



Android



Jean-Claude Tarby

Laboratoire LIFL

Université Lille 1



Android, Google, San Francisco...





Avant tout...

- On charge les téléphones pour pouvoir les utiliser tout à l'heure...
 - Fiches de prêts impérativement à chaque séance
- Laisser les téléphones éteints pour le moment
 - Charge + rapide
 - On testera d'abord sur l'émulateur



Vous connaissez déjà Android ?

- Qu'en pensez-vous ?
 - Facile, simple, puissant... OU difficile, compliqué... ?
 - Pourquoi cet avis ?
 - Que pensez-vous par exemple :
 - des « fragments »,
 - de la gestion des BDD Sqlite
 - des layouts ?



Avis personnel

- Android est très puissant
 - On peut faire tout ce qu'on veut (y compris attaquer le noyau)
 - Fonctionne sur un vaste ensemble de périphériques
 - de la montre connectée à la TV connectée, en passant par la voiture connectée, etc.
- Mais...
 - Très long à apprendre
 - Parfois très complexe (tordu ?)
 - En perpétuel changement
 - Chaque année une nouvelle version au moins...
 - Outils encore trop basiques.



Nouveautés depuis 4.4/L

- New Android Runtime (ART)
 - Avant Dalvik, compilation à la volée
 - Maintenant ART avec optimisation du garbage collector, compilation en code machine natif lors de l'installation, support de débogage
- Nouvelles façons de notifier les utilisateurs
- Nouveaux composants graphiques
 - RecyclerView (mieux que ListView), CardView (vs. Fragment ?), touch feedback animations,...
- Modifications autour du multimédia
- Modifications autour du réseau
- ...
- Détails à <http://developer.android.com/preview/api-overview.html>



Ergonomie, UI Patterns, ...



Attention aux tailles des écrans !

The screenshot shows an Android application interface. At the top, there's a title bar with an Android logo and the text "Ma première application". Below it, a greeting "Bonjour tout le monde !" is displayed. A button labeled "Button" is visible. A progress bar is shown with a blue dot. Below the progress bar, a text field contains the text "ceci est un champ texte". At the bottom, there's a list of items: "Item 1", "Sub Item 1", "Item 2", "Sub Item 2", "Item 3", and "Sub Item 3".

3.2in HVGA slider (ADP1) 5.1in WVGA 10.1in WXGA (Tablet) Nexus One

Nexus S Galaxy Nexus Nexus 7 Nexus 4

Nexus 10

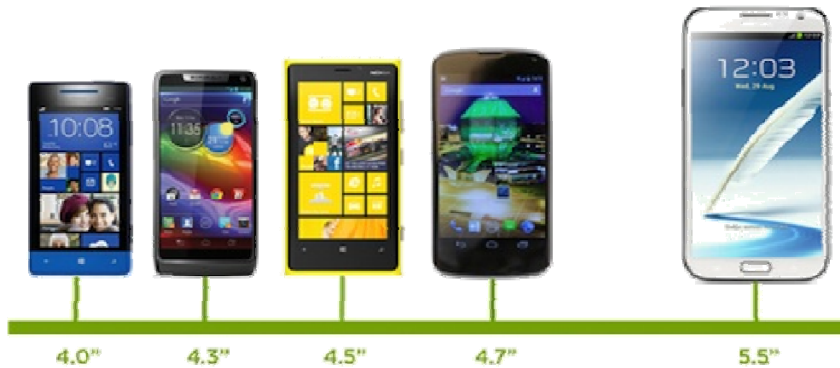
3 items

10 items

Beaucoup de téléphones différents



Attention aux tailles des écrans !

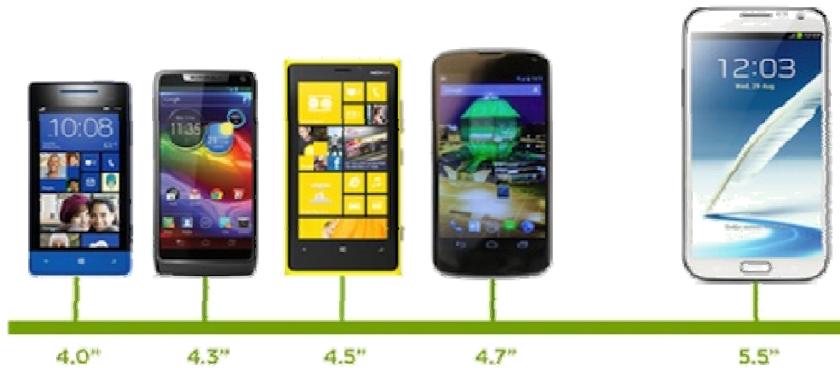


Beaucoup de téléphones, tablettes, ordinateurs...





Attention aux tailles des écrans !



Mais aussi...





Des montres et des bagues connectées





Des télévisions connectées





Des voitures connectées



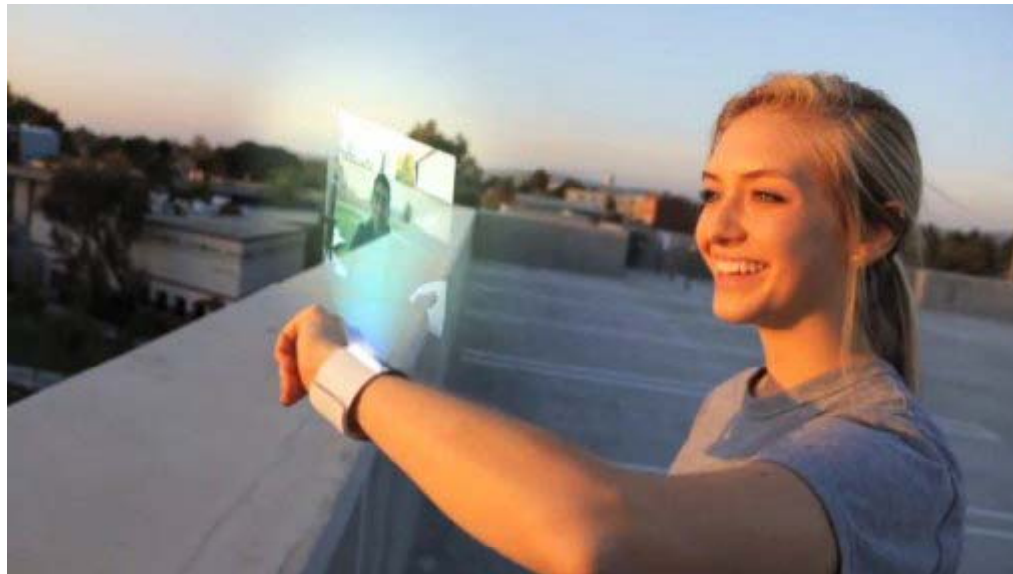


Des très grands écrans connectés





Des hologrammes connectés demain ?



Exemple à

<http://www.infohightech.com/un-casque-de-chantier-a-realite-augmentee/>

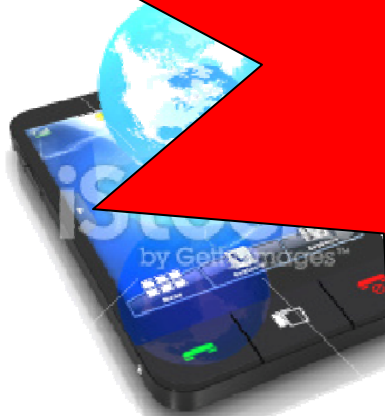




Des hologrammes connectés demain ?

Ca devient très compliqué !

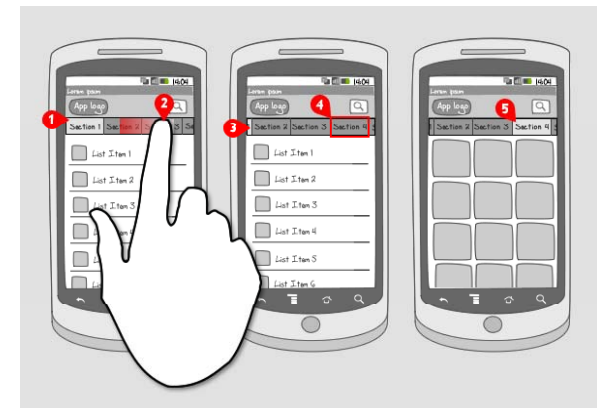
« card » Google Glass → 900”





Qualité de vos applications

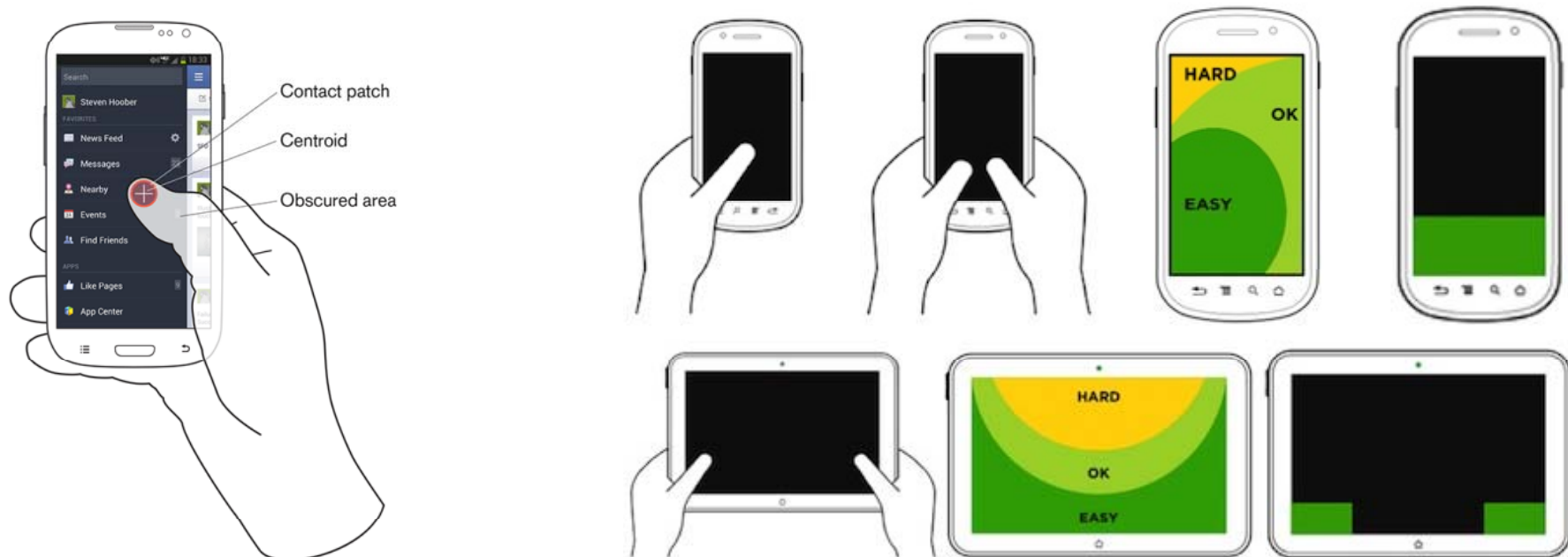
- De plus en plus de guidelines à respecter !
 - <http://developer.android.com/distribute/googleplay/quality/core.html>
 - À quand un outil pour le faire automatiquement ?
- Deux sources d'inspiration
 - <http://www.mobile-patterns.com/>
 - <http://www.androidpatterns.com/>





Quelques articles sur l'IHM mobile

- <http://www.simpleweb.fr/tag/tablette/> et <http://www.simpleweb.fr/> (mine d'infos)
 - <http://www.simpleweb.fr/2012/04/24/pour-bien-demarrer-dans-la-conception-dinterfaces-mobiles/>
 - <http://www.simpleweb.fr/2011/06/27/des-bibliotheques-de-composants-dinterfaces-mobiles/>
- <http://uxmag.com/articles/framework-for-designing-for-multiple-devices>





Aperçu de ce qu'on pourrait voir...

...si on avait 200h !

- Installation de l'environnement + 1ère application automatique
- Activités (StartActivity, Intents)
- Permissions
- IHM
 - layouts
 - widgets de base
 - ListView et Adapters
 - Spinner
 - Boite de dialogue...
 - Styles
 - Menus et ActionBar
- Data
 - BDD SQLite
 - Persistance des données (sauf SQLite)
 - Passage de données entre applications (hors fournisseurs de contenu/content provider)
 - préférences d'une application (sharedPreferences)
 - Stockage de texte simple dans un fichier
 - Fournisseur de contenus
- QRCode (lire et générer)
- Préférences (utilisateur et téléphone)
- C2DM (cloud to device messaging);
- Capteurs
 - GPS
 - Géoloc
 - Calcul itinéraire
 - Accéléromètre
 - Réseau
 - 3G, Wifi, Bluetooth...
 - Autres capteurs...
- AsyncTask
- Service
- Notification
- Threads
- Filtres
- Notification
- Appel/réception d'appel et de SMS
- Multimédia
 - son
 - vidéo
 - synthèse vocale
 - text-to-speech
- JSON et XML (parsing)
- Web services, connexion PHP /MySQL
- Publication sur le market + stats
- ...et encore plein d'autres choses...



Sommaire général

- Ce qu'on verra en cours/TD/TP
 - Architecture générale et fonctionnement d'une application
 - Layout
 - Activité
 - Ressources...
 - Appel d'autres applications (carte Google...)
 - Base de données SQLite
 - Capteurs
 - Fragments
 - ...?



Pour ceux qui connaissent déjà Android...

- Suivre le cours comme les autres ?
- Être « personne ressource » ?
- Découvrir par vous-mêmes des points particuliers d'Android ?
 - Les services, JSON, Cryptage,...
 - En faire un retour à toute la promotion pour que ça profite à chacun
- (Etre dispensé ? (après évaluation))



Quelques outils pour maquetter...

- Indigo Studio :
 - description succincte à <http://www.developpez.com/actu/51009/Infragistics-lance-Indigo-Studio-l-outil-gratuit-pour-la-conception-de-prototypes-d-IU-interactifs-pour-applications-Web-Desktop-et-mobile/>
 - outil dispo à <http://www.infragistics.com/products/indigo-studio>.
 - V2 payante (\$400 !), mais 30 jours gratuits
 - V1 gratuite : me contacter si nécessaire.
- Balsamiq (30 jours gratuits) : <http://balsamiq.com>
- Pencil (gratuit) : <http://pencil.evolus.vn/>
- un super petit outil pour montrer vos maquettes en contexte : <http://placeit.breezi.com/>
 - devenu payant ! ☹ ... mais « print screen » possible ☺
 - Quelques images encore gratuites
- <https://code.google.com/p/android-ui-utils/> et <http://www.journaldugeek.com/2011/11/28/android-design-preview-consulter-vos-maquettes-mobile/> : pour connecter Photoshop à vos mobiles
- <http://wireframesketcher.com/>
- <http://www.justinmind.com/prototyper/free-edition>
- <http://fluidui.com/android/> (adopté en 2013-2014 par vos prédécesseurs en e-services)
- <http://www.axure.com/>



(Très) Bonnes références (tutoriels, vidéos, forums)

- <http://developer.android.com/index.html>: **LA référence**
- <http://developer.android.com/training/index.html> : **tutoriaux** (mais des erreurs !)
- <http://developer.android.com/guide/practices/index.html> : « **best practices** » **officielles**
- ~~<http://www.siteduzero.com>~~ ==> <http://fr.openclassrooms.com/>: TB tutoriaux pour Android et iPhone
- <http://www.vogella.com/articles/Android/article.html> et <http://www.vogella.com/android.html>
 - dont ActionBar et Navigation Drawer (menu glissant) : <http://www.vogella.com/articles/AndroidActionBar/article.html>
- <http://blog.ace-art.fr/categories/Android> : tutoriels
- <http://stackoverflow.com/>: forum pour Android et iPhone
- <http://android.developpez.com/cours/>
- <http://www.tutos-android.com/>
- <http://forum.frandroid.com>
 - <http://forum.frandroid.com/topic/16808-serie-de-tutoriauxdevelopper-sous-android/>
- <http://android-coding.blogspot.fr> : tutoriels moins courants
- <http://www.mysamplecode.com> : tutoriels Android et iOS
- <http://blog.restomaniak.com/restomaniak-sur-votre-mobile/> : tutoriels
- <http://www.tutomobile.fr/> : tutoriels pour Android et iPhone
- <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/presentation-android.html> : cours pour se faire une idée générale d'Android
- Vidéos
 - <http://thenewboston.org>
 - <http://www.mybringback.com/tutorials/>



Autres références

- PhoneGap : <http://phonegap.com> pour produire sur 7 plateformes mobiles avec un seul code + développement hybride (natif + webview...)
- Bibliothèque SQLite : <http://ormlite.com>
 - Il y en a d'autres :
 - <https://github.com/pardom/ActiveAndroid>
 - <https://code.google.com/p/droidpersistence/> (bon ?)
 - <http://andorm.the-pixelpla.net/> (bon ?)
- Bibliothèque pour les webservices + cache
 - <http://www.technotalkative.com/android-volley-library-example/>
- Autres bibliothèques:
 - <http://www.androidviews.net/category/libraries/> : plein de refs de bibliothèques dont :
 - <http://androidannotations.org/> pour écrire beaucoup moins de code grâce aux annotations ! (cf. document de vos prédécesseurs belges sur Android 4)
 - <https://github.com/amigold/FunDapter> pour simplifier les Adapter dans les ListViews
- Pensez aussi à lire le document de vos prédécesseurs belges sur Android 4 (plein d'infos intéressantes)

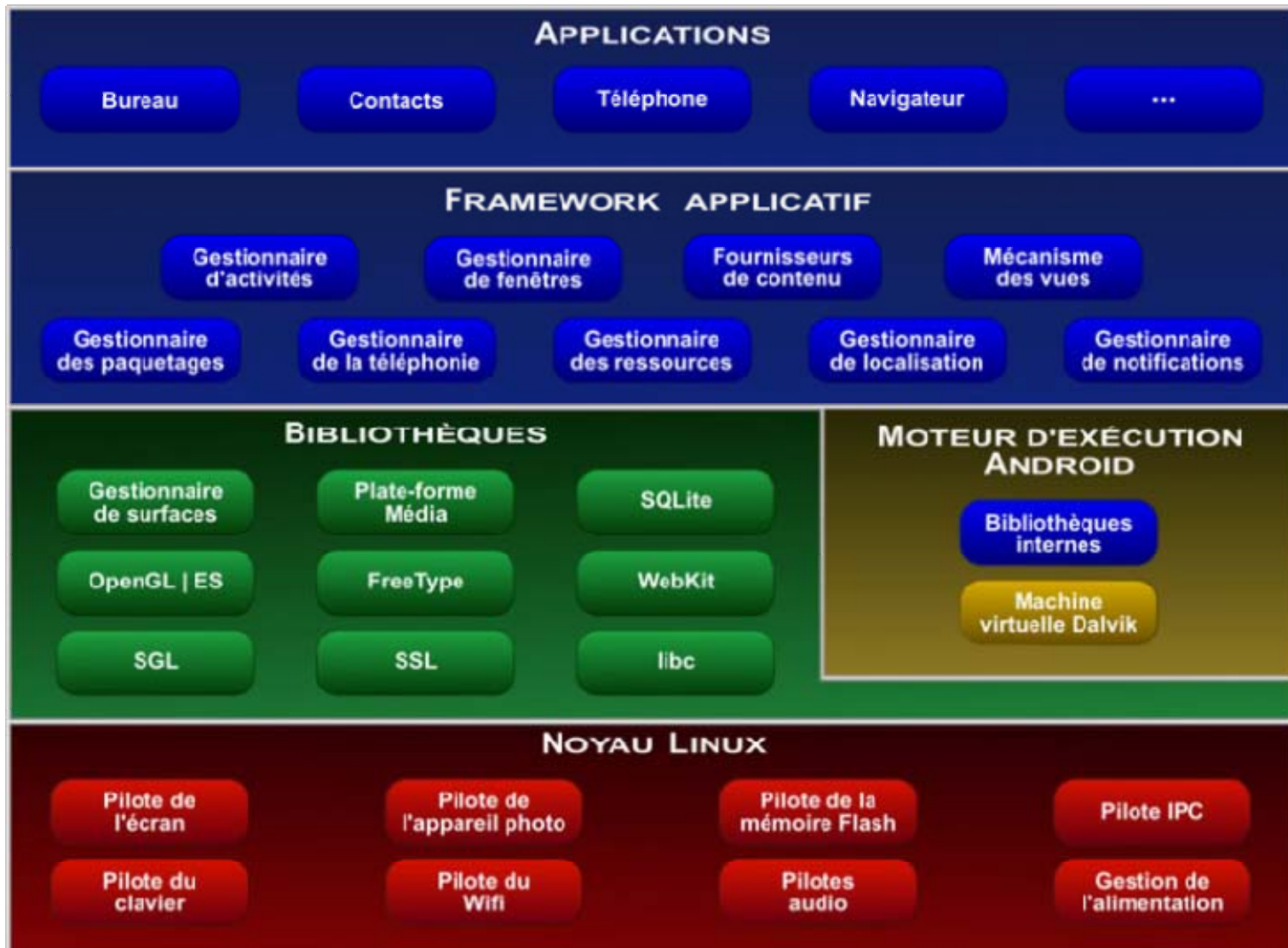


Préambule

- Android est né en septembre 2008 (V 1.0). Aujourd'hui 4.4.4 (L en preview).
 - Comparatif TB fait des versions :
<http://socialcompare.com/fr/comparison/android-versions-comparison>
- Sachez que les SDK Android contiennent beaucoup d'exemples de code (dans « samples »), pensez à les consulter).
 - Si vous souhaitez utiliser les exemples, les sources de ces applications ne comportent pas de fichier de projet → créer un nouveau projet et importer tous les fichiers de l'exemple dans la structure du projet
 - Nouvel assistant dans ADT pour voir des exemples tout faits



Préambule





Remarque pour Linux

- Vos postes sont configurés pour reconnaître nos téléphones ACER Liquid et Stream + quelques tablettes.

```
/etc/udev/rules.d/51-android.rules :  
SUBSYSTEM=="usb",ATTRS{idVendor}=="0502",ATTRS{idProduct}=="3202",MODE="0666"  
SUBSYSTEM=="usb",ATTRS{idVendor}=="0502",ATTRS{idProduct}=="3317",MODE="0666"  
SUBSYSTEM=="usb",ATTRS{idVendor}=="04e8",ATTRS{idProduct}=="681c",MODE="0666"
```

- Si vous voulez utiliser votre propre téléphone Android, ou un autre téléphone/tablette, il faudra faire ajouter de nouvelles règles par l'administrateur du M5

- lancer la commande « `lsusb` » qui vous enverra un message du style
Bus 001 Device 013: ID 22b8:708b Motorola PCS
- Cela signifie que `idVendor=22b8` et `idProduct=708b`. La ligne sera donc :
`SUBSYSTEM=="usb", ATTRS{idVendor}=="22b8", ATTRS{idProduct}=="708b", MODE="0666"`
- Envoyez **TOUS** les « Bus xxx Device yyy: ID zzzz:zzzz
XXXX » dans un mail groupé à mickael.carlier@univ-lille1.fr

+ d'infos à <http://doc.ubuntu-fr.org/android>



Introduction rapide à Android

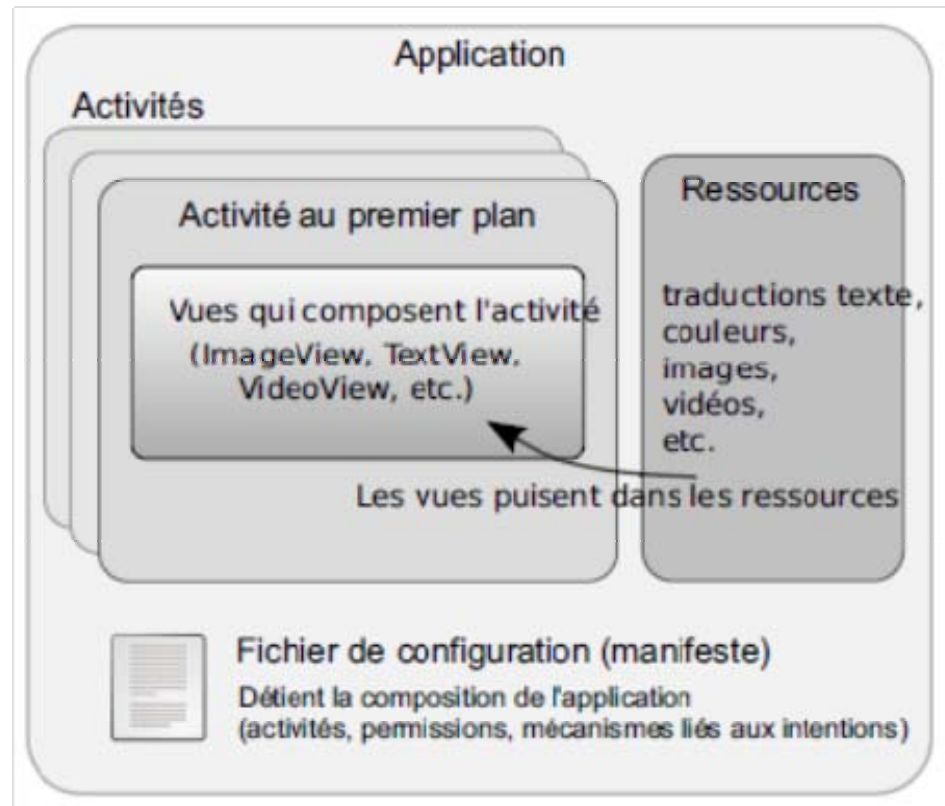
Quelques concepts de base

- Une application est décomposée en *activités* (et *fragments*).
- Une activité peut appeler d'autres activités (de l'application ou d'autres applications).
 - Si Activité 1 appelle Activité 2, on empile A2 sur A1
 - Android gère la mémoire et ces piles d'activités en conséquence
 - Bouton « back » = dépiler la dernière activité
- Les appels d'activités se font par des Intents
 - Intent, Activité, Vue
 - « qui sait faire X ? », « qui sait faire Y ? »
 - « métaphore de personnes avec habits »



Quelques concepts de base

- Activité
- Vues et contrôles (+ leur mise en page) ;
- Ressources
- le fichier Manifest





Activité

- Une **Activité**
 - Est un écran contenant des vues et des contrôles composant l'IHM de façon logique
 - Elle contient une hiérarchie de vues et de sous-vues.
 - Exemple : un formulaire d'ajout de contacts.
 - Est décomposée en :
 - logique + gestion du cycle de vie dans le **code java** de sa classe (héritant de Activity)
 - Il existe des classes d'activité qui facilitent (parfois) le travail, par exemple ListActivity
 - **IHM** : soit définie dans le code Java ci-dessus soit dans un fichier XML
- En général, si une application a « N » écrans, elle aura « N » d'activités.
 - Ce n'est plus vraiment le cas avec les Fragments...



Vues et Contrôles

- **Vues**
 - Ce sont les parties composant l'IHM
 - Elles contiennent des composants organisés suivant des mises en page (layout)
- **Contrôles** = widgets (boutons, checkbox...)
 - sous-ensemble des vues
 - accèdent aux textes et aux images qu'ils affichent en puisant dans les fichiers « ressources » de l'application



Ressources

- Stockées dans le dossier « res ».
- Contiennent les textes, images, sons, ... mais aussi les layouts, etc.

Type de ressource	Répertoire associé	Description
Valeurs simples	res/values	Fichiers XML convertis en différents types de ressources. Ce répertoire contient des fichiers dont le nom reflète le type de ressources contenues : 1. <code>arrays.xml</code> définit des tableaux ; 2. <code>string.xml</code> définit des chaînes de caractères ; 3. ...
Drawables	res/drawable	Fichiers <code>.png</code> , <code>.jpeg</code> qui sont convertis en bitmap ou <code>.9.png</code> qui sont convertis en "9-patches" c'est-à-dire en images ajustables. Note : à la construction, ces images peuvent être optimisées automatiquement. Si vous projetez de lire une image bit à bit pour réaliser des opérations dessus, placez-la plutôt dans les ressources brutes.
Layouts	res/layout	Fichiers XML convertis en mises en page d'écrans (ou de parties d'écrans), que l'on appelle aussi gabarits.
Animations	res/anim	Fichiers XML convertis en objets animation.
Ressources XML	res/xml	Fichiers XML qui peuvent être lus et convertis à l'exécution par la méthode <code>resources.getXML</code> .
Ressources brutes	res/raw	Fichiers à ajouter directement à l'application compressée créée. Ils ne seront pas convertis.



Multi-lingue...facile !

- res/values
 - strings.xml
- res/values-**fr**
 - strings.xml
- res/values-**es**
 - strings.xml



Mode portrait et paysage...facile !

- res/layout
- res/layout-land



Disposition sur l'écran...facile !(?)

- res/layout
- res/layout-**large**



Résolution et Style... facile ?

- res/values
 - dims.xml
 - styles.xml
- res/values-v11
- res/values-v14
- res/values-w820dp
- res/values-sw820dp
- ...
- Solution : les fragments (?)



Ressources

- Toutes ces ressources seront placées dans l'APK.
- Android crée une **classe statique R** qui sera utilisée pour accéder aux ressources depuis le code.
 - dans le « /gen »

```
package com.eyrolles.android.exemples;
public final class R {
    public static final class string { ①
        public static final int bienvenue-0x7f040000; ②
        public static final int texte_bouton_quitter-0x7f040001;
        public static final int texte_titre_ecran-0x7f040002;
    };
    public static final class layout {
        public static final int ecran_de_demarrage-0x7f030001;
        public static final int ecran_principal-0x7f030000;
    };
    public static final class drawable {
        public static final int image_android-0x7f020000;
    };
};
```



Fichier Manifest

- 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 *permissions* nécessaires à l'exécution du programme (accès à Internet, accès à l'appareil photo...).
 - dans les dernières versions, les « tools: »
 - <http://tools.android.com/tech-docs/tools-attributes>
 - Exemple : `tools:listitem="@android:layout/simple_list_item_2" />`



Fichier Manifest

- Une application « complexe » Android peut faire appel à :
 - des **activités** (composant applicatif)
 - des **services** (composant applicatif)
 - des **fournisseurs de contenus** (composant applicatif), par exemple pour accéder à des données en BDD
 - des **gadgets** (composant applicatif)
 - des **Intents**
 - des **Récepteurs d'Intents**
 - des **Notifications**.



Intents

- Les objets **Intents** permettent de diffuser des messages demandant la réalisation d'une action → permettent de fournir ou de demander des services. C'est Android qui « choisit » QUI fera l'action demandée.
- **Récepteurs d'Intents** : permettent à l'application d'être à l'écoute pour répondre aux objets Intents qui lui demanderaient quelque chose.
- **Notification** : signale une information à l'utilisateur sans interrompre ses actions en cours.
- **Filtres d'Intents** : un objet Intent peut mentionner explicitement un composant cible. Dans le cas contraire, Android choisit le meilleur composant grâce aux filtres (cf. <intent-filter> du fichier de config Manifest).



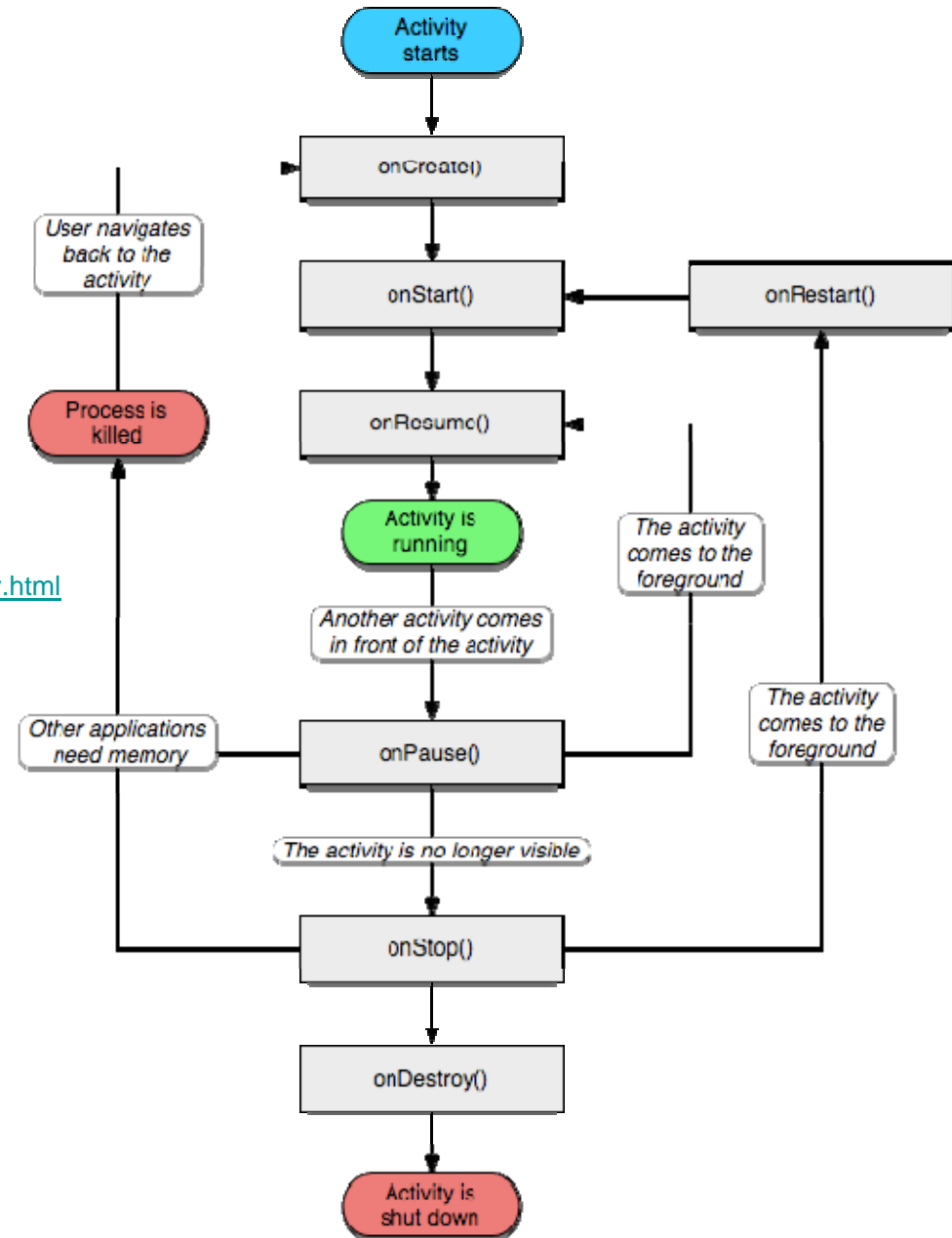
Permissions

- Pour accéder à certaines fonctionnalités, on doit déclarer des permissions dans le Manifest
 - par exemple pour envoyer des SMS, accéder au compte Google, etc.
- Permet à l'utilisateur d'en avoir conscience et de choisir.
- Permet aussi au Play Store de filtrer les applications pour ne proposer que celles qui peuvent fonctionner sur le périphérique connecté.
 - Peut être contourné en utilisant intelligemment `<uses-permission>` et `<uses-features android:required = "false">`
 - À vous de faire attention dans votre code !



Cycle de vie d'une application

source : <http://developer.android.com/reference/android/app/Activity.html>





Outils de développement (1/2)

- **ADT / Eclipse + plug-in Android**
 - <http://developer.android.com>
 - **Bug dans la version actuelle** (septembre 2014) !
 - Importer le « support v7 » pour que ça marche
 - Détails à <http://developer.android.com/tools/support-library/setup.html>
 - » Détails : <http://developer.android.com/tools/support-library/features.html>
- **Android Studio**
 - Basé sur IntelliJ IDEA
 - <http://developer.android.com/sdk/installing/studio.html>
 - Futur bon outil ? (utilisé partout à SF)
 - + d'options que dans ADT, mais encore plein de « soucis »

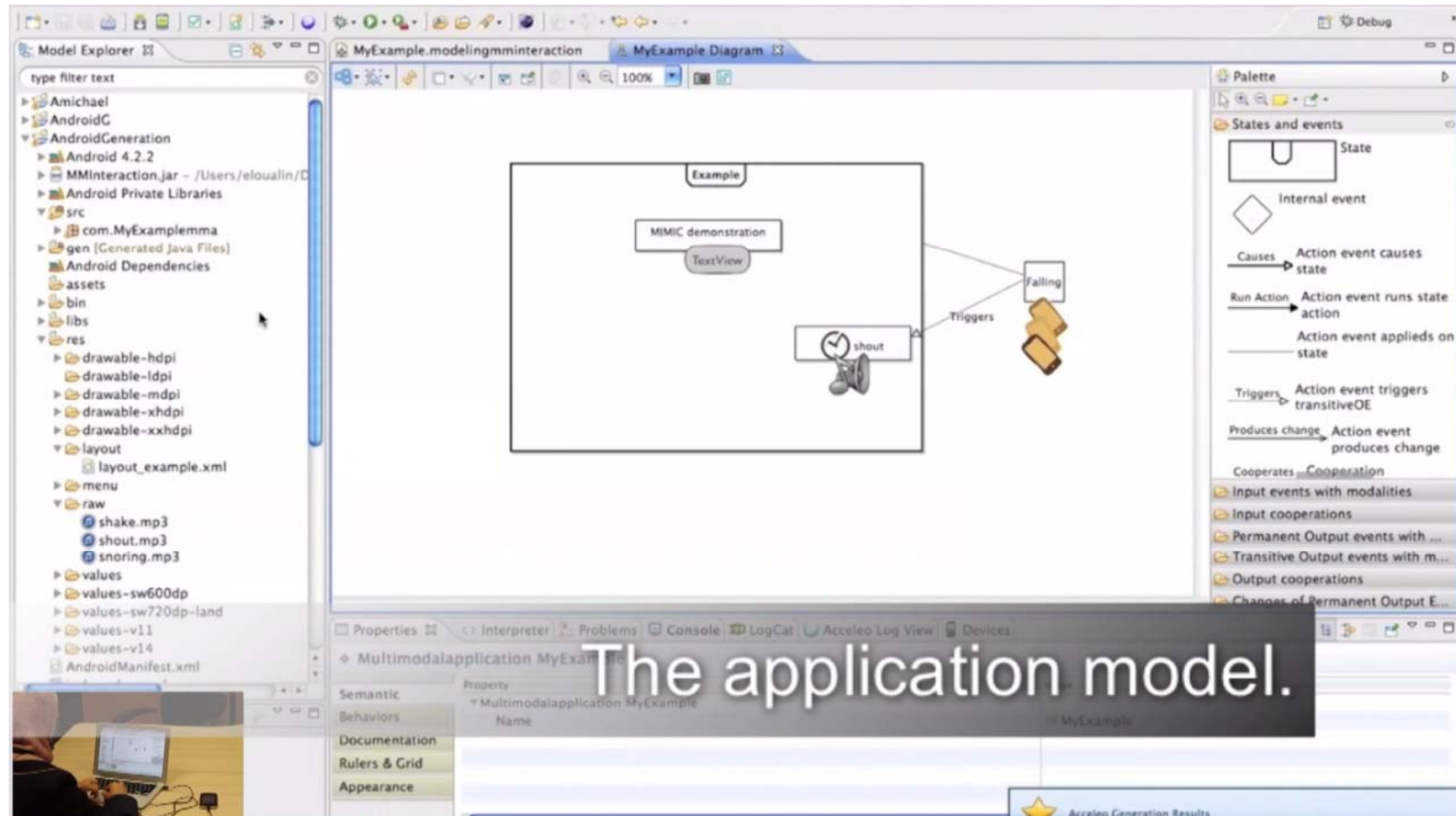


Outils de développement (2/2)

- **AppInventor (MIT)**
 - Basé sur Scratch du MIT : <http://scratch.mit.edu/>
 - <http://appinventor.mit.edu>
- **Intel XDK**, pour du cross-platform
 - <https://software.intel.com/en-us/intel-mobile-development-kit-for-android>
 - <http://xdk-software.intel.com/index.html>
 - <https://software.intel.com/en-us/html5/tools>
- Notepad++ et autres éditeurs + compilateur en ligne de commande...(si vous êtes courageux !)



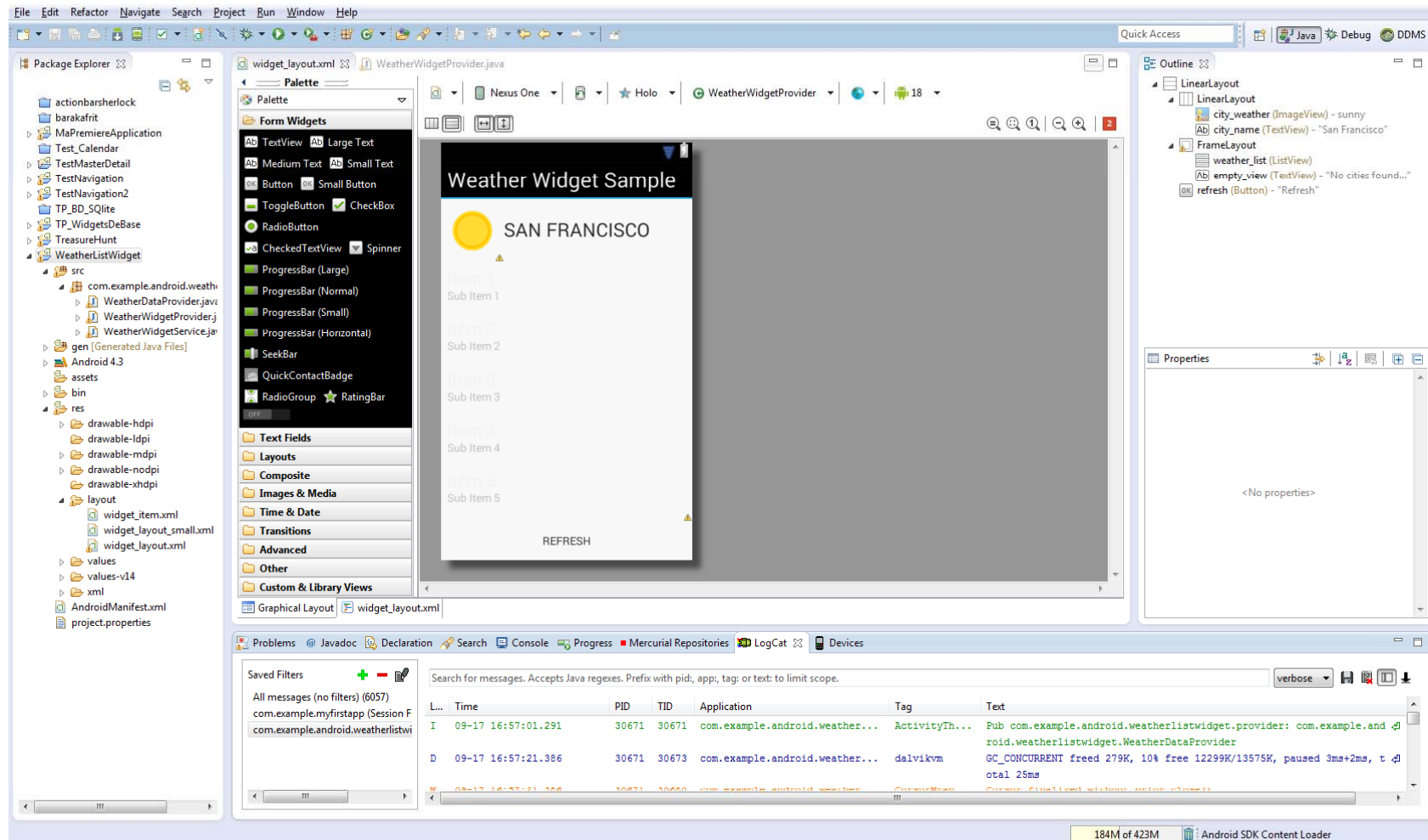
Futurs outils ?



<http://youtu.be/g0Ekioe1CQ0?list=UUofGGFqq9t1kfKOAzn5HJGQ>



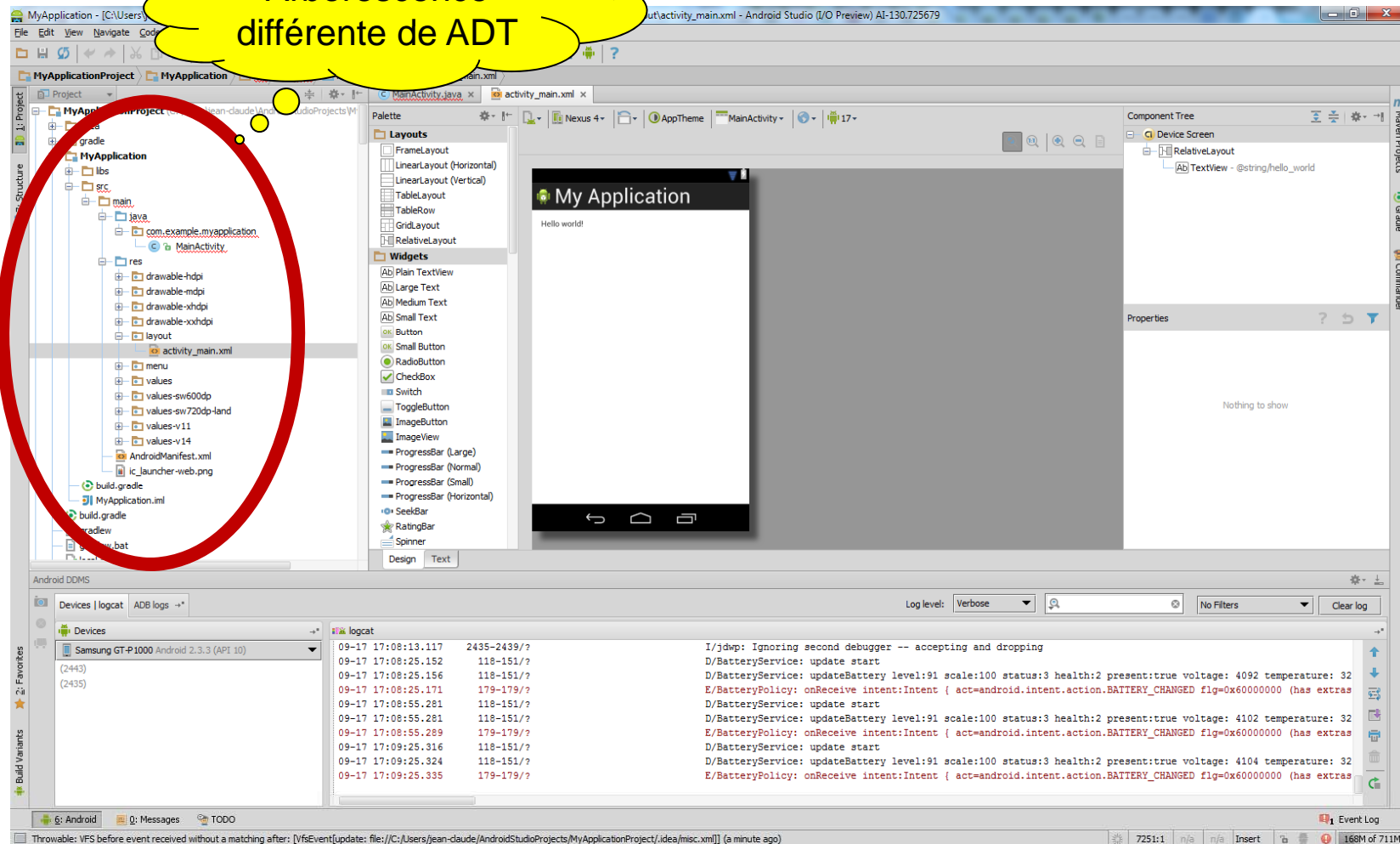
ADT (ou Eclipse + plug-in Android)





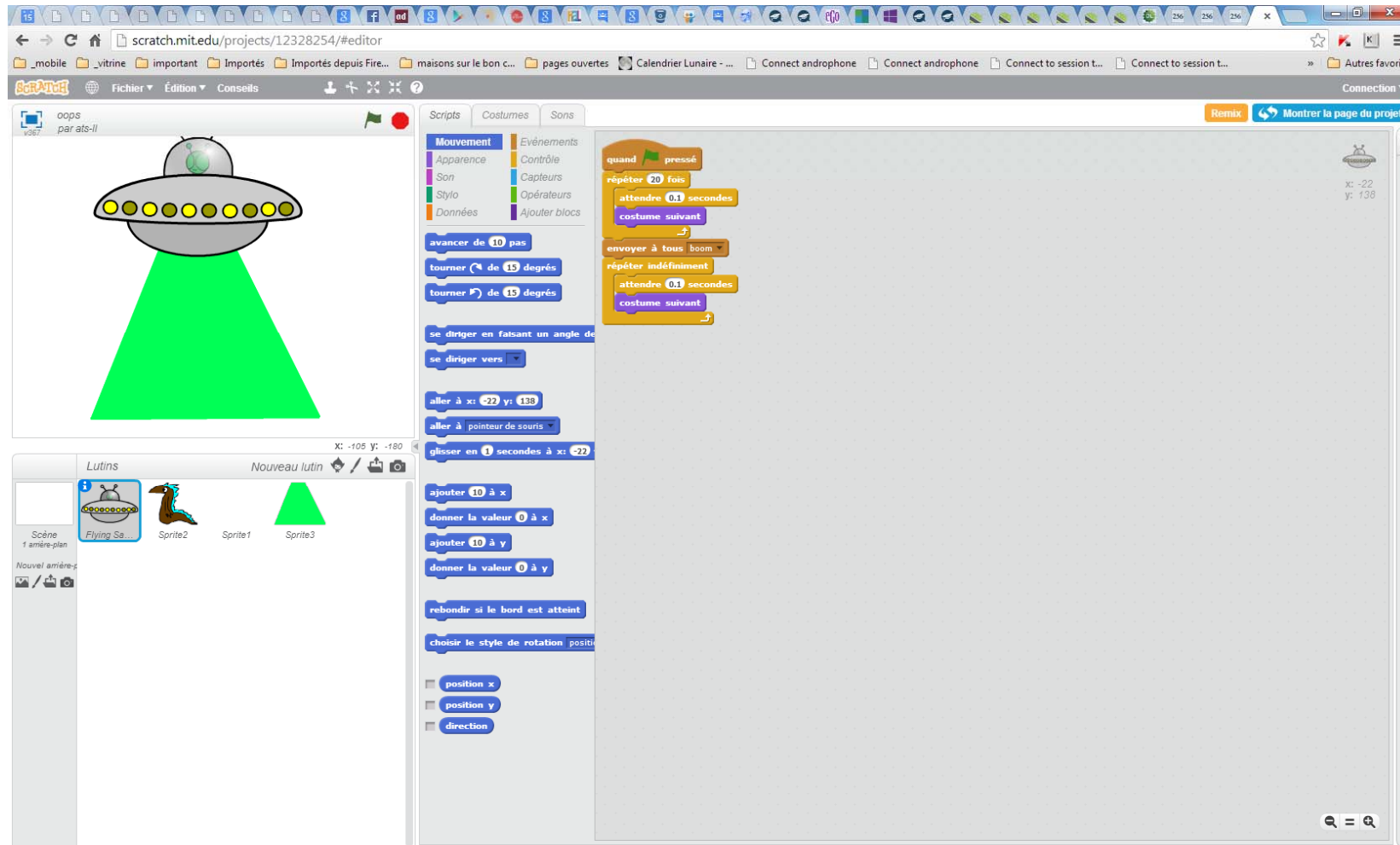
Android Studio

Arborescence
différente de ADT



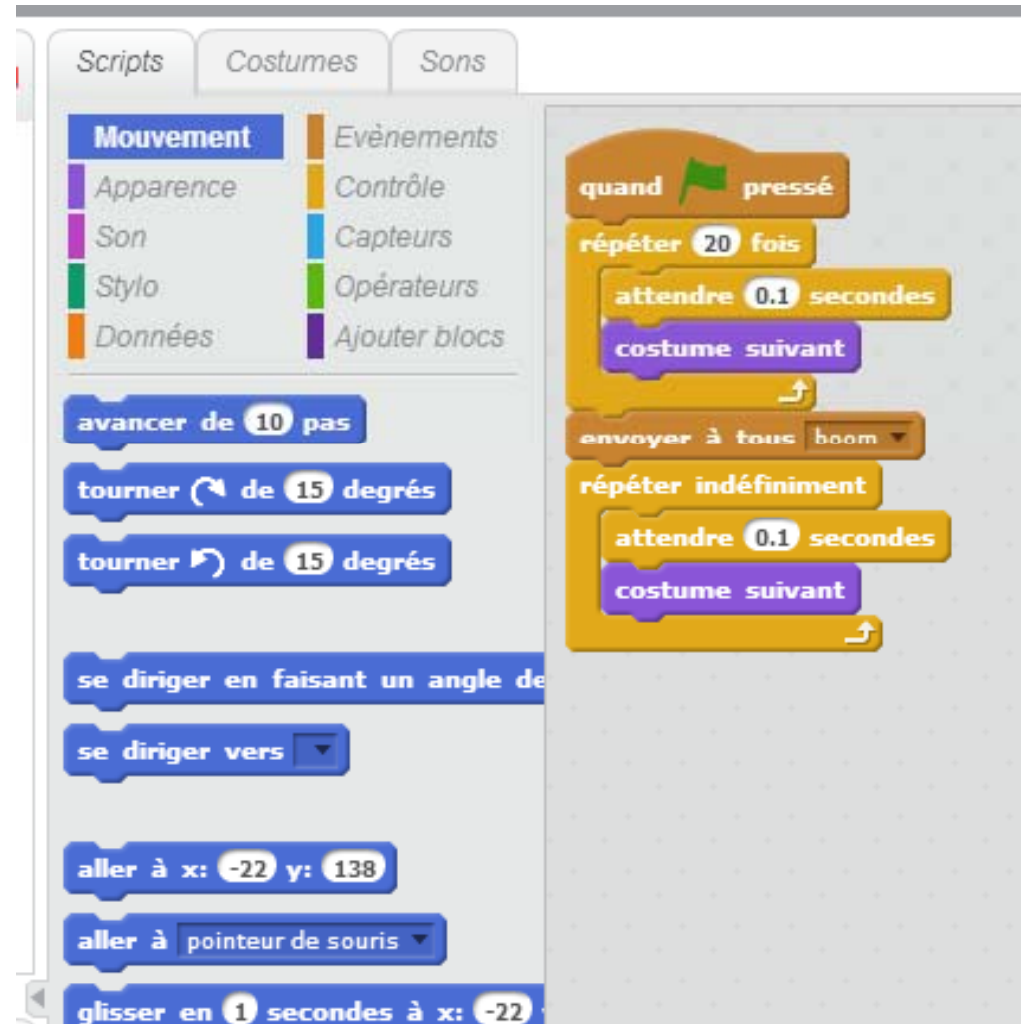


Scratch (à partir de 8 ans !)





Scratch (à partir de 8 ans !)





App Inventor (à partir de 8 ans ???)

