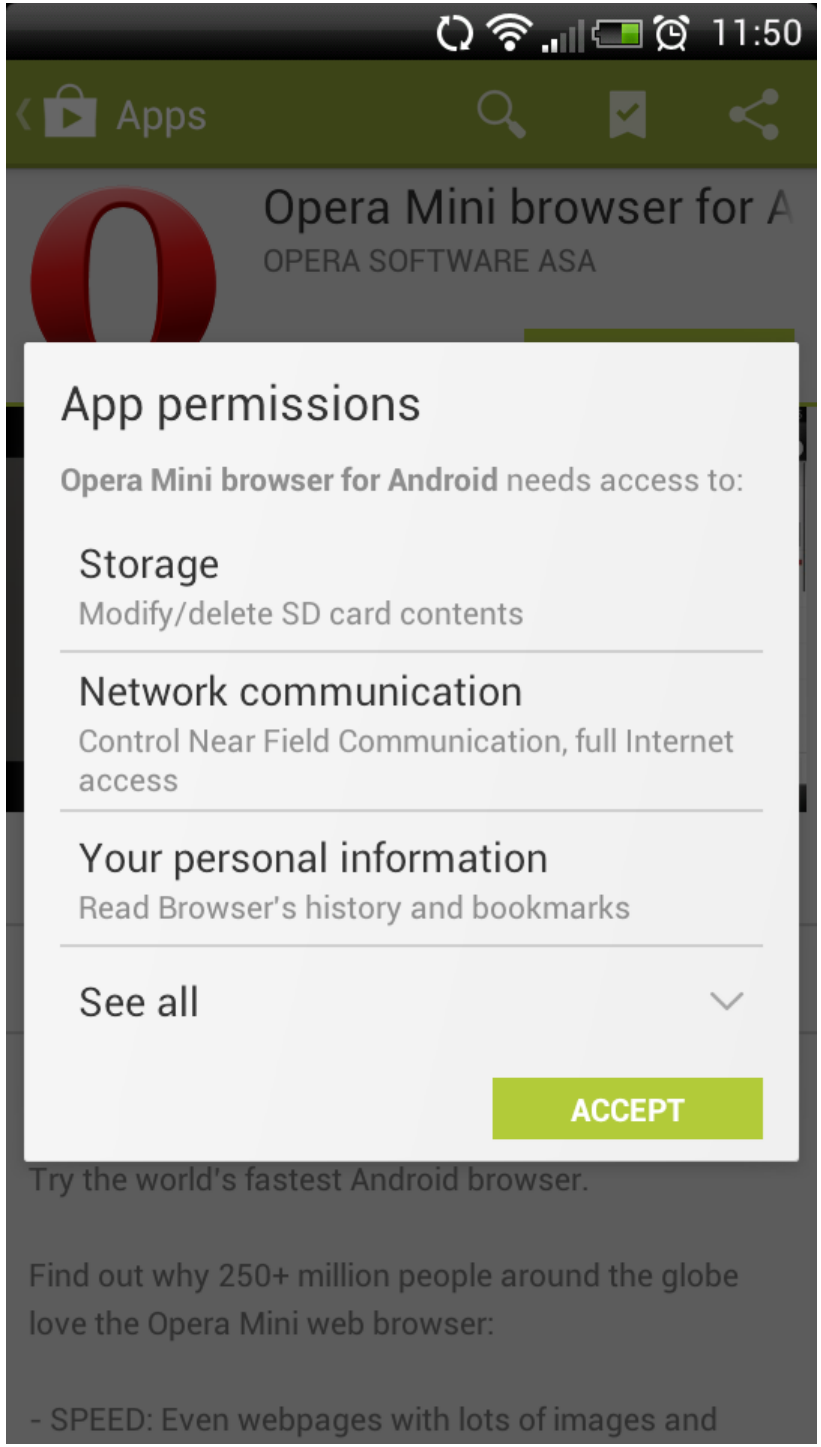
	Permission	Allow application to…
String	BLUETOOTH	connect to paired bluetooth devices
String	CALL_PHONE	initiate a phone call
String	CAMERA	access the camera device.
String	INTERNET	open network sockets.
String	NFC	perform I/O operations over NFC
String	READ_CONTACTS	read the user's contacts data.
String	READ_EXTERNAL_STORAGE	read from external storage.
String	READ_SMS	read SMS messages.
String	RECEIVE_SMS	monitor incoming SMS messages
String	RECORD_AUDIO	record audio
String	SEND_SMS	send SMS messages.
String	WRITE_CONTACTS	write the user's contacts data.
String	WRITE_EXTERNAL_STORAGE	write to external storage.
String	WRITE_SMS	write SMS messages.

http://developer.android.com/reference/android/Manifest.permission.html

Permission groups

- the system asks the user to grant the permissions at install time
- the system just tells the user what permission *groups* the app needs, not the individual permissions.

Permission group	Permissions
CALENDAR	READ_CALENDAR WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE CALL_PHONE READ_CALL_LOG WRITE_CALL_LOG ADD_VOICEMAIL USE_SIP PROCESS_OUTGOING_CALLS
SENSORS	BODY_SENSORS
SMS	SEND_SMS RECEIVE_SMS READ_SMS RECEIVE_WAP_PUSH RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE


11:50
Apps
Opera Mini browser for A
OPERA SOFTWARE ASA
App permissions
Opera Mini browser for Android needs access to:
Storage
Modify/delete SD card contents
Network communication
Control Near Field Communication, full Internet access
Your personal information
Read Browser's history and bookmarks
See all
ACCEPT
Try the world's fastest Android browser.
Find out why 250+ million people around the globe love the Opera Mini web browser:
- SPEED: Even webpages with lots of images and

Declaring the activities

```
<activity android:name=".MainActivity”
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

- Declare each activity of the app
- *android.intent.action.MAIN* declares the activity as the main activity
  - The one which will be called when the app starts for the first time
  - Only one main activity is allowed

- Intent filters contains the list of the intents to which the application can respond
  - More later…

Declaring custom permissions

- An application can also protect its own components with permissions using **android:permission**.
- It can employ any of the Android-defined permissions or declared by other applications.
- It can also define its own with the <permission> element.

```
<manifest . . . >
  <permission android:name="com.example.project.DEBIT_ACCT" . . . />
  <uses-permission android:name="com.example.project.DEBIT_ACCT" />
  . . .
  <application . . .>
    <activity android:name="com.example.project.FreneticActivity"
        android:permission="com.example.project.DEBIT_ACCT"
    . . . >
    . . .
    </activity>
  </application>
</manifest>
```

Recap

- **Activity Lifecycle**
  - Activities always kept alive,
  - terminated only when resources are needed
  - Order of importance in killing app
  - Callbacks available for each state transition (`onCreate()`, `onStart()` etc..)

Recap

- **Activity Lifecycle**
  - Activities always kept alive,
  - terminated only when resources are needed
  - Order of importance in killing app
  - Callbacks available for each state transition (`onCreate()`, `onStart()` etc..)
- **Resources**
  - Keep all the visual elements separated from the code
  - XML files to define the GUI
  - XML files to define strings
  - `R.java` automatically generated with the references to each element

Recap

- **Activity Lifecycle**
  - Activities always kept alive,
  - terminated only when resources are needed
  - Order of importance in killing app
  - Callbacks available for each state transition (`onCreate()`, `onStart()` etc..)
- **Resources**
  - Keep all the visual elements separated from the code
  - XML files to define the GUI
  - XML files to define strings
  - `R.java` automatically generated with the references to each element
- **Manifest file**
  - All the components (activities, services, content provider…)
  - Permissions granted to the app (access to resources / hardware)

Recap

- **Activity Lifecycle**
  - Activities always kept alive,
  - terminated only when resources are needed
  - Order of importance in killing app
  - Callbacks available for each state transition (`onCreate()`, `onStart()` etc..)
- **Resources**
  - Keep all the visual elements separated from the code
  - XML files to define the GUI
  - XML files to define strings
  - `R.java` automatically generated with the references to each element
- **Manifest file**
  - All the components (activities, services, content provider…)
  - Permissions granted to the app (access to resources / hardware)