Programmation avec android Cours 1

Hadjila Fethallah

Maître de Conférences au Département d'Informatique

F_hadjila@mail.univ-tlemcen.dz

Introduction

- Une application mobile est un logiciel embarqué destiné à être exécuté sur un appareil ayant
  - Des contraintes d'alimentation (faible énergie)
  - De faible capacités de mémoire (RAM, Flash,Rom)
  - Des capacités de calcul restreintes (horloge moins rapide, jeu d'instruction réduit,…)
- L'application mobile doit adapter son affichage en fonction de la taille de l'écran, sa résolution, orientation… (smartphone, tablette, montre, TV,…)
- Gérer les événements tactiles, rotation,…
- Accéder et manipuler les capteurs(GPS, accéléromètre, Camera,….)

introduction

- Android est une pile de logiciels « open-source » (sous licence apache)incluant un système d'exploitation basé sur le noyau de linux.
- il est destiné à gérer les appareils mobiles (smartphones, tablettes, montres, ….)
- Développé par Open Handset Alliance, menée par Google, et d'autres entreprises, vers la fin de l'année 2007
- la version 1.0 est crée à la fin de l'année 2008

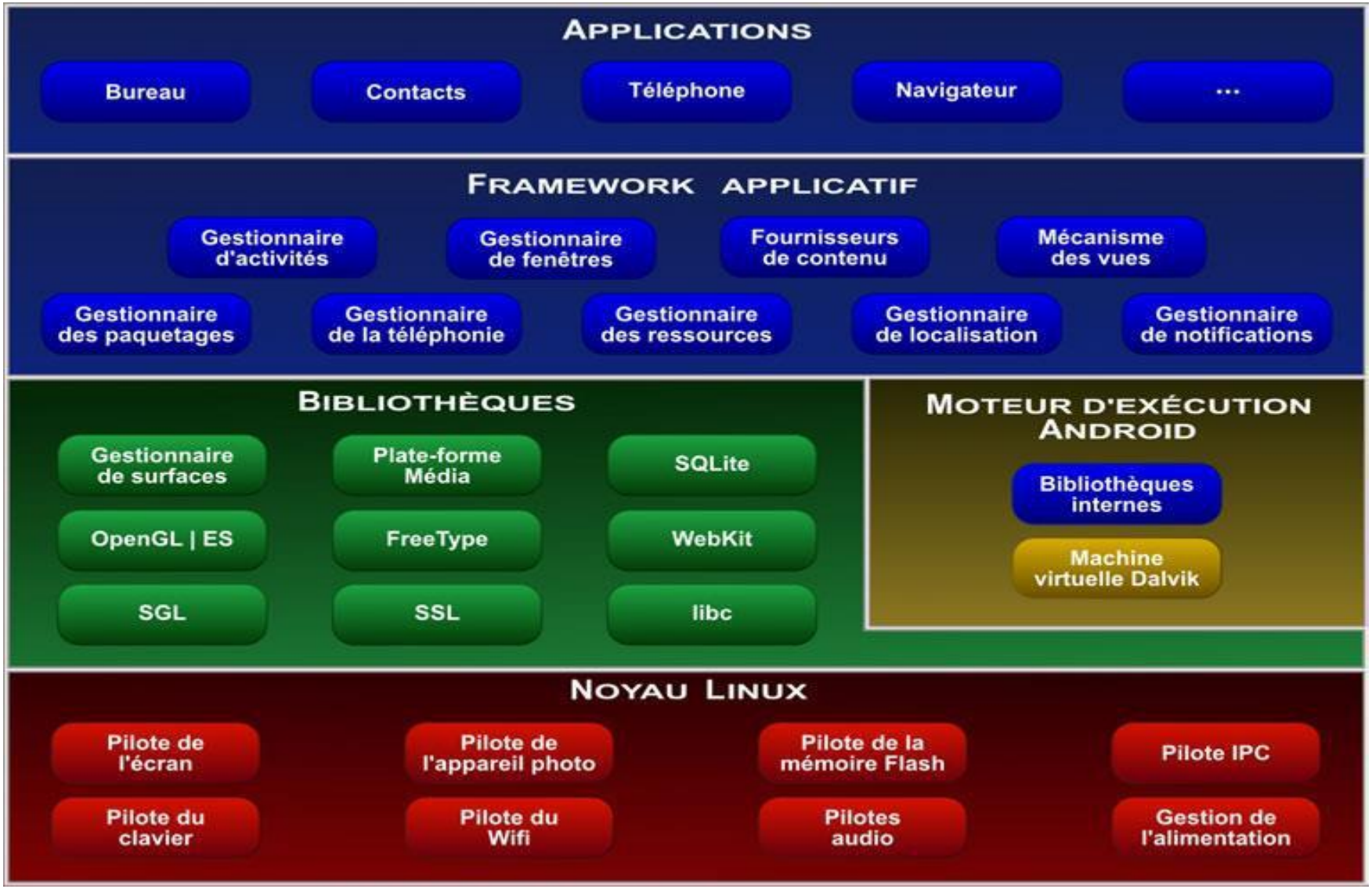
Versions d'android

Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

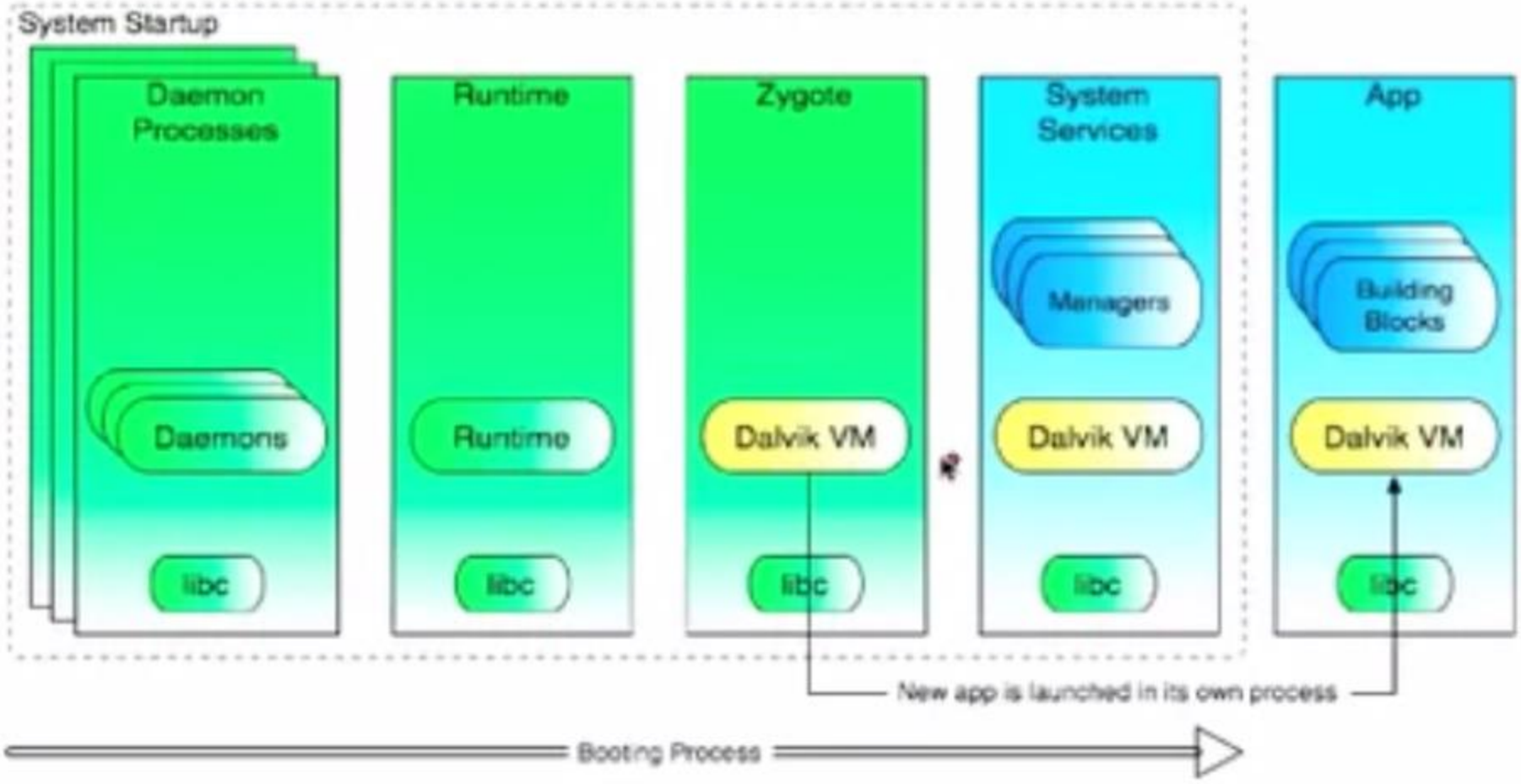
Versions d’android

Platform Version	API Level	VERSION_CODE
Android 8.0	26	Oreo
Android 7.0	24	Nougat
Android 6.0	23	MARSHMALLOW
Android 5.1	22	LOLLIPOP_MR1
Android 5.0	21	LOLLIPOP
Android 4.4W	20	KITKAT_WATCH
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH

Platforme Android



Démarrage d'android



Démarrage d'android

```
root@generic:/ # ps
ps
USER      PID   PPID  VSIZE  RSS     WCHAN    PC         S  NAME
root      1     0     640    496     c00be88c 0001a098 S  /init
root      2     0     0      0       c0033c60 00000000 S  kthreadd
root      3     2     0      0       c001e74c 00000000 S  ksoftirqd/0
root      4     2     0      0       c002f068 00000000 S  kworker/0:0
root      5     2     0      0       c002f068 00000000 S  kworker/u:0
root      6     2     0      0       c002e978 00000000 S  khelper
root      7     2     0      0       c0094d38 00000000 S  sync_supers
root      8     2     0      0       c00953e0 00000000 S  bdi-default
root      9     2     0      0       c002e978 00000000 S  kblockd
root      10    2     0      0       c002e978 00000000 S  rpciod
root      12    2     0      0       c008e784 00000000 S  kswapd0
root      13    2     0      0       c00e428c 00000000 S  fsnotify_mark
root      14    2     0      0       c002e978 00000000 S  crypto
root      25    2     0      0       c021764c 00000000 S  mtdblock0
root      26    2     0      0       c021764c 00000000 S  mtdblock1
root      27    2     0      0       c021764c 00000000 S  mtdblock2
root      29    2     0      0       c002e978 00000000 S  binder
root      30    2     0      0       c002e978 00000000 S  deferwq
root      31    2     0      0       c002f068 00000000 S  kworker/u:2
root      32    2     0      0       c0244d88 00000000 S  mmcqd/0
root      33    1     588    312     c00be88c 0001a098 S  /sbin/ueventd
root      35    2     0      0       c0144430 00000000 S  jbd2/mtdblock0-
root      36    2     0      0       c002e978 00000000 S  ext4-dio-unwrit
root      39    2     0      0       c00d0d8c 00000000 S  flush-31:1
root      41    2     0      0       c0144430 00000000 S  jbd2/mtdblock1-
root      42    2     0      0       c002e978 00000000 S  ext4-dio-unwrit
root      44    1     1428   140     c00e7644 0001120c S  /sbin/healthd
system    45    1     1000   340     c025622c b6f2f41c S  /system/bin/servicemanage
r
root      46    1     4660   1172    ffffffff b6ebdd14 S  /system/bin/vold
root      48    1     9784   1276    ffffffff b6ee4d14 S  /system/bin/netd
root      49    1     2972   2468    c02653fc b6f6c110 S  /system/bin/debuggerd
radio     50    1     5500   856     ffffffff b6efcd14 S  /system/bin/rild
system    51    1     38392  2292    ffffffff b6f525cc S  /system/bin/surfaceflinge
r
root      52    1     202708 39816   ffffffff b6f8f568 S  zygote
drm       53    1     6924   2524    ffffffff b6e3f41c S  /system/bin/drmserver
media     54    1     20500  5432    ffffffff b6ec141c S  /system/bin/mediaserver
install   55    1     988    452     c02f9e5c b6f0b158 S  /system/bin/installd
keystore  56    1     3340   1204    c025622c b6ec741c S  /system/bin/keystore
root      57    1     920    364     c00e7644 b6ede5cc S  /system/bin/qemud
shell     60    1     924    464     c01ec2e0 b6f56158 S  /system/bin/sh
root      61    1     4836   212     ffffffff 000190ac S  /sbin/adbd
root      307   2     0      0       c002f068 00000000 S  kworker/0:2
system    381   52    268384 41012   ffffffff b6f905cc S  system_server
media_rw  471   1     3508   456     ffffffff b6f09158 S  /system/bin/sdcard
root      503   2     0      0       c00d0d8c 00000000 S  flush-179:0
radio     531   52    235180 25900   ffffffff b6f905cc S  com.android.phone
u0_a8     543   52    227468 30812   ffffffff b6f905cc S  com.android.launcher
u0_a39    563   52    212880 17552   ffffffff b6f905cc S  com.android.printspooler
u0_a2     598   52    217728 23028   ffffffff b6f905cc S  android.process.acore
u0_a7     634   52    227776 36468   ffffffff b6f905cc S  com.android.systemui
u0_a29    673   52    221440 24752   ffffffff b6f905cc S  com.android.inputmethod.l
atin
```

Platforme Android

- **Linux kernel:**
- contient les pilotes du matériel, offre la gestion de la mémoire/processus/l'alimentation/IPC
- la première version adoptée est 2.6
- **Librairies natives:** permettant par exemple la gestion des BDD, l'affichage des objets graphiques 2D ou 3D, l'affichage des pages web, le chiffrement, le décodage des formats audio/video …
- **Machine Virtuelle Dalvik :**une JVM optimisée pour les systèmes embarqués
- **bibliotheques internes (core Lib):** les classes java basiques ou specifies à android (java.*, javax.*,android.app.*, android.graphic.* android.view, android.os,..
- **Applications clés:** des classes java (services) réutilisables par la majorité des applications utilisateurs
- **applications** : les programmes manipulés par l'utilisateur final

middleware

- Moteur WebKit
- Libc (Bionic)
- SQLite : SGBD
- librairies pour lire/ enregistrer les fichiers audio/video
- SSL : envoi de messages sécurisés
- Open GL :bibliothèque de graphisme 2D et 3D
- freeType: gérer l'affichage du texte sur les images BitMap
- HAL: des interfaces et des pilotes pour d'autres types de capteurs/matériels

Applications clés

- **Activity Manager** : Contrôle le cycle de vie des activités et la pile des activités.
- **Content Providers :** permet le partage des données entre activités.
- **Resource Manager** : permet l'accès aux ressources (strings, couleurs, disposition d'interfaces ou layouts).
- **Notifications Manager** – permet l'affichage de messages d'alertes /notifications aux utilisateurs.
- **View System** – un ensemble de vues (UI) utilisables par les applications finales.
- **Package Manager** – permet aux applications de connaitre des informations sur toutes les applications installées sur l'appareil (favorise la coopération)
- **Telephony Manager** : fournit aux applications des informations sur les services de téléphonie disponibles sur l'appareil (ex: état)

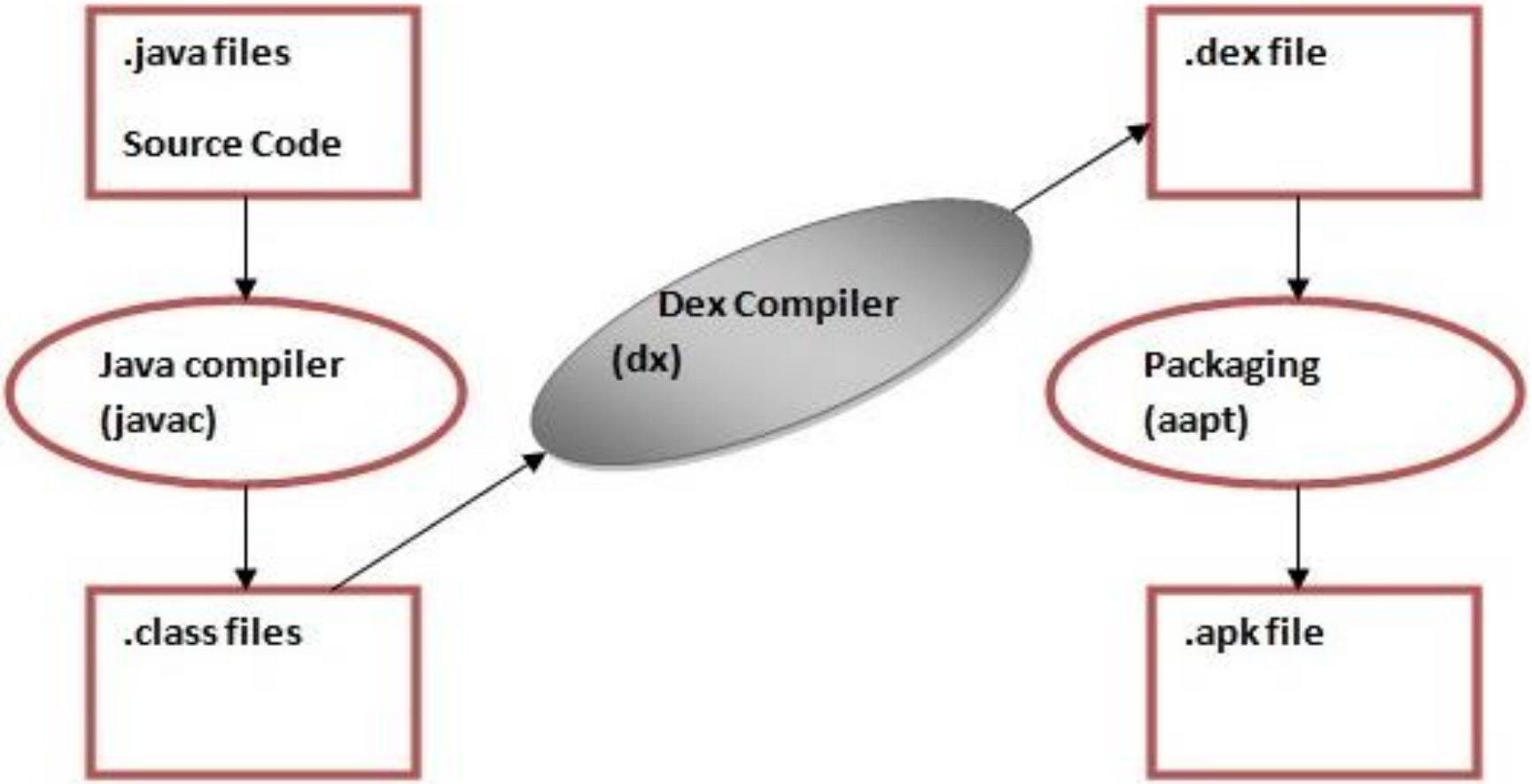
Programmation avec android

- Android Development Tools Bundle :
  - C’est environnement contenant:
    - IDE : éclipse
    - ADT plugin
    - SDK (software development kit)
  - Sous Licence éclipse (libre)
- Android studio
  - Développé par IntelliJ
  - L’IDE officiel de Google
  - Sous licence apache (libre)

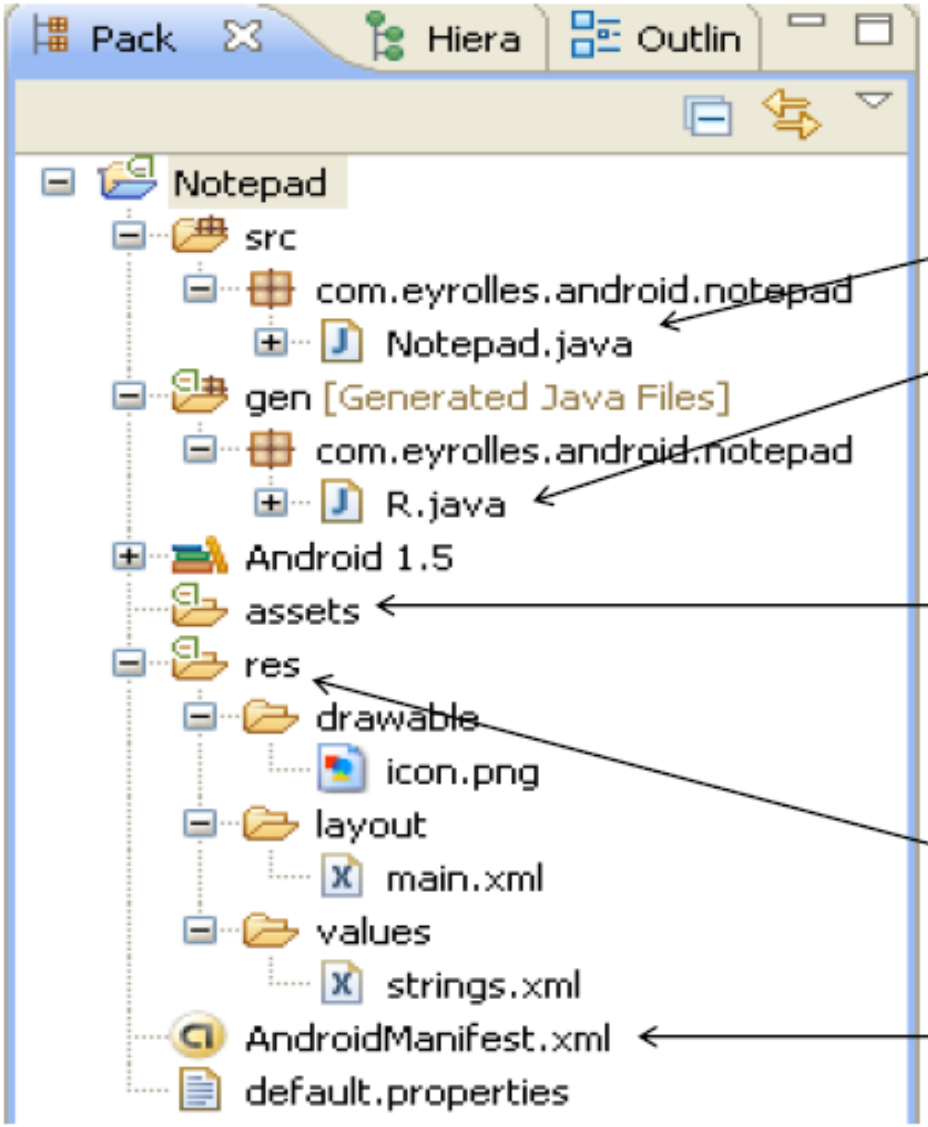
Eléments du SDK

- **Plateformes android :** plusieurs versions ou api level sous forme de “android.jar”
- **Doc et exemples / plateforme**
- **Sdk Tools:** contient un debogueur DDMS, Virtual device manager, emulateur, mksdcard, traceview, hierarchyview, logcat, sqlite3
- **Sdk platform-tools:** Android Debug Bridge, aapt
- **Build tools**: dx, apksigner, zipalign, jobb qui crypte les fichiers d’extension d’apk.
- **Image système :** pour processeur x86, X86_64,ARM, MIPS
- **Google API**: manipulation des MAP
- **Code source pour SDK**

Processus de génération du APK



Structure d'un projet android



Code java du développeur

Code généré automatiquement

Ressources brutes

Ressources: images/
chaines de caractères/ dispositions

Document décrivant l'application et
Ses composantes

Ressources

- Values/
  - Chaines de caractère
  - Tableaux
  - Dimension : la taille des marges/espacement
  - Couleurs
  - ID…
- Drawable/ : images de type gif,png,jpg, ou fichier xml
- Layout/ : description de l'interface graphique
- menu/ : description des menus
- Raw/:fichiers brutes accessible grace *Resources.openRawResource()*+ ID de la ressource, elles sont empaquetées sans aucun traitement
- Xml/ : contient les fichiers XML supplémentaires
- anim/ : décrit les propriétés des animations

String

- Mis dans res/values/strings.xml
- **<resources>**
- **<string** name= "b1_text"**>** ok **</string>**
- **<string** name="app_name"**>** essai **</string>**
- **<string** name="hello"**>** bonjour **</string>**
- **</resources>**

Tableau de chaines de caractères

- **<resources>**
- **<string-array** name="test"**>**
- **<item>**it1**</item>**
- **<item>**it2**</item>**
- **</string-array>**
- **</resources>**

Dimensions

- Mis dans res/values/dimens.xml
- <resources >
- < dimen name = " activity _ horizontal_margin " > 16dp < /dimen >
- <dimen name ="activity _ vertical _margin “>16dp < /dimen >
- <dimen name="button_height">48dp</dimen>
- <dimen name="title_size">32sp</dimen>
- < /resources >
- **Acces en XML:**
- android:layout_width="@dimen/button_height"

Constantes de type couleurs

- Mis dans res/values/colors.xml
- <resources> <color name=“rouge_opaque">#f00</color> <color name=“rouge_transparent">#80ff0000</color> </resources>
- **Utilisation:**
- Format #RGB ou #AARRGGBB
- **Accés en XML:**
- android:textColor="@color/ rouge_transparent
- **Accés en Java:**
- Resources res = getResources(); int color = res.getColor(R.color.rouge_opaque);

Exemple:drawable

- <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height=“match_parent" android:orientation="vertical" >
- <ImageView android:layout_height="wrap_content" android:layout_width="wrap_content" android:src="@drawable/image1" />
- …….
- </LinearLayout>

Identifiants

- Mis dans res/values/ids.xml
- <resources> <item type="id" name="button_ok" /> <item type="id" name="textview1" /> </resources>
- **Acces en XML:**
- <Button android:id="@**id/button_ok**" android:layout_width="wrap_content" />

Classe R

- Classe auto-génerée:
- Contient les IDs des ressources du projet
- On utilise findViewById ou getResources pour accéder à ces ressources
- **Exemple:**
- Button b = (Button)findViewById(R.id.b1)
- String s= getResources().getString(R.string.*hello*));

Classe R

- package com.example.tp2;
- public final class R {
- public static final class dimen {
- public static final int padding_large=0x7f040002;
- public static final int padding_medium=0x7f040001;
- public static final int padding_small=0x7f040000; }
- public static final class drawable {

- public static final int ic_action_search=0x7f020000;
- public static final int ic_launcher=0x7f020001; }

Classe R

- public static final class id {
- public static final int menu_settings=0x7f080000; }
- public static final class layout {
- public static final int activity_main=0x7f030000; }
- public static final class string {

- public static final int app_name=0x7f050000;
- public static final int hello_world=0x7f050001;
- public static final int menu_settings=0x7f050002;
- public static final int title_activity_main=0x7f050003;
  } }

Principaux composants d'une application mobile

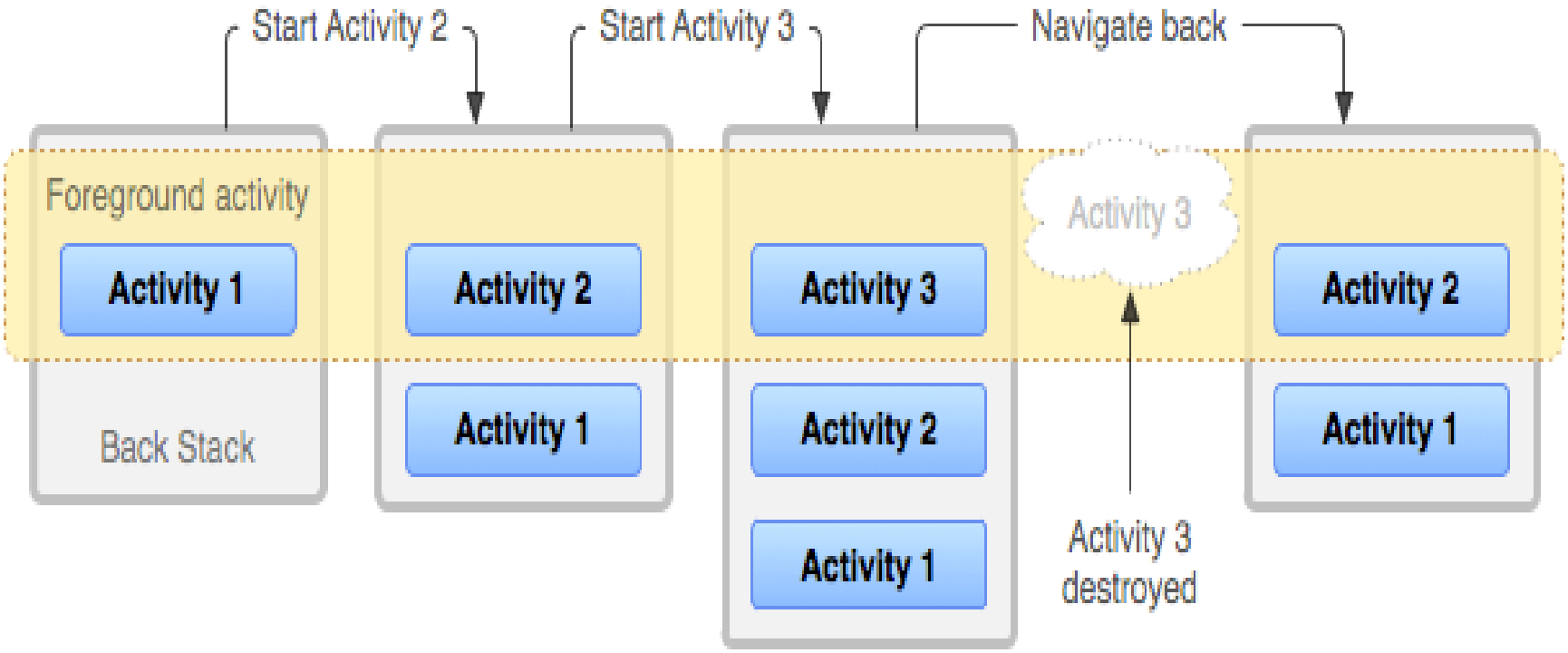
Composant	**Description**
Activities	Affiche l'IU et gère l'interaction avec l'utilisateur
Services	Exécute un processus en arrière plan associé avec l'application
Broadcast Receivers	Gere la communication entre Android et les applications
Content Providers.	Gere les bases de données

Classe Activity

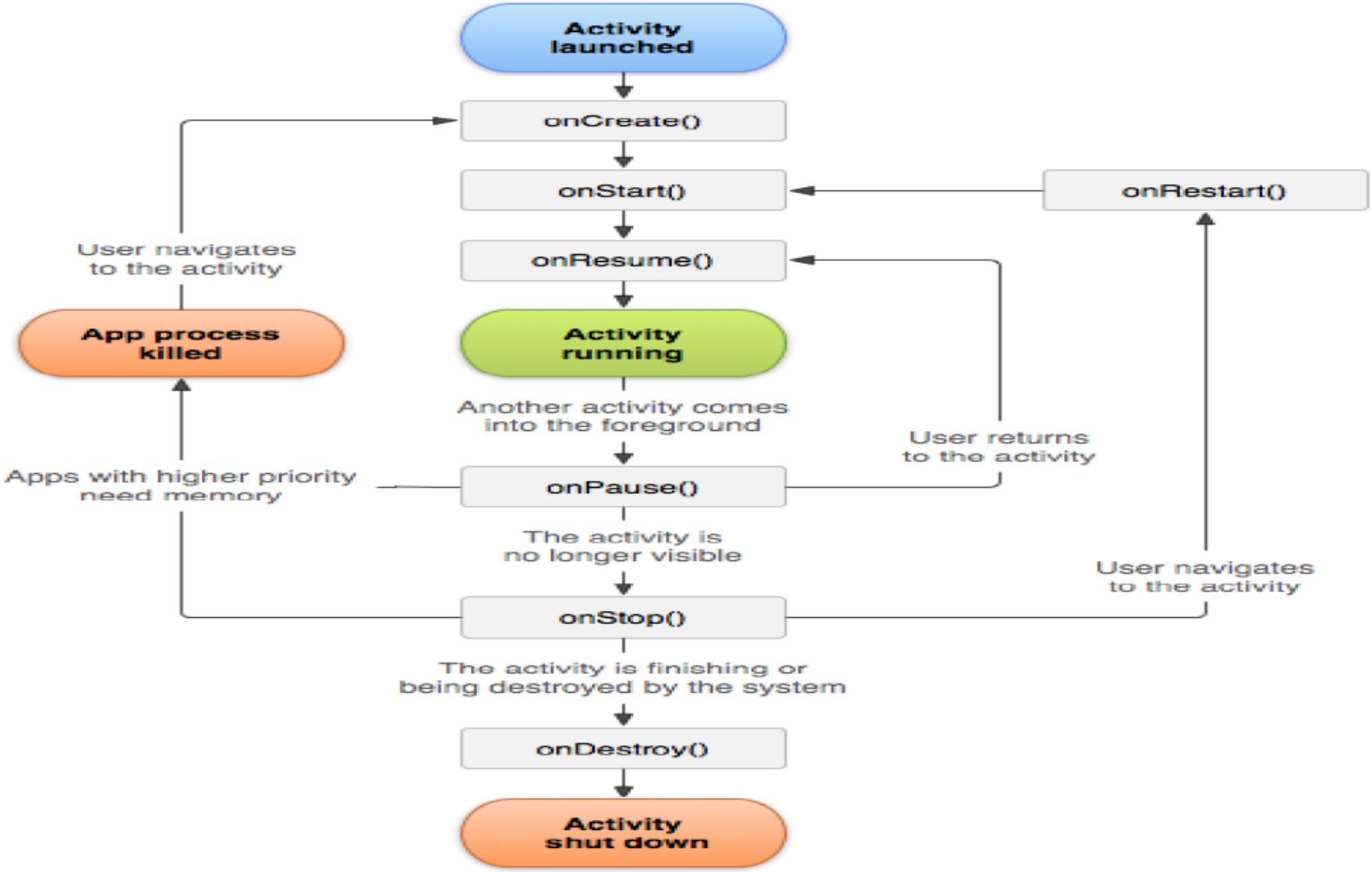
- Une *activité* est une classe affichant un rectangle occupant tout l'écran, et contenant un ensemble de vues (UI Controls) organisées selon une disposition prédéfinie (layout).
- Elle est aussi responsable sur la définition des écouteurs

Notion de tâche

- Collection d'activités appartenant éventuellement à des applications différentes



Classe Activity



Actions typiques

- OnCreate():
- S'éxecute une seule fois dans la durée de vie de l'activité
- Parmi ses responsabilités:
  - Initialiser les données des listes (partie modèle)
  - Lancer les threads (de type Arrière-plan)
  - Instancier quelques variables
  - Récupérer l'état précédent de l'activité (bundle)
- onStart() autoriser la visibilité
- elle rend l'activité visible, en initialisant l'UI
- Eventuellement elle active un broadcastreceiver

Actions typiques

- onResume() → autoriser l'interaction
- Une activité entre en état « resumed» et invoque onResume(), après un évènement lançant l'application en question.
- Parmi ses responsabilités:
- initialiser les ressources utilisées (telles que les capteurs GPS, Camera, Broadcastreceivers..)
- Lancer les opérations d'avant plan (animation, lecture de vidéo/audio)
- L'activité sort de l'état « Resumed » après des évènements tels que: la navigation vers une autre activité, mise en veille, lancement d'un processus prioritaire (appel téléphonique)…

Actions typiques

- OnStop()→(blocage de l'interaction + invisibilité totale)
- une activité devient invisible, et entre en état « stopped» suite à des évènements tels que (l'affichage d'une autre activité/ fin ordinaire)
- Libération de (presque) toutes les ressources
- On doit sauvegarder l'état des données
- L'etat des vues est stocké dans un objet bundle grâce à onSaveInstanceState() (après OnResume, et durant ou avant OnStop)
- Exécution d'actions garantissant la cohérence de l'app
- L' objet « activity » reste en mémoire (pour une éventuelle reprise), sauf en cas de pénurie de mémoire

Actions typiques

- OnDestroy() →(le processus est tué par android)
- Executée Lorsque:
- l'activité exécute finish(),
- android a besoin de mémoire
- Appui du bouton retour
- Apres changement d'orientation appel immédiat de oncreate()
- La libération de toutes les ressources qui n'étaient pas libérées au niveau de onstop()
- En cas de besoin de mémoire, android tue le processus (avec toutes ses activités)

Destruction d'un processus

% de tuer un processus	Etat du Processus	Etat de l'activité
petit	Foreground (ayant, ou voulant avoir le focus)	Crée Démarrée Poursuivie (Resumed)
moyen	Background (focus perdu)	suspendue (Paused)
élevé	Background (invisible)	arrêtée (Stopped)
	vide	Détruite (Destroyed)

exemple

- package com.example.tp1;
- import android.os.Bundle;
- import android.app.Activity;
- public class MainActivity extends Activity {
- public void onCreate(Bundle savedInstanceState) {
- **super.onCreate(savedInstanceState);**
- **setContentView(R.layout.activity_main);**
- **button = (Button) findViewById(R.id.b1);**
  button.setOnClickListener(new OnClickListener()
- **{public void onClick(View arg0) {**
- **……..}**
- **});** }}

« Manifest » d'une application

- le *fichier de configuration* de l'application. C'est un fichier indispensable à chaque application qui décrit entre autres :
- le point d'entrée de votre application (quel code doit être exécuté au démarrage de l'application) ;
- quels composants constituent ce programme ;
- les compétences de chaque activité (affichage de page web, Main, envoi d'SMS,….)
- les permissions nécessaires à l'exécution du programme (accès à Internet, accès à l'appareil photo...)

« Manifest » d'une application

- **<manifest**
  xmlns:android="http://schemas.android.com/apk/res/android"
- package=" com. example.tp1"
- android:versionCode="1"
- android:versionName="1.0">
- <uses-sdk
- android:minSdkVersion="8"
- android:targetSdkVersion="15" />
- **< uses-permission android:name= "android.permission. CALL_PHONE"/>**
- **< uses-permission android:name= "android.permission.SEND_SMS"/>**
- < application android:icon = " @ drawable/icon1 "
- android :label = " @string/app_name " >
- < activity android:name = “.ActivitePrincipale" >

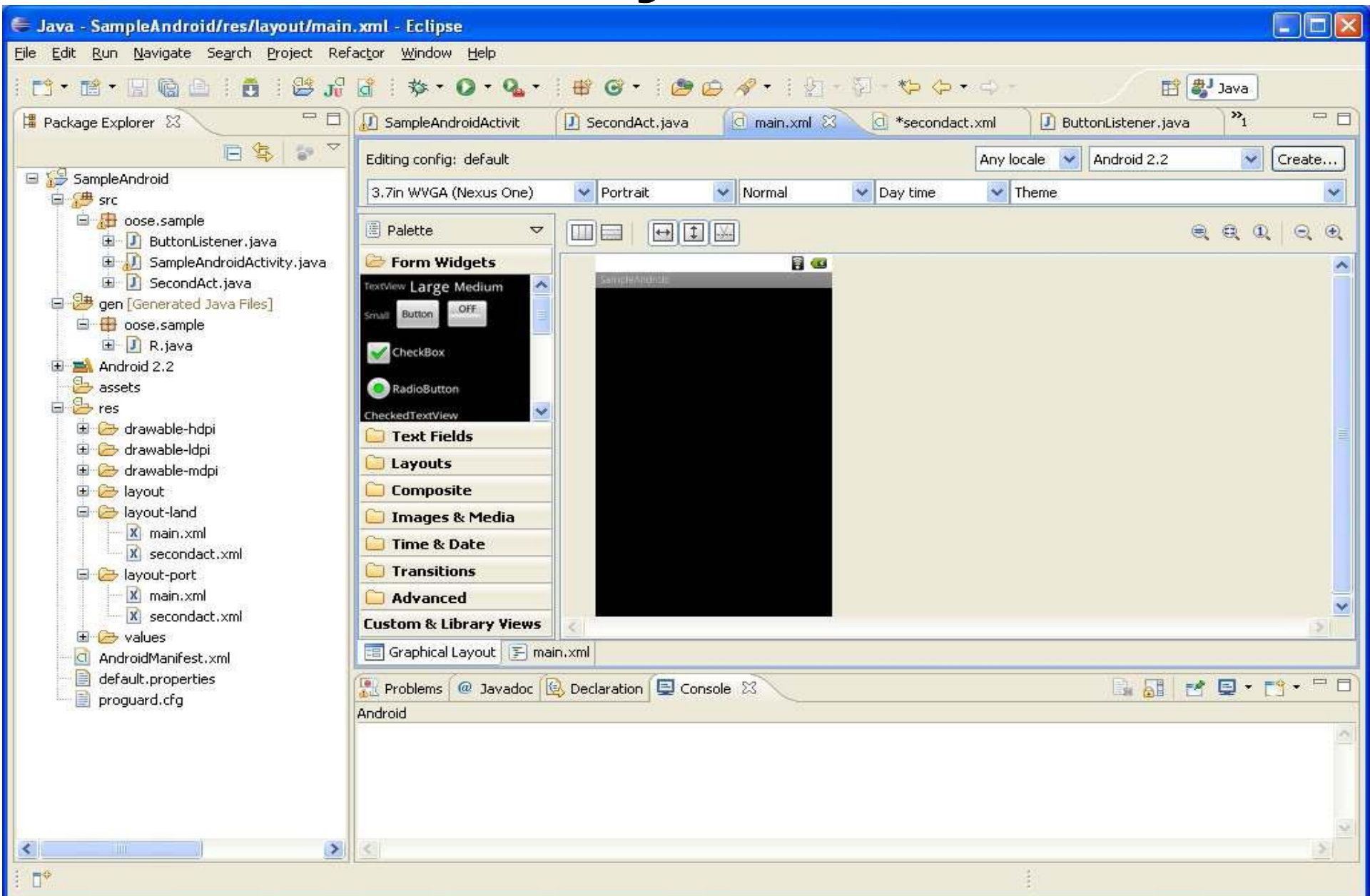
« Manifest » d'une application

- **<intent-filter>**
- **<action android:name="android.intent.action.MAIN" />**
- **<category android:name="android.intent.category.LAUNCHER" />**
- **</intent-filter>**
- </activity>
- <service>...</service>
- <receiver>...</receiver>
- <provider>...</provider>
- </application>
- **</manifest>**

Vues

- Les *vues* sont les éléments de l'interface graphique que l'utilisateur voit et sur lesquels il pourra agir.
- Les vues peuvent être des:
- vues simples(UI Control)
- Botton, TextView, EditText, RatingBar,…
- Vues-groupes (viewGroup): vues invisibles qui rassemblent/organisent d'autres vues
- Layout (gabarit) :lineaire, relative, grid (grille)
- ViewAdapter / Adapter : Gallery, Spinner, ListView
- RadioGroup, TimePicker, DatePicker, WebView,…
- …..

Layout


Java - SampleAndroid/res/layout/main.xml - Eclipse
File Edit Run Navigate Search Project Refactor Window Help
Java
Package Explorer
SampleAndroid
src
oose.sample
ButtonListener.java
SampleAndroidActivity.java
SecondAct.java
gen [Generated Java Files]
oose.sample
R.java
Android 2.2
assets
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
layout
layout-land
main.xml
secondact.xml
layout-port
main.xml
secondact.xml
values
AndroidManifest.xml
default.properties
proguard.cfg
SampleAndroidActivit
SecondAct.java
main.xml
*secondact.xml
ButtonListener.java
Editing config: default
Any locale
Android 2.2
Create...
3.7in WVGA (Nexus One)
Portrait
Normal
Day time
Theme
Palette
Form Widgets
TextView Large Medium
Small Button OFF
CheckBox
RadioButton
CheckedTextView
Text Fields
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Custom & Library Views
Graphical Layout
main.xml
Problems
Javadoc
Declaration
Console
Android

layout

- <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"
- android:paddingLeft="@dimen/activity_horizotal_margin"
- android:paddingRight="@dimen/activity_horizotal_margin"
- android:paddingTop="@dimen/activity_vertical_margin"
- android:paddingBottom="@dimen/activity_vertical_margin">

layout

- <TextView
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:layout_centerHorizontal="true"
- android:layout_centerVertical="true"
- android:padding="@dimen/padding_medium"
- android:text="@string/hello_world"
- android : text Size = "100sp"/>
- **<Button** android:text="Go !"
- android:id="@+id/b1"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"**>**
- **</Button>**
- </RelativeLayout>

FIN du Cours1

Quelques références

- Damien Guignard, Julien Chable, Emmanuel Robles. Programmation Android De la conception au deployment avec le SDK Google Android 2. eyrolles.2010.
- Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano. Android Programming: The Big Nerd Ranch Guide. Big Nerd Ranch, Inc. 2015.
- Marilyn Wolf. Computer as Components Principles of Embedded Computing System Design.Elsevier. 2012.
- John Horton. Android Programming for Beginners. Packt Publishing. 2015.
- J. F. DiMarzio. Programming with Android Studio. John Wiley & Sons, Inc. 2017.