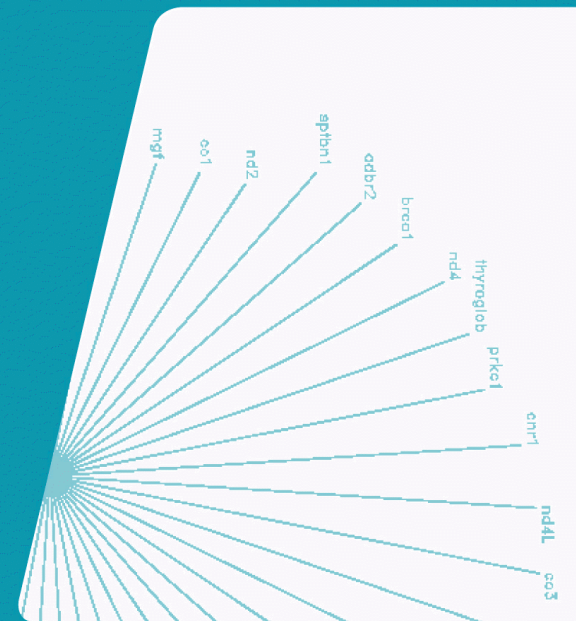




INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON



Outils, langage et approche Android

Une introduction

FORMATION



membre de
UNIVERSITÉ DE LYON

Nicolas Stouls

nicolas.stouls@insa-lyon.fr



Webographie

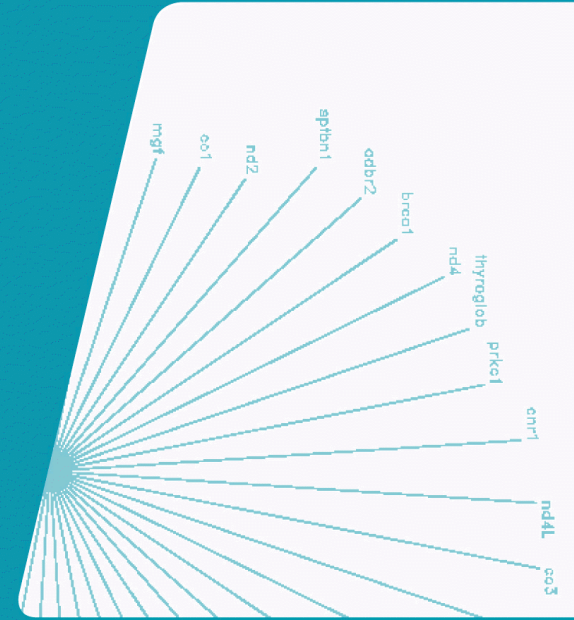
- La bible contenant « tout » :
<http://developer.android.com/index.html>
- Les supports cette intervention (slides et exemple) :
<http://perso.citi-lab.fr/nstouls/?ZoomSur=Enseignements>

Plan des points abordés

- Aperçu des spécificités d'une application Android
- Outils
- Ma première application
- Les points notables



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON



Spécificités d'une application Android

Un aperçu

FORMATION



membre de



Plate-forme à puissance et écran restreints

■ Puissance restreinte

- De moins en moins vrai, mais pas adapté aux gros calcul scientifiques

■ Écran restreint :

- Paradigmes classique d'IHM totalement remis à plat
- Pas de notion de **fenêtre**
- **Tailles d'écran** très variables.
 - *IHM donc très adaptatives*
 - *Listes déroulantes et tabulations très bien adaptées*

API \approx Java standard – Swing + Android

- Portage d'une application Java sous Android \approx
 - Définition / création d'une nouvelle IHM
 - Copié / collé de tout le reste
- L'accès aux périphériques fait parti de l'API
 - Java : machine virtuelle générique
 - Android : machine virtuelle avec contraintes sur le matériel
 - Accès facile et transparent aux périphériques / services :
 - *Wifi*,
 - *bluetooth*,
 - *USB*,
 - *GPS*,
 - *Accéléromètre*,
 - *téléphone*,
 - *SMS*,
 - *etc.*

Programmation orientée service

■ Service = Activité

- 1 application = Ensemble d'activités + activité « principale »
- 1 « écran » \approx 1 activité
- 1 activité =
 - *Description de son rôle*
 - *Ensemble de classes java avec point d'entrée*

■ Principe de l'appel de service :

- Direct : nom de la classe voulue

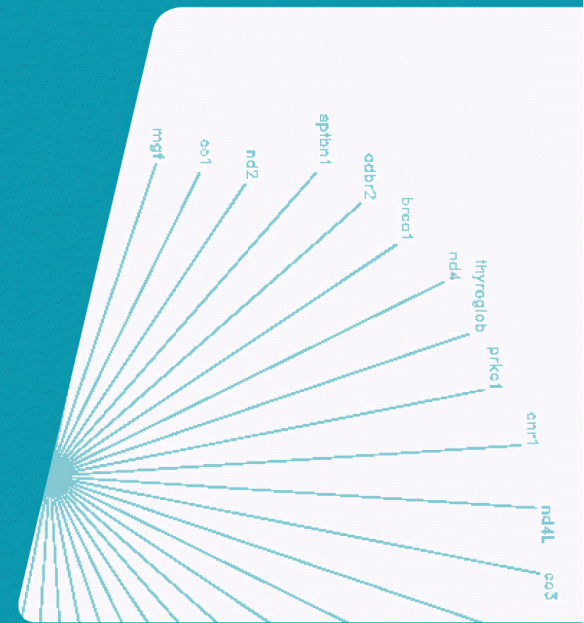
ou

- Indirect : description du service attendu
(*Ex : lecteur HTML ou édition texte*)

- Peuvent être décrites en Java ou XML
- Description XML plus condensée et facilite la réutilisation



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON



Outils

FORMATION



membre de



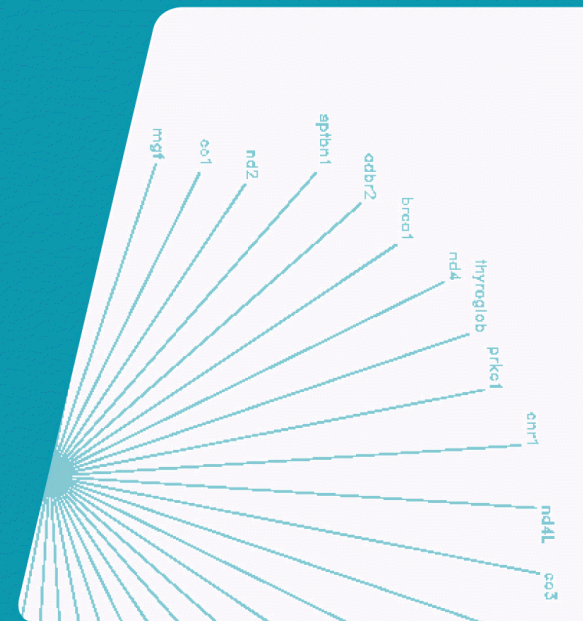
Outils

(32 ou 64bits : oui, mais rester homogène)

- Installer un kit de développement consiste à :
 - Installer Eclipse (Éditeur conseillé)
 - Installer SDK (outils)
 - *Driver pour communiquer avec un périphérique Android*
 - *Machine virtuelle permettant de tester des applications*
 - *Si utilisation machine virtuelle : créer une target*
 - *Choix taille écran, version API et taille mémoire*
 - Installer API (code)
 - *Présent sur les plate-formes, mais nécessaire à la compilation et au test*
 - *Standard VS Google : Google = Standard + API GoogleMaps (+?)*
 - Faire le lien entre éditeur et SDK
 - *Installation ADT*
- <http://developer.android.com/sdk/index.html>



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON



Ma première application

FORMATION



membre de



Démo

■ Grandes lignes :

– New -> **Android project**

- *Project name* -> *HelloWorld (Nom dans eclipse)*
- *API (Pour les tests)* -> *7*
- *Application Name (Nom affiché dans Android)* -> *Mon Helloworld*
- *Package Name* -> *(Nom du package) -> fr.insa*
- *Create Activity* -> *principale*
- *Min SDK Version (API minimum supportée)*-> *7*

– Run

- *Génération d'un fichier .apk*
- *Chargement du fichier dans le simulateur*

– Oh que c'est bô ! ;)

Démo

■ Qu'est-ce que cela a généré ?

– ./src/fr/insa/Principale.java

- *Activité initiale avec son « constructeur »*

public void onCreate(Bundle savedInstanceState) { ... }

– ./res/layout/main.xml

- *Descripteur de l'interface graphique*

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout .... ><TextView ... android:text="@string/hello" /></LinearLayout>
```

– ./res/drawable/{h/m/l}dpi/icon.png

- *Icône de l'application en haute, moyenne et basse résolution*

– ./AndroidManifest.xml

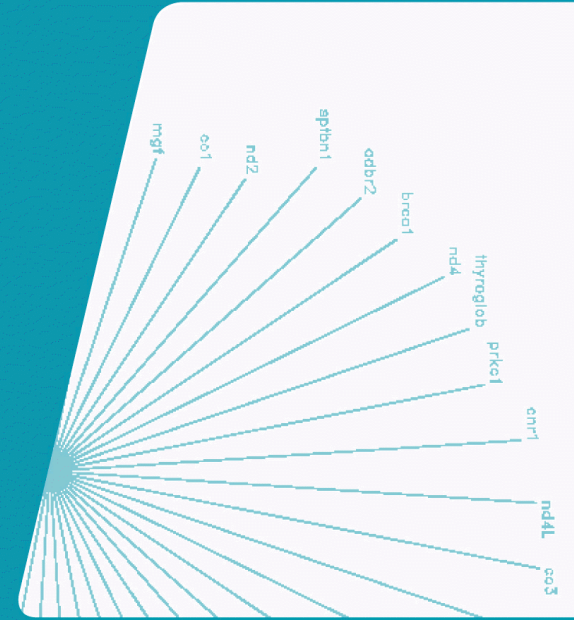
- *Description de l'application et des différentes activités qui la composent*

– ./default.properties

- *Mémorise le SDK minimum requis (entre autres choses)*



INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON



Les points notables

FORMATION



membre de



Primitives spécifiques à Android

■ Divers

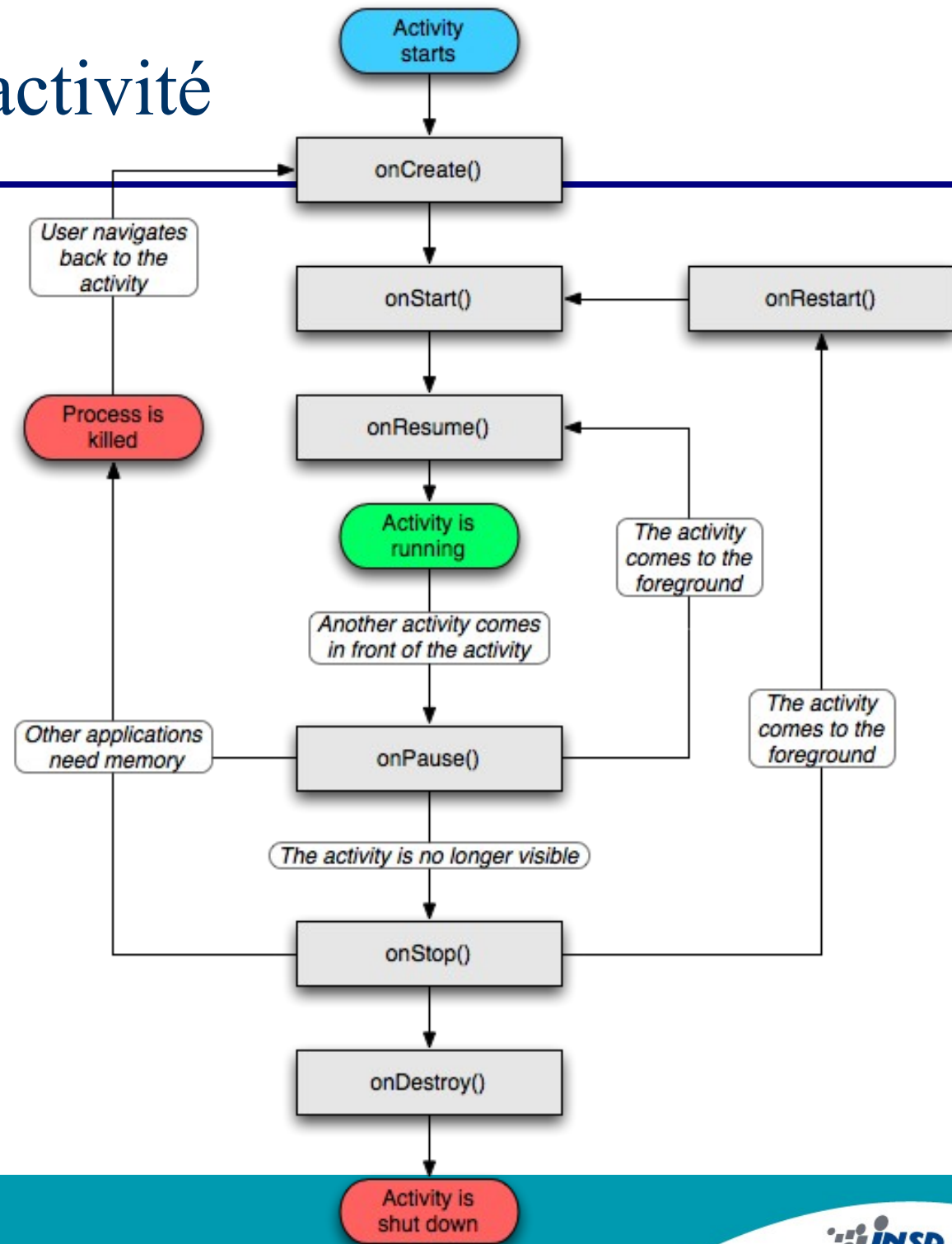
- **Activity** : Conteneur de tout service
 - *ListActivity*
 - *TabActivity*
- **Intent** : Descripteur de service

■ Graphique

- **View** : Structure graphique élémentaire (*Jpanel*)
 - *TextView* : champs texte en lecture seule (*JLabel*)
 - *ListView* : liste d'éléments
- **LinearLayout** : habilleur élémentaire (*FlowLayout*)
- **Button** : bouton classique (*JButton*)
- **EditText** : champs texte (*JtextField*)

Cycle de vie d'une activité

- Activité = service
- 1 Activité \approx 1 écran
- onCreate : action à redéfinir pour le lancement
- 1 activité à le focus et les activités s'empilent



onCreate

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
}
```

- **SavedInstanceState** : mémorise certaines informations lors d'un onPause() ou onStop().
- **SetContentView** : choisit le layout d'habillage de l'activité.

IHM : description en XML

■ Fichier XML contient :

– Déclaration encodage :

- `<?xml version="1.0" encoding="utf-8"?>`

– Conteneur de l'IHM compatible avec le type d'activité :

- *Activity* : `<LinearLayout ...></LinearLayout>`

- *ListActivity* : `<ListView ...> </ListView>`

- *TabActivity* : `<TabHost ...> </TabHost>`

- ...

■ Lien entre Java et XML :

– Nommage des entités XML :

`<Button android:id="@+id/myButton" />`

– Appel des entités nommées :

`setContentView(R.layout.main);`

`myButton = (Button)findViewById(R.id.myButton);`

Sécurité et manifest

- AndroidManifest.xml doit déclarer certaines actions pour que l'application puisse les faire :
 - android.permission.ACCESS_WIFI_STATE
 - android.permission.CHANGE_NETWORK_STATE
 - android.permission.CHANGE_WIFI_STATE
 - android.permission.INTERNET
 - android.permission.BLUETOOTH
 - android.permission.SEND_SMS
 - ...

Création d'un affichage personnalisé

Exemple : GraphView

■ Définition :

```
public class GraphView extends View
```

■ Modification de la méthode d'affichage :

```
protected void onDraw(Canvas canvas)
```

■ Appel depuis le XML (*Il faut totalement le qualifier*) :

```
<insalyon.tracecourbe.GraphView  
    android:id="@+id/graph"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent" />
```

Threads et mise à jour de l'affichage

- Principe :

Tout processus qui modifie l'affichage risque de générer un conflit avec la lecture de l'affichage par la plate-forme.

- Mise à jour de l'affichage en Java : `repaint()`
- Mise à jour de l'affichage en Android : `invalidate()`
- Mise à jour lorsque ressource dispo : `postInvalidate()`

Appel d'une activité

Appel explicite

- Exemple d'appel d'une activité nécessitant un paramètre (Para) :

```
Intent intent = new Intent(this, ConnectedActivity.class);
intent.setAction("fr.insa.START");
intent.putExtra("fr.insa.Para", ValeurATransmettre);
startActivity(intent);
```

- Déclaration de l'activité :

```
<activity android:name="ConnectedActivity">
  <intent-filter>
    <action android:name="fr.insa.START" />
    <data android:name="fr.insa.Para" android:mimeType="text/plain" />
  </intent-filter>
</activity>
```

- Récupération du paramètre par l'activité appelée :

```
Bundle b = getIntent().getExtras();
String parametre = b.get("fr.insa.Para").toString();
```

Appel d'une activité

Appel implicite

■ Exemple d'appel d'une activité de recherche sur google :

```
String requete = "http://www.google.fr/search?q=" + maRequete;  
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(requete));  
startActivity(intent);
```

■ Déclaration pouvant répondre à cet appel :

```
<activity android:name="GoogleSearch">  
  <intent-filter>  
    <action android:name="android.intent.action.VIEW" />  
    <data android:name="uri" android:mimeType="text/plain" />  
  </intent-filter>  
</activity>
```

Exemple pris sur : <http://www.pointgphone.com/tutoriel-android-introduction-intents-7779>

Communications entre services

- Paramètre lors de l'appel d'une activité
- Communication réseau sur le 127.0.0.1
- Données statiques publiques (attention aux accès concurrents)
- Gestion des préférences d'application
- Fichier
- SQLite DB