Programmation Android

Ce que vous devez savoir

Plan

1. Connaissances essentielles
2. Définition d'une GUI et adaptation au contexte
3. Cycle de vie d'une activité
4. L'objet Intent

Sur la plate-forme Android

Le Android SDK contient (entre autres)

- les **utilitaires** nécessaires au développement (**tools**) : création de projet, gestion des SDK, débogueur, etc.
- les librairies nécessaires au développement d'application : **APIs**, organisées par version
- le **Android Virtual Device Manager** (création et gestion des émulateurs)

Création d'une application android

Points essentiels → définis dans AndroidManifest.xml

1. **le nom de package** (dans la balise racine : **<manifest . . . />**)
2. les versions min et max d'Android supportées : **<uses-sdk . . . />**.
3. caractéristiques de l'application : balise **<application . . .**
   - nom : **<application . . . android :label="@string/app_name" >**
   - caractéristiques de chaque activité définie :
     - classe : **<activity android :name=".AfficheURL" >**
     - filtre(s) pour les **<intent-filter>**

./AndroidManifest.xml : description

activity_main.xml | main.tex | strings.xml | HelloWorld Manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.iutmontp.helloworld"
    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>
```

Organisation des sources d'un projet

Dossier App

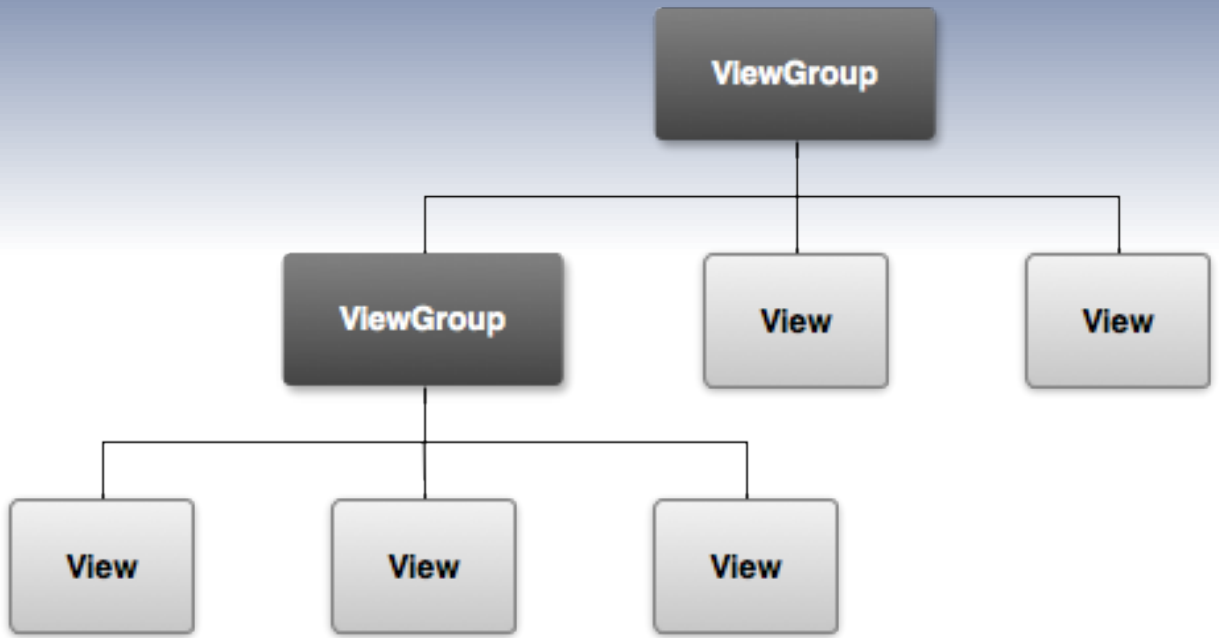
- **build/** : fichier générés
- **libs/** : librairies additionnelles
- **src/ : intégralité des sources pour le développement (appli + tests)**

Contenu du projet

Dossier App/src/

- **androidTest/** : tests de l'application dans l'environnement android
- **test/** : tests internes à l'application
- **main** : sources de l'application
  - **main/java** : sources Java, e.g. la classe qui lance l'activité
  - **main/res** : les ressources de l'application
    - **drawable-(h)(m)(l)dpi/** images dans différentes résolutions
    - **layout/** GUI design général
    - **values/** valeurs des variables
    - **menu/** définition des menus
- **main/AndroidManifest.xml** : description et éléments-clés de l'application (nom, activité principale, intents, etc.)

Organisation d'une GUI Android



- **View** : élément d'interface type *widget* (bouttons, champ texte, etc.)
- **ViewGroup** : un type de **View** contenant d'autres View, gérées par un même gestionnaire de mise en page : positionnement des éléments les uns par rapport aux autres (grille, liste verticale, etc.).

Layouts : gestion de la mise en page

Layout : sous classe de ViewGroup

- Un **Layout** définit la manière dont les **Views** contenues sont disposées les unes par rapport aux autres.
- Des **ViewGroup** standards peuvent être créés avec du code XML

Exemples :

- **RelativeLayout** : chaque **View** définit son déplacement par rapport à une autre **View**
- **LinearLayout** : disposition des éléments en 1 ligne ou 1 colonne dans l'ordre où ils sont définis dans le XML

Adaptaztion de la GUI au contexte

Propriétés d'un écran

- **size** → **small**, **normal**, **large** ou **xlarge**
- **density** → **low** (ldpi), **medium** (mdpi), **high** (hdpi), **extra high** (xhdpi)

Principe et gestion de l'adaptation

- Chaque layout ou bitmap est placé dans un sous répertoire de **res** ayant pour nom la taille ou la résolution correspondantes.
- Note : le changement d'orientation (portrait ou paysage) est considéré comme une modification de la taille de l'écran

Gestion de différents layout

```
MyProject/
    res/
        layout/
            main.xml
        layout-large/
            main.xml
```

Un layout par configuration

- Pour chaque taille à supporter : **un fichier layout de même nom**.
- Chaque configuration est placée dans un sous répertoire de **res** correspondant à la taille : **./res/layout-*<screen_size>***/
  e.g. **./res/layout-large**.
- Par défaut, **layout/** est utilisé pour l'orientation portrait.

Gestion de différentes orientation

```
MyProject/
    res/
        layout/                 # default (portrait)
            main.xml
        layout-land/            # landscape
            main.xml
        layout-large/           # large (portrait)
            main.xml
        layout-large-land/      # large landscape
            main.xml
```

Gestion clique, solution 1 : XML

```
<Button
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="sendMessage"
    android:text="@string/button_send" />
```

- Ajout de l'attribut **android :onClick** à l'élément *Button*
- valeur : méthode définie dans l'activité contenant la *view*
- signature standardisée : ***public void*** et un paramètre de type ***View***

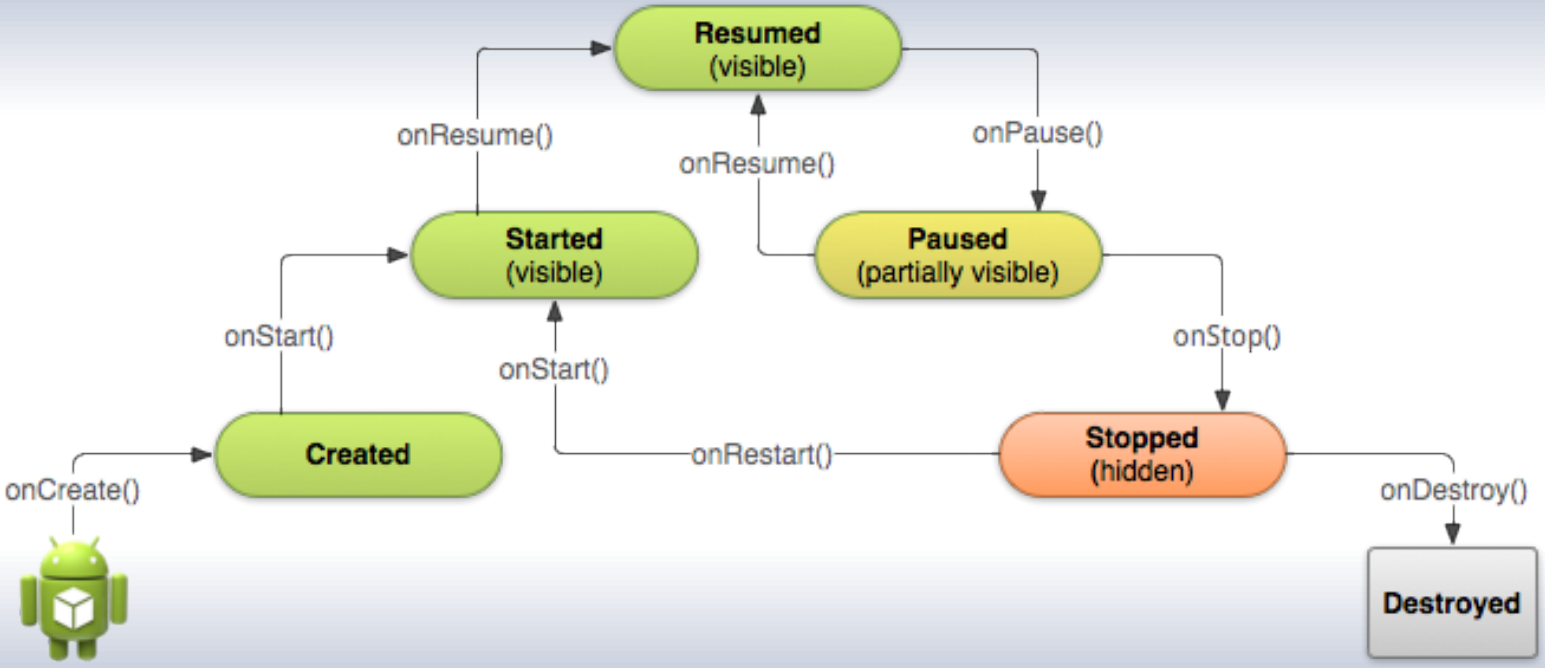
```
/** Called when the user touches the button */
public void sendMessage(View view) {
    // Do something in response to button click
}
```

Gestion clique, solution 2 : dans le code Java

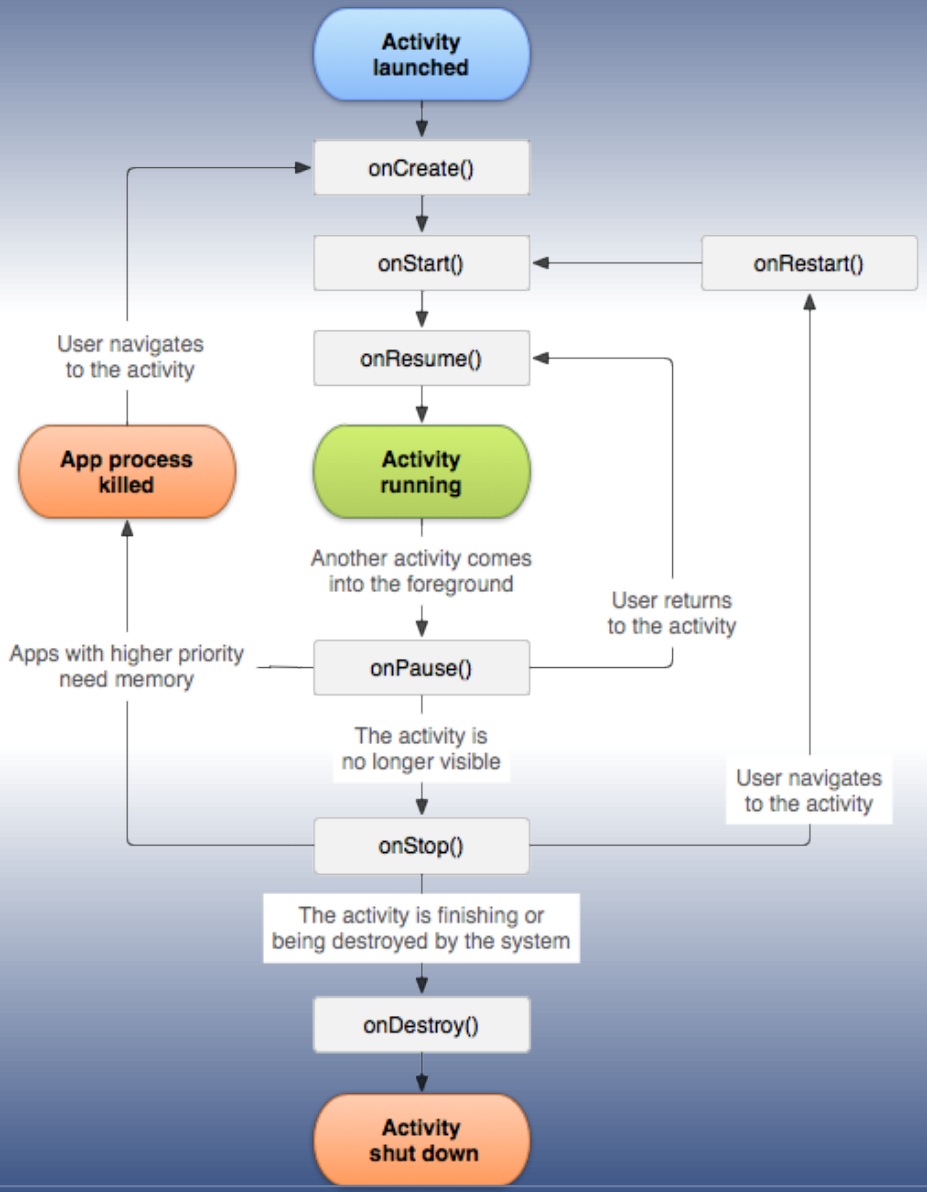
Ajout d'un écouteur (listener) au bouton, par exemple au moment de la création de l'activité :

```
Button button = (Button) findViewById(R.id.button_send);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // Do something in response to button click
    }
});
```

Les méthodes du cycle de vie



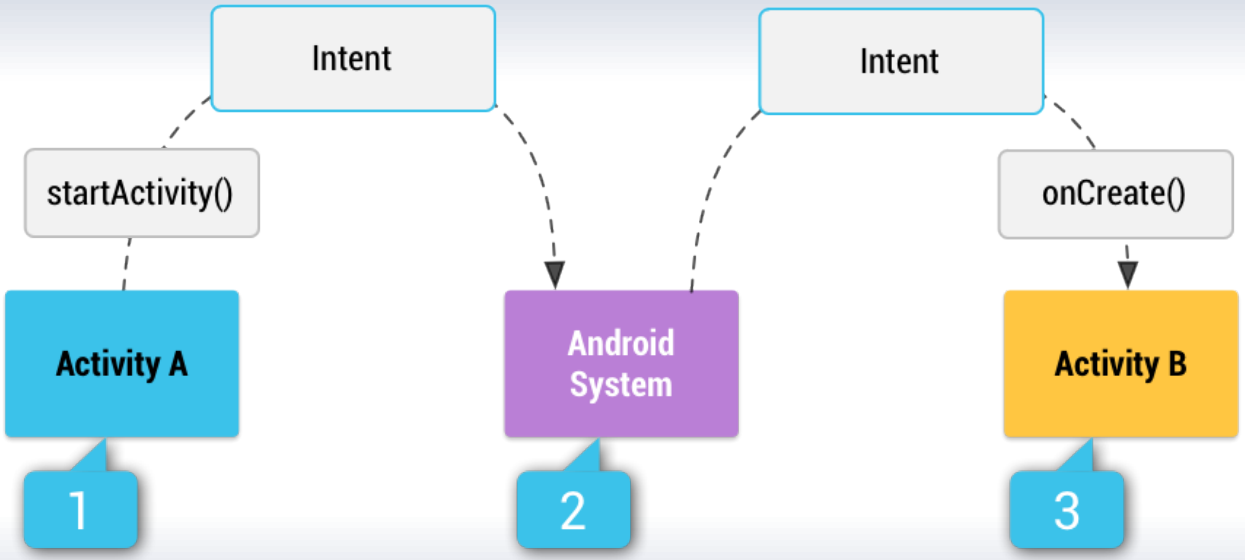
Cycle de vie : interactions utilisateurs



L'objet Intent

- **Intent**
  - lancement : **android.app.Activity.startActivity(Intent)**
  - **explicite** : lancement d'une activité spécifique
  - **implicite** : demande de lancement d'un service
  - communication entre activités : **android.content.Intent.putExtra(K,V)**

L'objet Intent



Quelques ressources web

LE site pour les développeurs Android

▸ Android developer

- Accès direct à la documentation ▸ Docs
- → ▸ Guides
- → ▸ API
- → ▸ Samples

The Busy Coder's Guide to Android Development

▸ Site

- Et sa base d'exemples open source très fournie ▸ GitHub