

# Technologies mobiles

Olivier Levitt

January 19, 2013



ios

Windows 8  
Phone



# Sommaire

## Présentation et objectifs du cours

- Organisation administrative

- Champ du cours

## Le développement mobile

- Comparaison des différents OS mobile

- Spécificités du développement mobile

## Le développement sur android

- Mise en place

- Architecture

# Planning

- ▶ 30 janvier : 3h de cours, 3h de TP
- ▶ 6 février : 3h de cours
- ▶ 13 février : 6h de TP
- ▶ Validation des sujets de projet avant le 20 février
- ▶ 20 février : 6h de TP dédiées au projet
- ▶ ? mars : Soutenance du projet

# Evaluation

- ▶ Projet + soutenance
- ▶ Groupe de 2
- ▶ Sujet “libre”
- ▶ 6h de TP dédiées au projet + travail personnel

# Contexte

- ▶ Smartphones, tablettes et assimilés (TV, montre, autoradio ...)
- ▶ Dev d'application, pas de dev de la plateforme
- ▶ 1ère partie : le dev mobile en général
- ▶ 2ème partie : application sous android

# Des appareils suréquipés

- ▶ Téléphonie (SMS, MMS, appels)
- ▶ Internet (GPRS, EDGE, 3G, 4G, WIFI)
- ▶ Réseaux locaux (Bluetooth, réseaux adhoc, NFC)
- ▶ Capteurs (Luminosité, proximité)
- ▶ Localisation (GPS, triangulation, SSID wifi)
- ▶ Notifications (Vibreur, haut-parleurs, LED)
- ▶ Stockage de données (Mémoire flash, SD externe, SQLite)
- ▶ Interactions (Ecran tactile, gestures, boutons physique)
- ▶ Et encore d'autres ...

Et des API pour utiliser tout ça !

# Des contraintes techniques importantes

- ▶ Processeur
- ▶ Mémoire RAM
- ▶ Stockage de données
- ▶ Gestion de la batterie
- ▶ Stabilité et débit de la connexion internet
- ▶ Cycle de vie de l'application
- ▶ Taille d'écran
- ▶ Inputs atypiques (clavier virtuel, gestes, peu de boutons ...)

Contraintes à garder en tête en permanence.

# La fragmentation

Une application publiée sur le google playstore cible plus de 2400 appareils différents !

- ▶ “Write once, run everywhere” ?
- ▶ Comment tester / débbugger pour tous ces appareils ?
- ▶ Eviter de gêner l'utilisateur (versions HD, appareils non compatibles)
- ▶ S'adapter quand une fonctionnalité n'est pas disponible



# La fragmentation, taille d'écran

Comment gérer toutes les tailles d'écran ?

- ▶ Montres connectées : de 1 à 2 pouces
- ▶ Smartphones lowcost : 3 pouces (Galaxy pocket, galaxy Y)
- ▶ Smartphones high-end : 4 à 5 pouces (iPhone 5, HTC 8X, nexus 4)
- ▶ Phablets : 5 à 6 pouces (Galaxy note, HTC butterfly)
- ▶ Tablettes : 7 pouces (Nexus 7, iPad mini), 8 pouces (Archos 80g9), 10 pouces (Nexus 10, iPad)

# De nombreuses autres sources de fragmentation

- ▶ Versions de l'OS
- ▶ Résolutions d'écran
- ▶ Elements hardware présents
- ▶ Puissance
- ▶ Modifications constructeur / “rom custom”
- ▶ ...

# Des Ecosystèmes forts

- ▶ Obligation d'utiliser le SDK fourni
- ▶ Suivre les guidelines
- ▶ Restrictions liées à la plateforme
- ▶ Utilisation des services de la plateforme
- ▶ Processus de déploiement des applications
- ▶ Règles des “store” (validation, monétisation ...)

# Sommaire

## Présentation et objectifs du cours

- Organisation administrative

- Champ du cours

## Le développement mobile

- Comparaison des différents OS mobile

- Spécificités du développement mobile

## Le développement sur android

- Mise en place

- Architecture

# Les marque-page

- ▶ [www.frandroid.com](http://www.frandroid.com) (actu FR)
- ▶ [www.androidpolice.com](http://www.androidpolice.com) (actu EN)
- ▶ [www.androidcentral.com](http://www.androidcentral.com) (actu EN)
- ▶ [www.d.android.com](http://www.d.android.com) (la bible EN)
- ▶ [www.stackoverflow](http://www.stackoverflow.com) (Q/A EN)
- ▶ #android et #android-dev sur freenode (chat irc EN)
- ▶ [www.breizhjug.org](http://www.breizhjug.org) et [www.paug.fr](http://www.paug.fr) (communautés FR)

# Bien commencer

La programmation android fait partie des plus accessibles :

- ▶ Des (bonnes) bases de programmation en JAVA
- ▶ Un ordinateur (Windows, Linux, Mac OS X)
- ▶ Un appareil android (conseillé, l'émulateur étant . . . moyen)

C'est tout !

# Présentation du SDK android

Téléchargement gratuit : [www.d.android.com/sdk](http://www.d.android.com/sdk)



add-ons



docs



extras



platforms



platform-tools



samples



sources



system-images



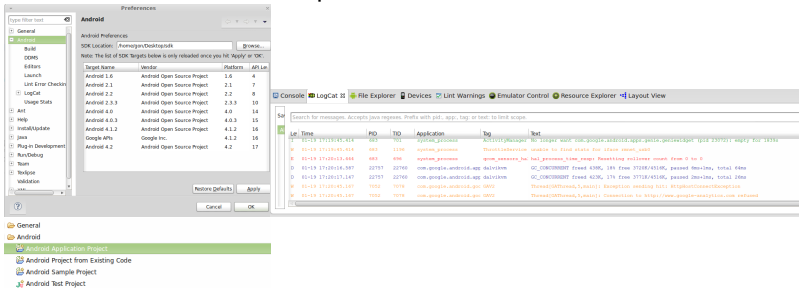
temp



tools

# Plugin android pour eclipse : ADT

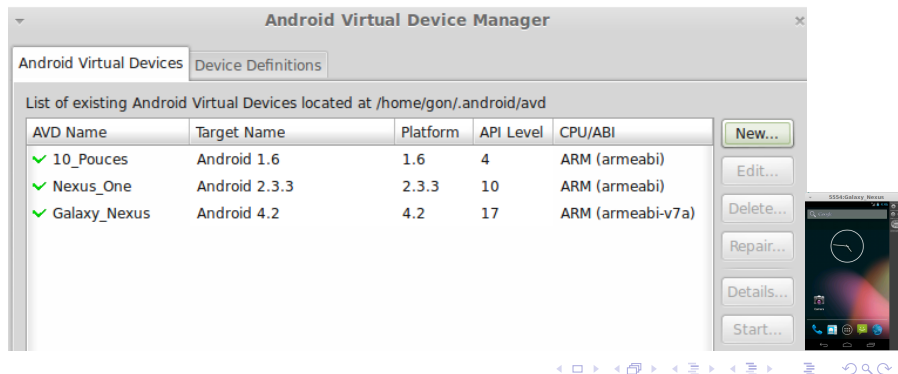
ADT fait le lien entre eclipse et le SDK android





# L'émulateur

- ▶ Utile pour tester certaines configurations
- ▶ ((très) (très)) lent
- ▶ Utiliser un appareil android à la place quand c'est possible



# Distribuer l'application

- ▶ Une application android = un APK (+/- équivalent d'un jar)
- ▶ Création et signature de l'APK simple sous eclipse
- ▶ Distribution directe de l'APK (ex : pour tester, bêta fermée)
- ▶ Publication sur le playstore, 25\$ à l'inscription
- ▶ Application gratuite ou payante (30% pour google)

# Activity et cycle de vie de l'application