

# CONCEPTION ET DEVELOPPEMENT D'APPLICATIONS INTERNET

## CDAI 5 - INTRODUCTION ANDROID



Université Paris Dauphine  
Master M2 MIAGE  
Année 2014-2015

Bekhouche Abdesslem  
Sobral Diogo



# PLAN

1. DEFINITION ANDROID
2. CIBLE : TERMINAUX VISES
3. CONCURRENCE
4. VERSIONS
5. REPARTITION DES VERSIONS
6. EVOLUTION DES ECRANS
7. PART DE MARCHE
8. PLATEFORMES ET ALTERNATIVES
9. ARCHITECTURE
10. CYCLE DE DEVELOPPEMENT
11. DEVELOPPEMENT (OUTILS, STRUCTURE et  
CONTENU, CYCLE DE VIE ACTIVITE)
12. CONCLUSION
13. WEBOGRAPHIE

# 1. DEFINITION ANDROID

- Système d'exploitation pour terminaux mobiles :
  - Ensemble de logiciels servant d'interface entre le matériel (composants du téléphone/tablette) et les logiciels applicatifs
- Basé sur Linux
- Open Source (licence Apache) :
  - Code disponible
  - Importante communauté
  - Environnement de développement gratuit
    - Programmation en Java ou en C
    - Kit de développement (SDK Android) disponible
- Souvent présenté comme l'alternative de Google à l'iPhone de Apple.

## 2. CONCURRENCE

- Apple IOS 13% : propriétaire, payant, mais moins de problématiques de compatibilité !
- Windows Phone
- Tizen : début 2013, open OS, développé par Samsung, pour des web applications HTML5, ....

Et d'autres encore (exemples) :

<http://www.techhive.com/article/2031515/four-alternatives-to-android-ios-and-windows-phone.html>

# 3. CIBLE : TERMINAUX VISES

- Téléphones portables (HTC, Samsung, Motorola...)
- Netbook/Smartbook (HP Airlife 100, Acer Aspire D250...)
- Tablette Multimedia (Archos, Samsung Galaxy Tab, ...)
- Automobile (Continental AutoLinq : Tesla, Ford...)

Avènement des objets connectés !

- Mais aussi : GPS, Réfrigérateur, Machine à laver...

# 4. VERSIONS

- Sources :
  - <http://www.android.com/history/>
  - <http://developer.android.com/tools/revisions/platforms.html>
- Versions d'Android :
  - 1.5 : Cupcake (Avril 2009)
  - 1.6 : Donut (Septembre 2009)
  - 2.0/2.1 : Eclair (Octobre 2009)
  - 2.2 : FroYo (Mai 2010)
  - 2.3 : Gingerbread (Novembre 2010)
  - 3.0 : Honeycomb (Février 2011)
  - 4.0 : Ice Cream Sandwich (ICS) (Décembre 2011)
  - 4.1/4.2/4.3 : Jelly Bean (Juin 2012)
  - 4.4 : KitKat (Octobre 2013)
  - 5.0 : Lollipop (Juin 2014)
- Remarques :
  - Evolution très rapide !
  - Problématique de déploiement

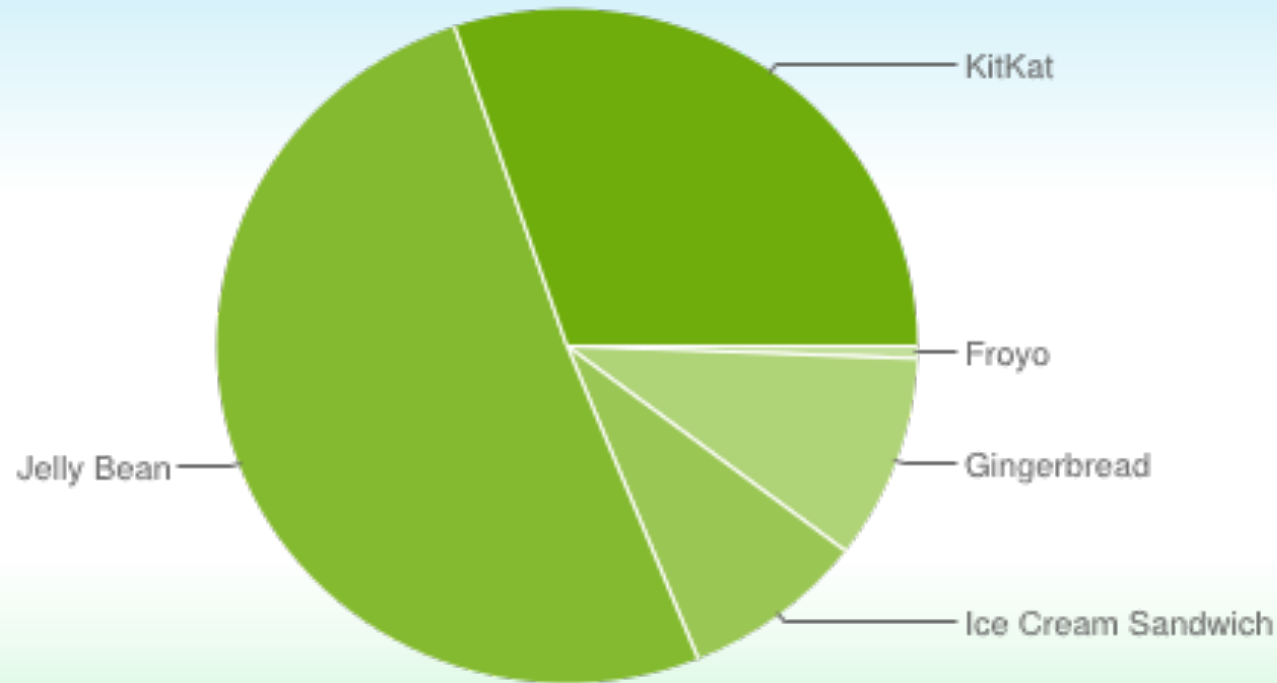
# 4. VERSIONS

- Source : <http://www.grapessoftware.com/history-of-android-then-and-now/>



# 5. REPARTITION VERSIONS

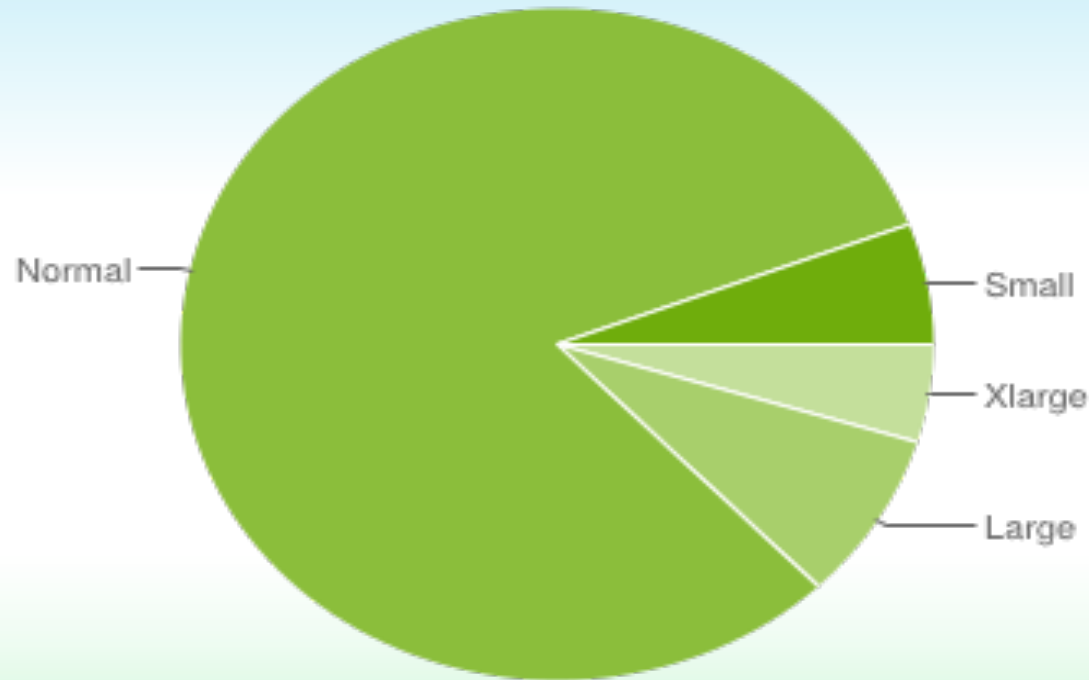
- Au 3 novembre 2014 (source <http://developer.android.com/about/dashboards/index.html>)





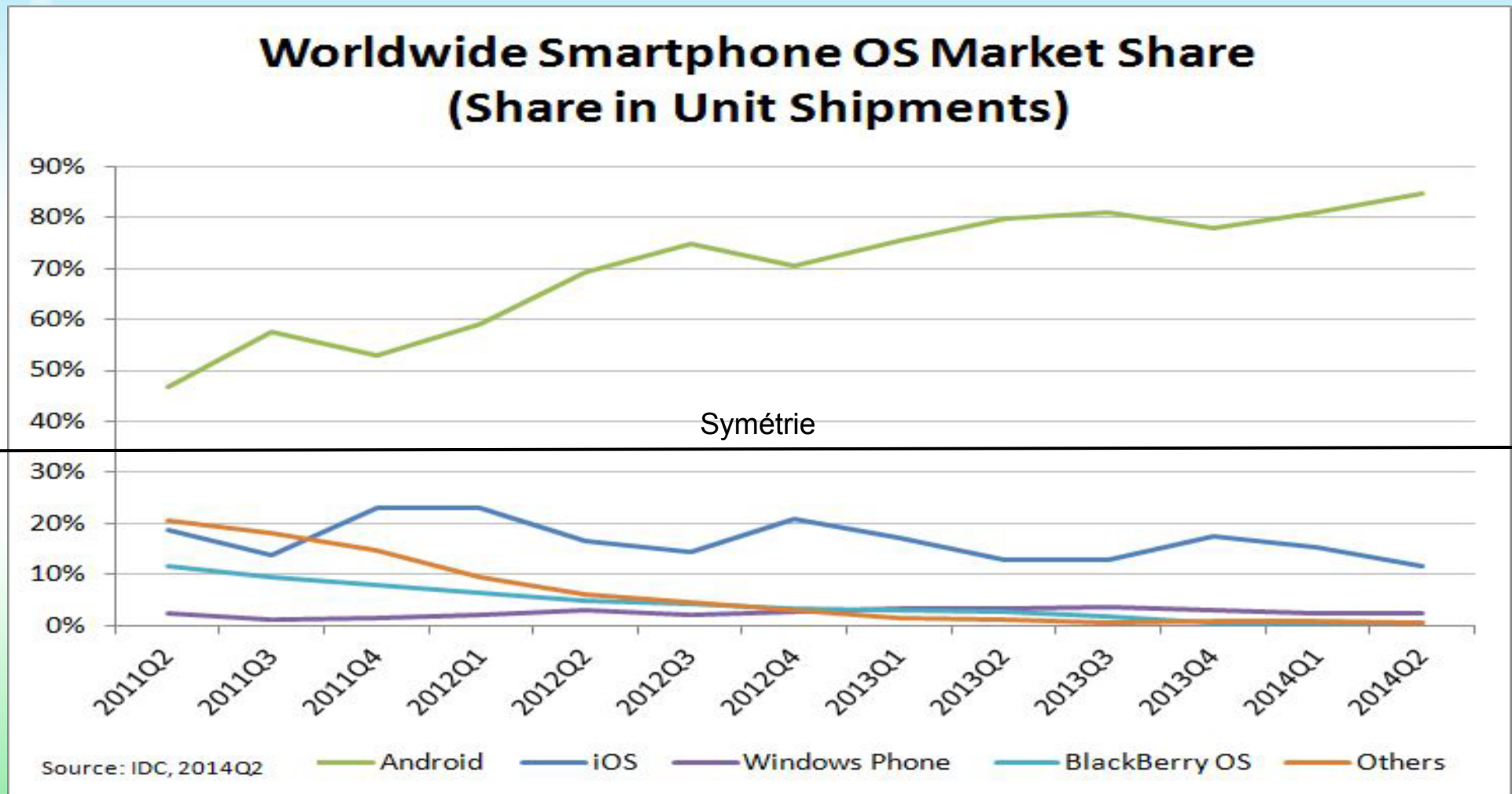
# 6. EVOLUTION DES ECRANS

- Au 3 novembre 2014 (source <http://developer.android.com/about/dashboards/index.html>)



# 7. PART DE MARCHE : GRAPHIQUE

- Au 2ème semestre 2014 (source <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>)



# 7. PART DE MARCHE : CHIFFRES

- Au 2ème semestre 2014 (source <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>)

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q2 2014	84.7%	11.7%	2.5%	0.5%	0.7%
Q2 2013	79.6%	13.0%	3.4%	2.8%	1.2%
Q2 2012	69.3%	16.6%	3.1%	4.9%	6.1%
Q2 2011	36.1%	18.3%	1.2%	13.6%	30.8%

Source: IDC, 2014 Q2

## 8. PLATEFORMES DE TELECHARGEMENT/DISTRIBUTION

Système standard de téléchargement/distribution d'applications.

- Pas de vérification des applications
- Nécessite un terminal certifié (caméra, 3G ...)
- Gestion des autorisations avant l'installation
- Des centaines de milliers d'applications disponibles (> 50% gratuites)

Si vous souhaitez vendre vos applications :

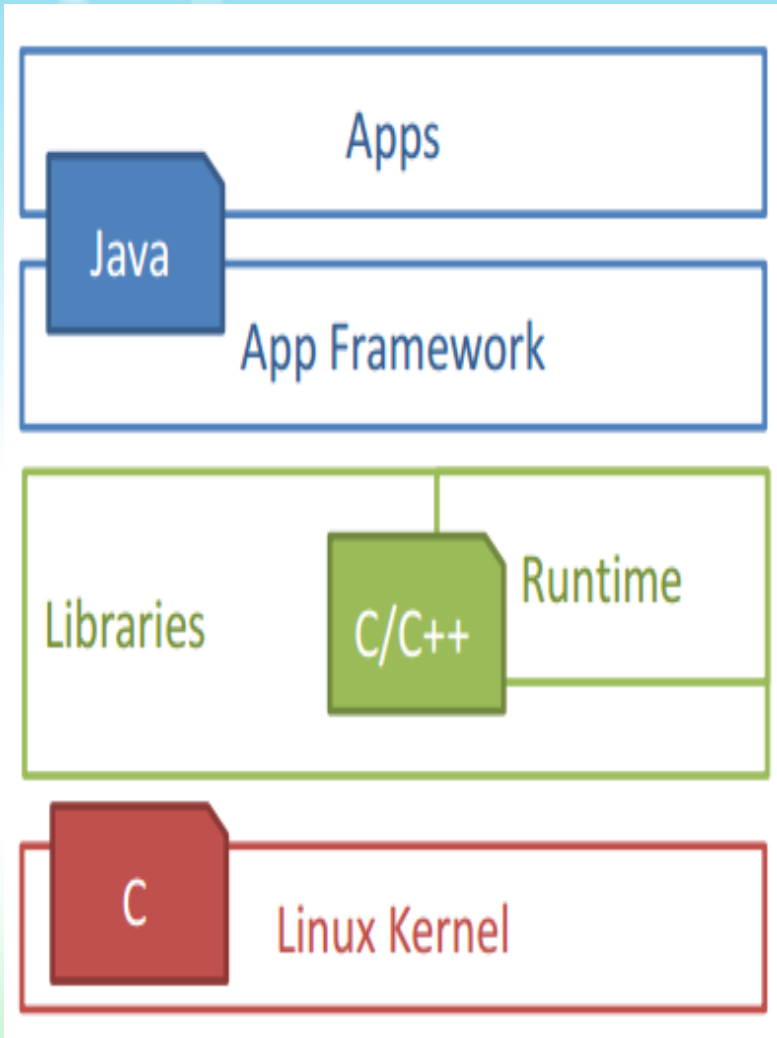
- 25\$ pour s'inscrire en tant que développeur
- 70% du prix revient au développeur, 30% à Google
- Revenus perçus via Google CheckOut

Cf. <https://play.google.com/apps/publish/signup/>  
<https://play.google.com/about/developer-distribution-agreement.html>  
<https://support.google.com/googleplay/android-developer/answer/113469?hl=fr>

# 8. ALTERNATIVES DE PLATEFORMES DE TELECHARGEMENT/DISTRIBUTION

<http://www.appslib.com/>  
<http://fr.androlib.com/>

# 9. ARCHITECTURE : VUE MACRO



**APPLICATION :** system installed (/system/app) and user installed (/data/app)

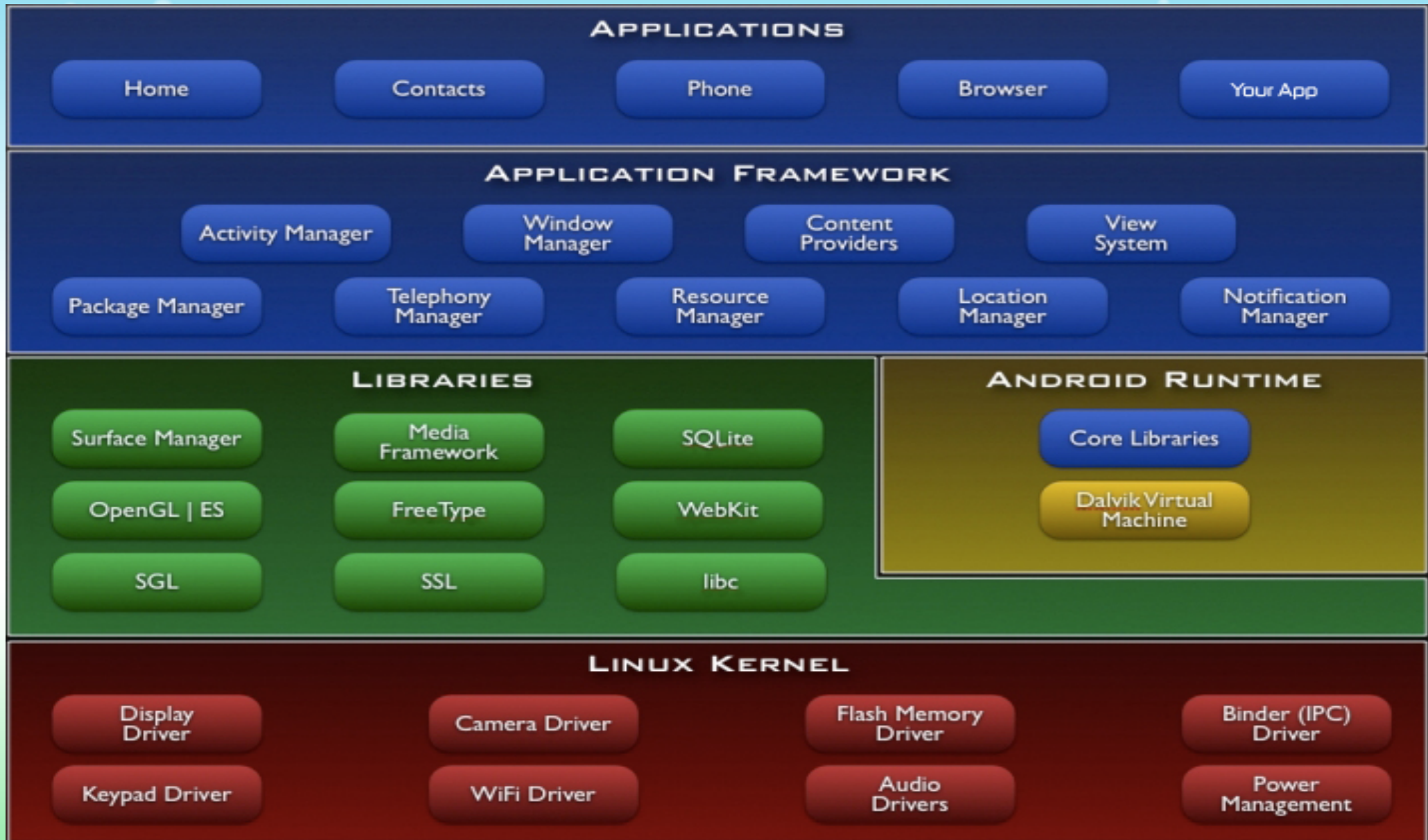
**APPLICATION FRAMEWORK :** basic functions of device (in use by developers)

**NATIVE LIBRAIRIES :** core libraries to support different types of data (audio and video formats, e. t.c.), Java libraries (Dalvik VM)

**LINUX KERNEL LAYER :** HAL, memory management, security, power management, drivers and network

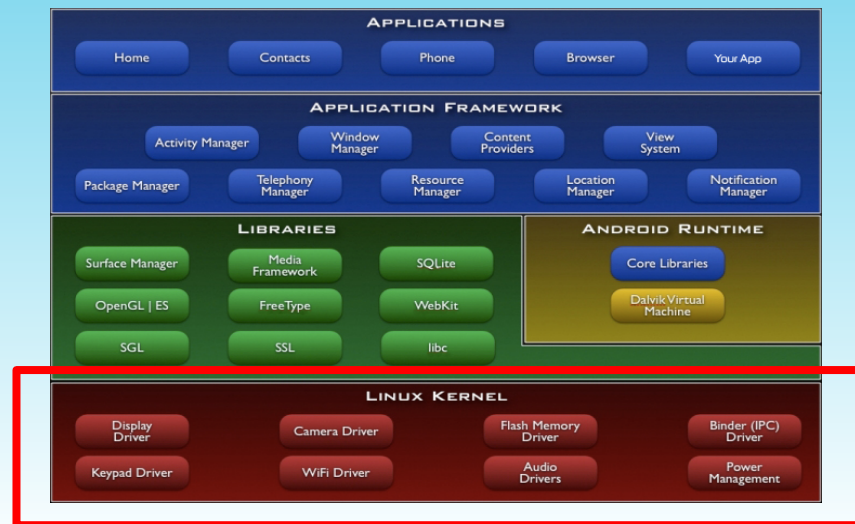
# 9. ARCHITECTURE : DÉTAILS

Source : <http://www.android-app-market.com/android-architecture.html>





# 9. ARCHITECTURE : LINUX KERNEL



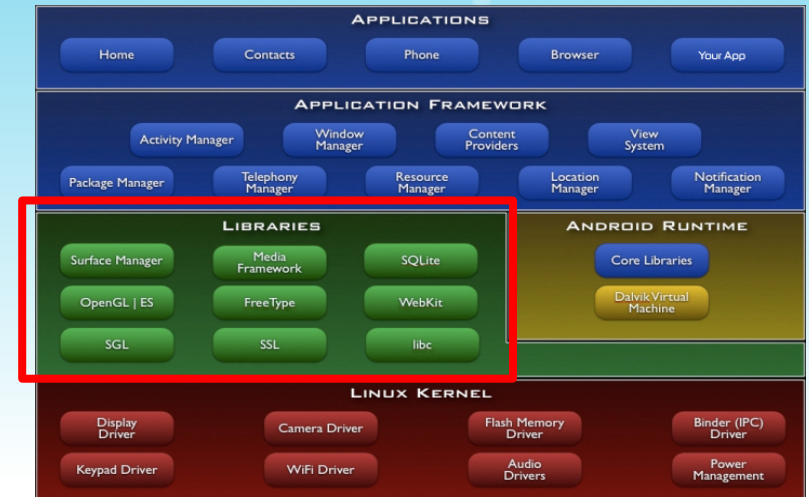
- Android OS est basé sur le noyau Linux 2.6
  - Hérite des fonctionnalités de base de Linux Core
    - Gestion de la mémoire
    - Gestion des processus
    - Réseau
    - Paramètres de sécurité
  - Drivers système



# 9. ARCHITECTURE : LIBRAIRIES

Librairies natives de Android :

- Libc: c standard library
- SSL: Secure Socket Layer
- SGL: 2D image engine
- OpenGL|ES: 3D image engine
- Media Framework: media codecs
- SQLite: Database engine
- WebKit: Kernel of web browser
- FreeType: Bitmap and Vector
- SurfaceManager: Compose window manager with off-screen buffering.



# 9. ARCHITECTURE : APPLICATION FRAMEWORK



Les composants avec lesquelles vos applications interagissent, parmi les plus importants :

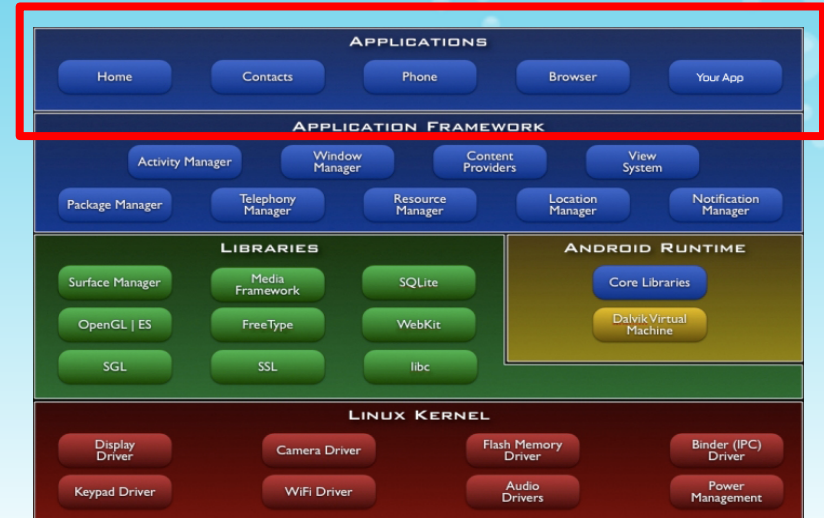
- Activity Manager : Cycle de vie de l'application ;
- Content Providers : Données partagées entre applications ;
- Telephony Manager : Gestion des appels téléphoniques ;
- Location Manager : Gestion de la localisation (GPS) ;
- Resource Manager : Gestion des ressources des applications.

# 9. ARCHITECTURE : APPLICATIONS

Les applications standard

- SMS client app
- Dialer
- Web browser
- Contact manager

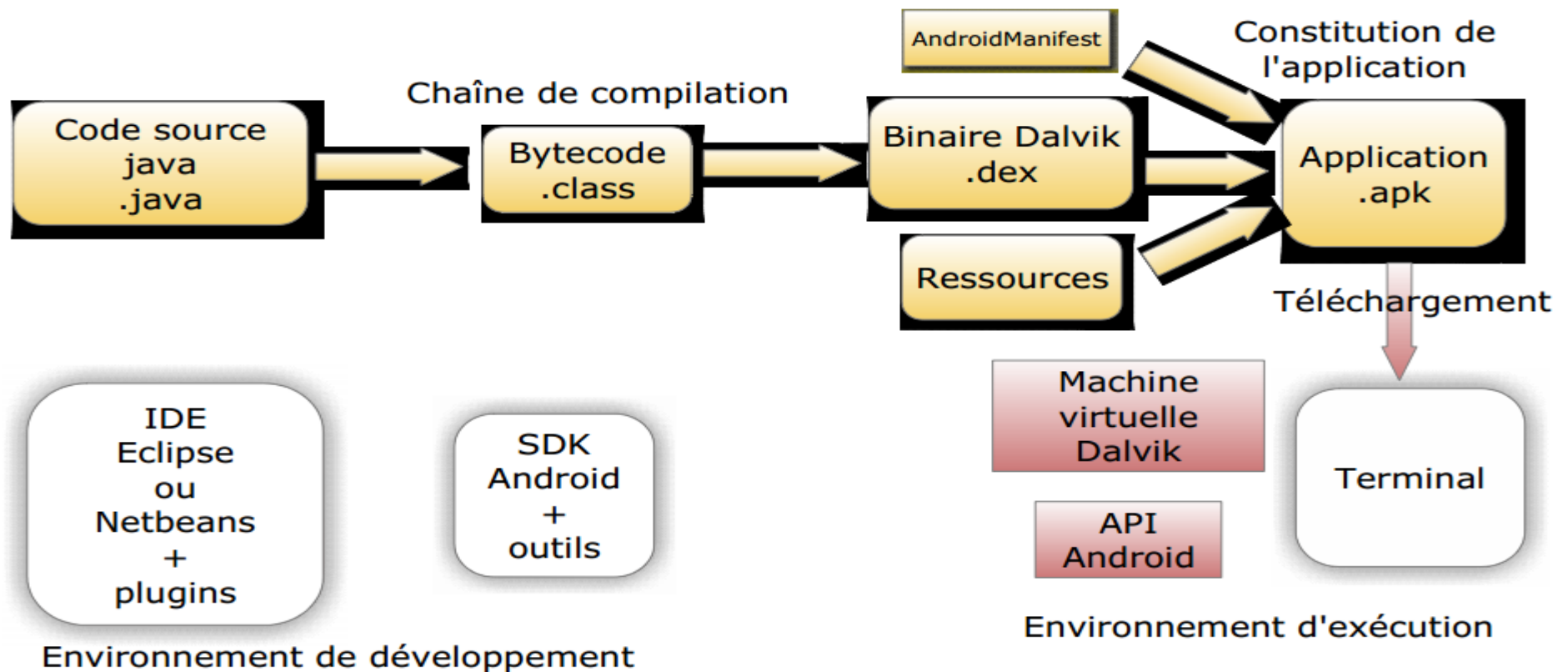
+ Les applications que vous allez développer !



# 10. CYCLE DE DEVELOPPEMENT

Source : <http://www.iutbayonne.univ-pau.fr/~dalmau/documents/cours/android/Cours%20Android.pdf>

## Production de logiciel



# 11. DEVELOPPEMENT : SDK

Télécharger le SDK

<http://developer.android.com/sdk/index.html>

Possibilité de télécharger le SDK seul (stand-alone) ou packagé avec Eclipse.

Propose le SDKManager qui permet de télécharger les plateformes et outils :

- Android versions xx
- Google API versions xx
- Outils (tools et platform-tools)
- etc

Propose le AVDManager pour l'émulation des périphériques mobiles.

Nom	Modifié le	Type	Taille
add-ons	26/10/2014 12:54	Dossier de fichiers	
build-tools	26/10/2014 12:02	Dossier de fichiers	
docs	26/10/2014 12:09	Dossier de fichiers	
extras	26/10/2014 12:14	Dossier de fichiers	
platforms	26/10/2014 12:13	Dossier de fichiers	
platform-tools	26/10/2014 12:02	Dossier de fichiers	
sources	26/10/2014 12:14	Dossier de fichiers	
system-images	26/10/2014 12:23	Dossier de fichiers	
temp	26/10/2014 12:54	Dossier de fichiers	
tools	26/10/2014 12:17	Dossier de fichiers	
AVD Manager.exe	02/07/2014 21:03	Application	216 Ko
SDK Manager.exe	02/07/2014 21:03	Application	216 Ko
SDK Readme.txt	02/07/2014 21:03	Document texte	2 Ko
uninstall.exe	26/10/2014 11:58	Application	70 Ko

# 11. DEVELOPPEMENT : OUTILS SDK

Accessibles à partir d'une ligne de commande (fenêtre DOS) :

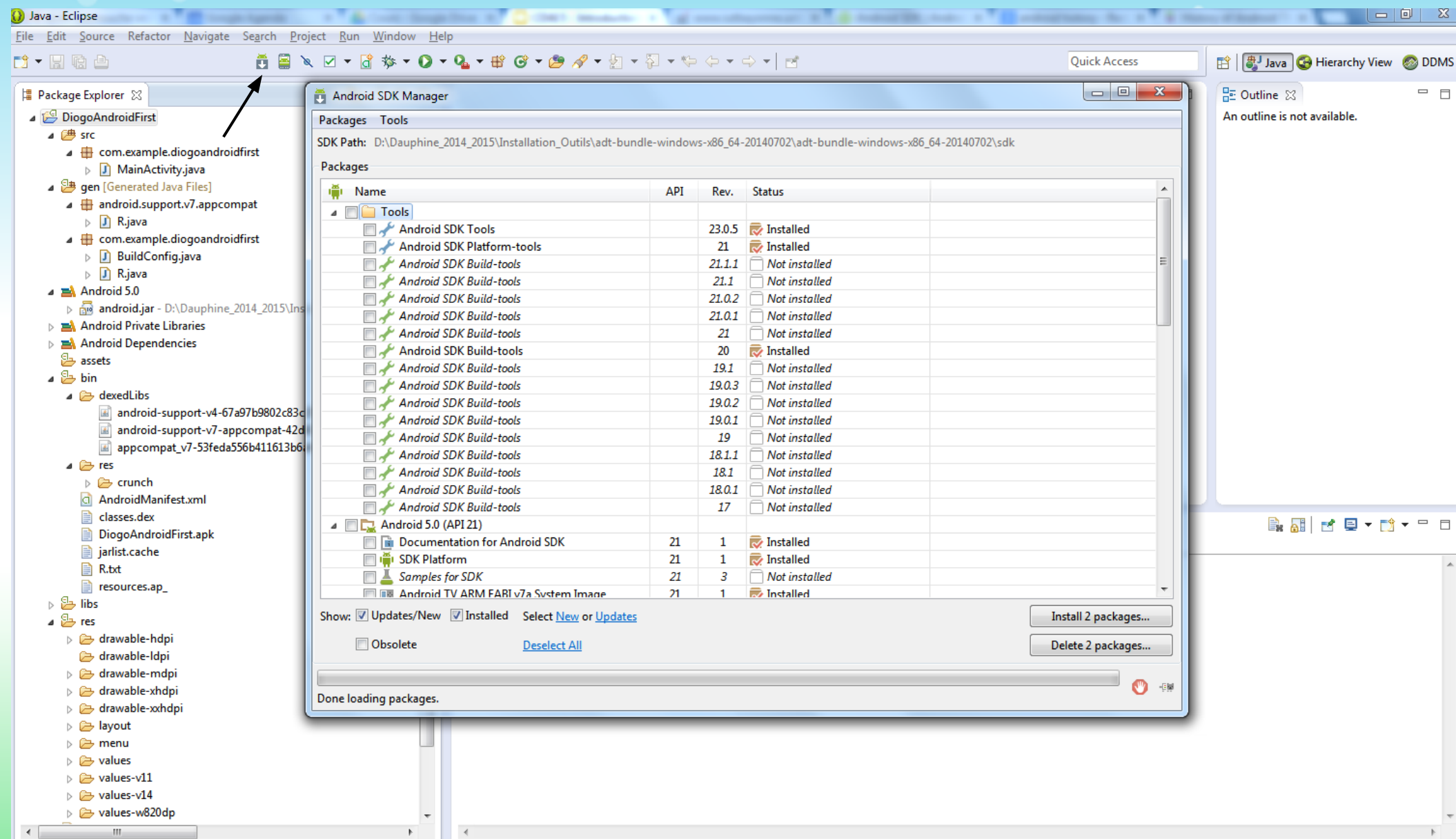
- **adb** permet la connexion au terminal (smartphone ou simulateur) pour :
  - Transférer des fichiers (push / pull)
  - Travailler en ligne de commande unix (shell)
  - Installer une application (install)
  - Paramétrer le réseau (forward)
  - Déboguer une application (logcat)
- **dx** transforme le bytecode java en code Dalvik
- **apkbuilder** constitue un fichier .apk téléchargeable sur le terminal

Remarque : Eclipse utilise ces outils directement mais ...



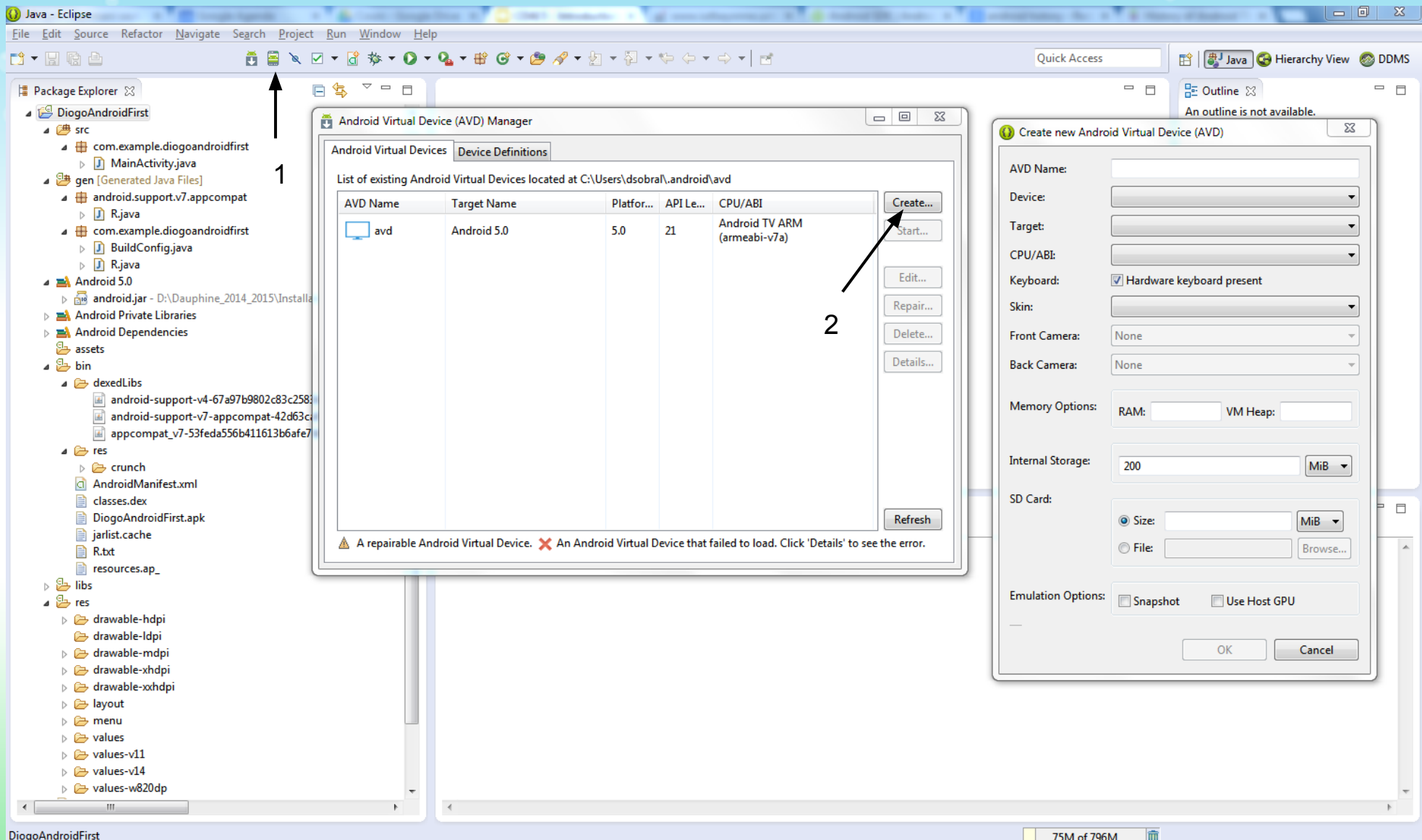
# 11. DEVELOPPEMENT : DEMO

## ECLIPSE -> SDK



# 11. DEVELOPPEMENT : DEMO

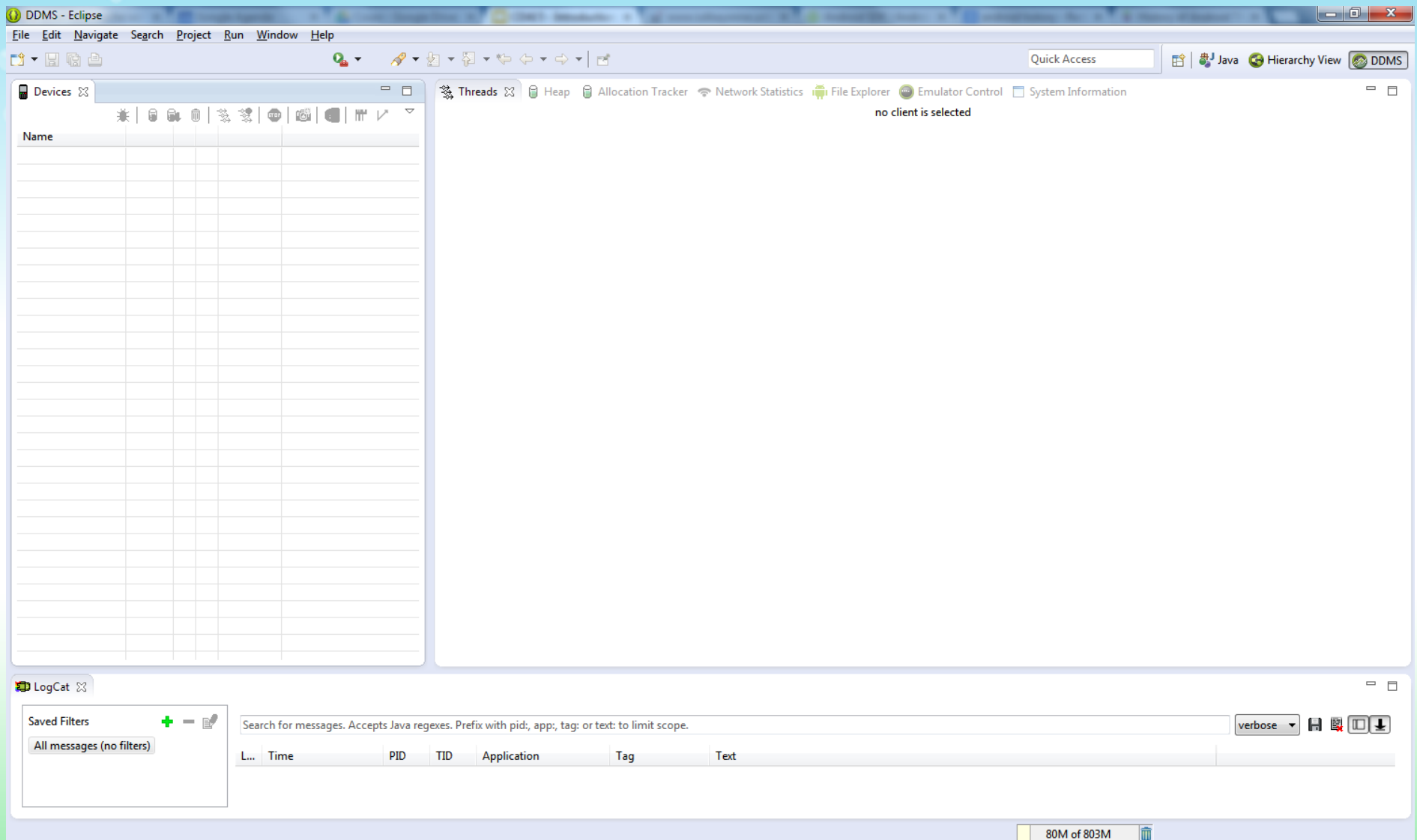
## ECLIPSE -> AVD





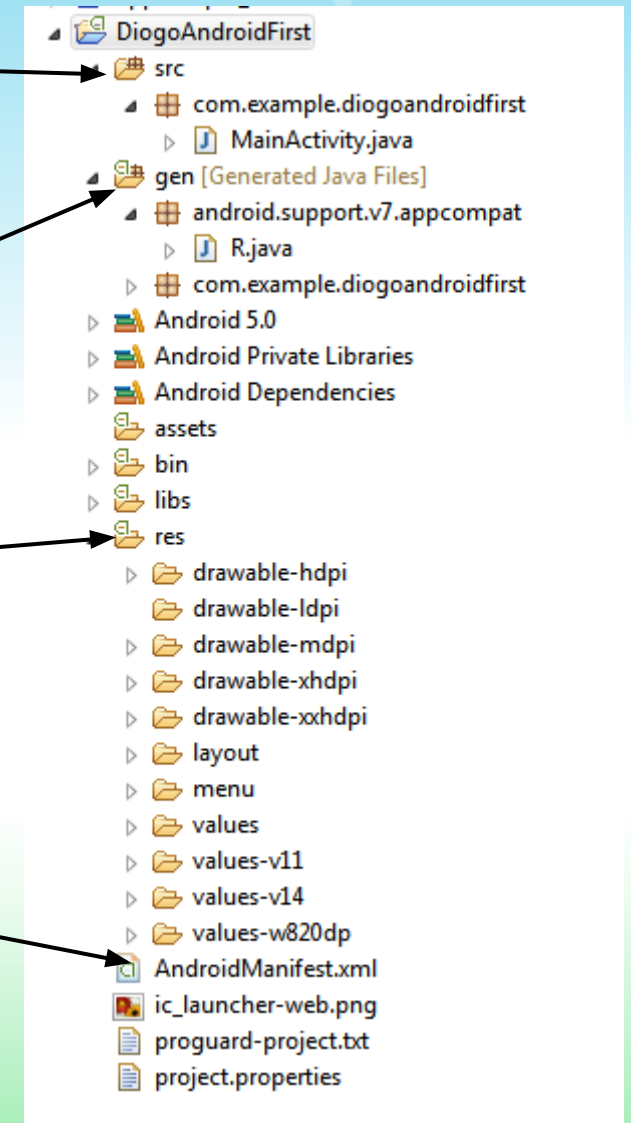
# 11. DEVELOPPEMENT : DEMO

## ECLIPSE -> DDMS (Dalvik Debug Monitor Service)



# 11. DEVELOPPEMENT : STRUCTURE D'UN PROJET

- **src** contient les fichiers source
- **gen** contient les fichiers générés par Eclipse
- **res** contient les ressources (images, descriptions d'interfaces, valeurs).  
Organisé en sous dossiers par types
- **AndroidManifest** décrit les propriétés de l'application



# 11. DEVELOPPEMENT :

## AndroidManifest

### Manifest

- Nom du paquetage
- Versions
- SDK min et target

### • Application

- Nom
- Icône
- Éléments constituant l'application (activités, services, ...)

### • Permissions

### • Instrumentation (pour les tests)

### • Autres

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".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>
```

# 11. DEVELOPPEMENT : RESSOURCES

Application embarquée

⇒ tout doit être dans le fichier .apk téléchargé

• Le répertoire res contient toutes les ressources qui seront mises dans le apk :

- drawable-hdpi (images en haute définition)
- drawable-ldpi (images en basse définition)
- drawable-mdpi (images en moyenne définition)
- layout (description en XML des interfaces)
- values (définitions en XML de constantes : chaînes, tableaux, valeurs numériques ...)
- anim (description en XML d'animations)
- menus (description en XML de menus pour l'application)
- xml (fichiers XML utilisés directement par l'application)
- raw (tous les autres types de ressources : sons, vidéos, ...)

On peut ajouter d'autres sous répertoires

# 11. DEVELOPPEMENT : RESSOURCES

## CLASSE R

La classe R :

C'est une classe générée par Eclipse qui permet à l'application d'accéder aux ressources

- Elle contient des classes internes dont les noms correspondent aux types de ressources (id, drawable, layout ...)
- Elle est constituée à partir des fichiers placés dans les sous répertoires du répertoire res
- Une propriété est créée pour :
  - Chaque image placé dans drawable-xxxx
  - Chaque identificateur défini dans des fichiers XML (objets d'interface, constantes)
  - Chaque fichier placé dans les répertoires xml , raw ...

# 11. DEVELOPPEMENT : UTILISATION RESSOURCES

- Référencement d'une ressource dans un fichier xml.

La forme générale est : "@type/identificateur"

Par exemple : @string/machaine

Fait référence à une chaîne contenue dans un fichier XML placé dans le répertoire res/values et définie comme suit :

```
<resources>
```

```
...
```

```
<string name="machaine">Contenu de cette chaîne</string>
```

```
...
```

```
</resources>
```

- Référencement d'une ressource dans le code.

La forme générale est : R.type.identificateur

Par exemple : R.string.machaine

Fait référence à la même chaîne

# 11. DEVELOPPEMENT : COMPOSANTS APPLICATION

- **Activité** (android.app.Activity)  
Programme qui gère une interface graphique
- **Service** (android.app.Service)  
Programme qui fonctionne en tâche de fond sans interface
- **Fournisseur de contenu** (android.content.ContentProvider)  
Partage d'informations entre applications
- **Écouteur d'intention diffusées** (android.content.BroadcastReceiver) :  
Permet à une application de récupérer des informations générales (réception d'un SMS, batterie faible, ...)
- **Éléments d'interaction**
  - Intention (android.content.Intent) : permet à une application d'indiquer ce qu'elle sait faire ou de chercher un savoir-faire
  - Filtre d'intentions (<intent-filter> : permet de choisir la meilleure application pour assurer un savoir-faire)

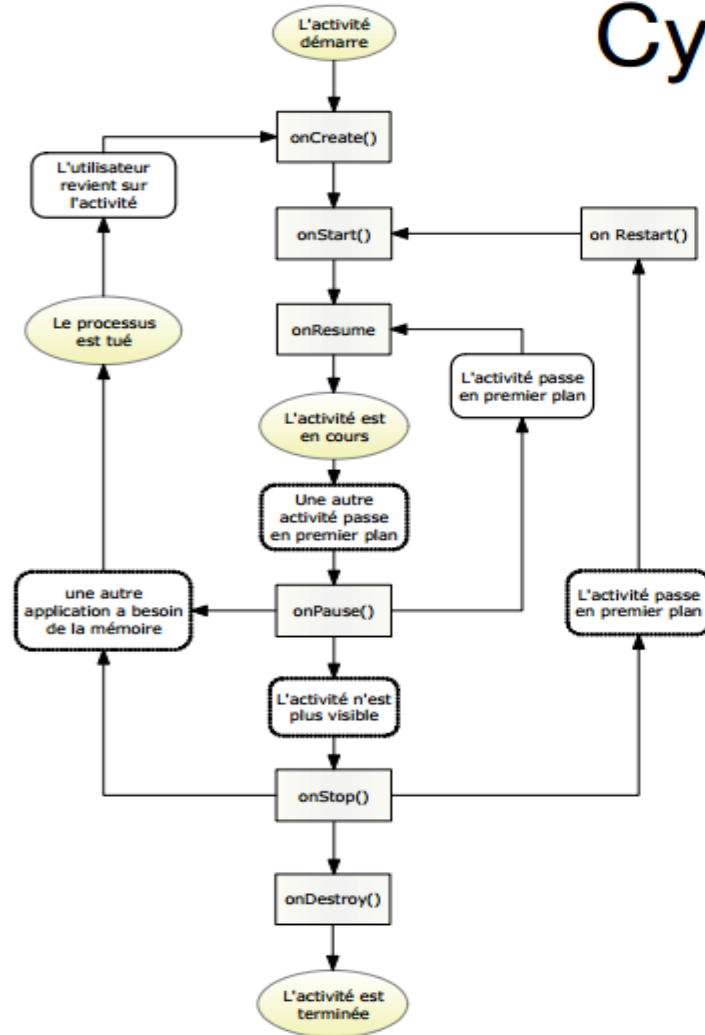
# 11. DEVELOPPEMENT : APPLICATION

- Une activité = un programme avec une interface
- Un service = un programme sans interface
- Une application =
  - Une activité principale
  - Éventuellement une ou plusieurs activités secondaires
  - Éventuellement un ou plusieurs services
  - Éventuellement un ou plusieurs écouteurs d'intentions diffusées
  - Éventuellement un ou plusieurs fournisseurs de contenu



# 11. DEVELOPPEMENT : APPLICATION ACTIVITE

## Cycle de vie d'une activité



- Etats principaux :
  - Active  
Après l'exécution de `onResume`
  - Suspendue  
Après l'exécution de `onPause`
  - Arrêtée  
Après l'exécution de `onStop`
  - Terminée  
Après l'exécution de `onDestroy`

# 11. DEVELOPPEMENT : POUR ALLER PLUS LOIN

DEFINIR UNE INTERFACE (MENUS, COULEURS, CONTENEURS)

GESTION DES EVENEMENTS

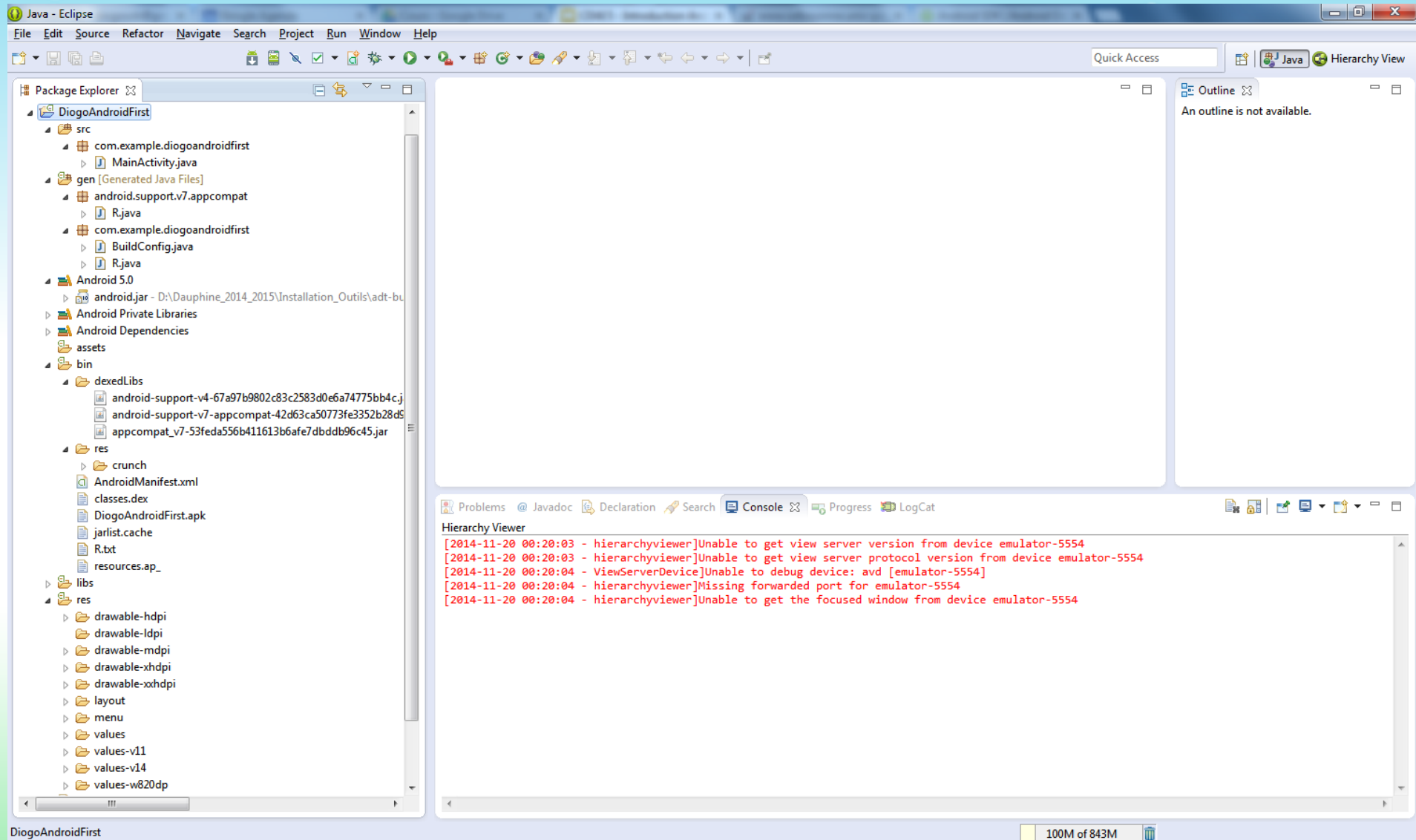
NAVIGATION et INTENTS

PERMISSIONS

PERSISTANCE

MATERIEL ET CAPTEURS

# 11. DEVELOPPEMENT : DEMO ECLIPSE



# 12. CONCLUSION

Téléphonie :

- domaine très dynamique, rentable, innovant
- remplace peu à peu les PC

Android a fait partie de la mouvance

Très gros potentiel

# 13. WEBOGRAPHIE

Référence: <http://developer.android.com/index.html>

Cours : <https://developers.google.com/university/courses/>

Statistiques : <http://androiddistribution.com/>