



Cours 1

Historique et situation du développement en jeu vidéo sur appareils portatifs.

Un défi constant!

- Contraintes technologiques
- En changement continu
 - Plus d'une révision majeure Android par année
 - Révisions annuelles de iOS
- Fragmentation du marché
- Peu de sensibilisation des clients



Domaine en croissance

- Ventes des appareils intelligents augmentent à chaque année
- La technologie est constamment en évolution
- Les utilisateurs en dépendent pour des tâches de plus en plus variées
- Depuis 2017, les systèmes d'exploitation mobile sont maintenant les plus communs (dépassant PC, Mac, consoles, etc.)

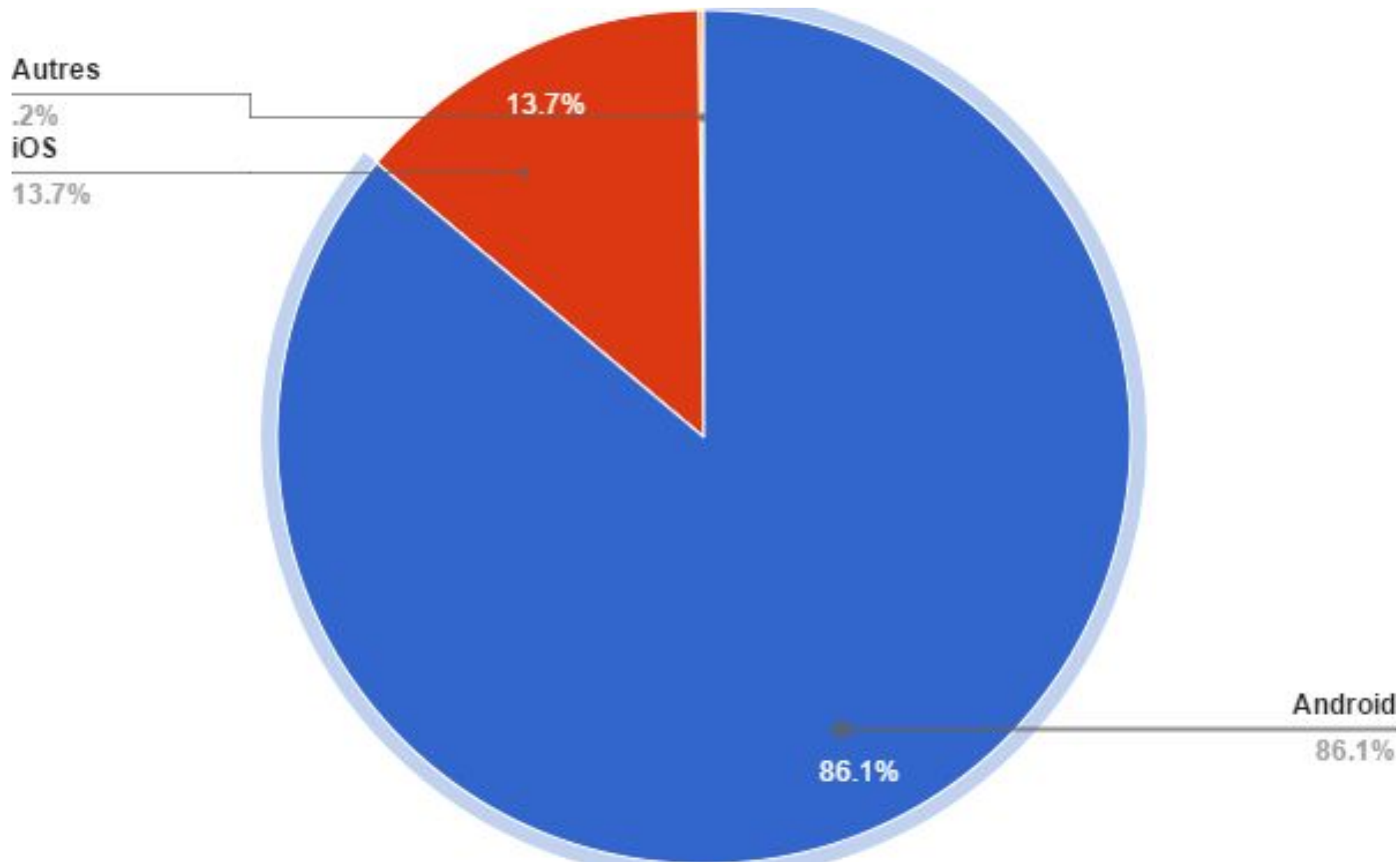
Historique des appareils intelligents

- Apparitions principales vers la fin des années 90 par Nokia, BlackBerry, et Palm
- Symbian (Nokia) a dominé le marché de 1996 jusqu'à 2010
- Apparition du iPhone en 2007, ainsi que la création du App Store en 2008
- Premier Android (HTC Dream) et Android Market en 2008

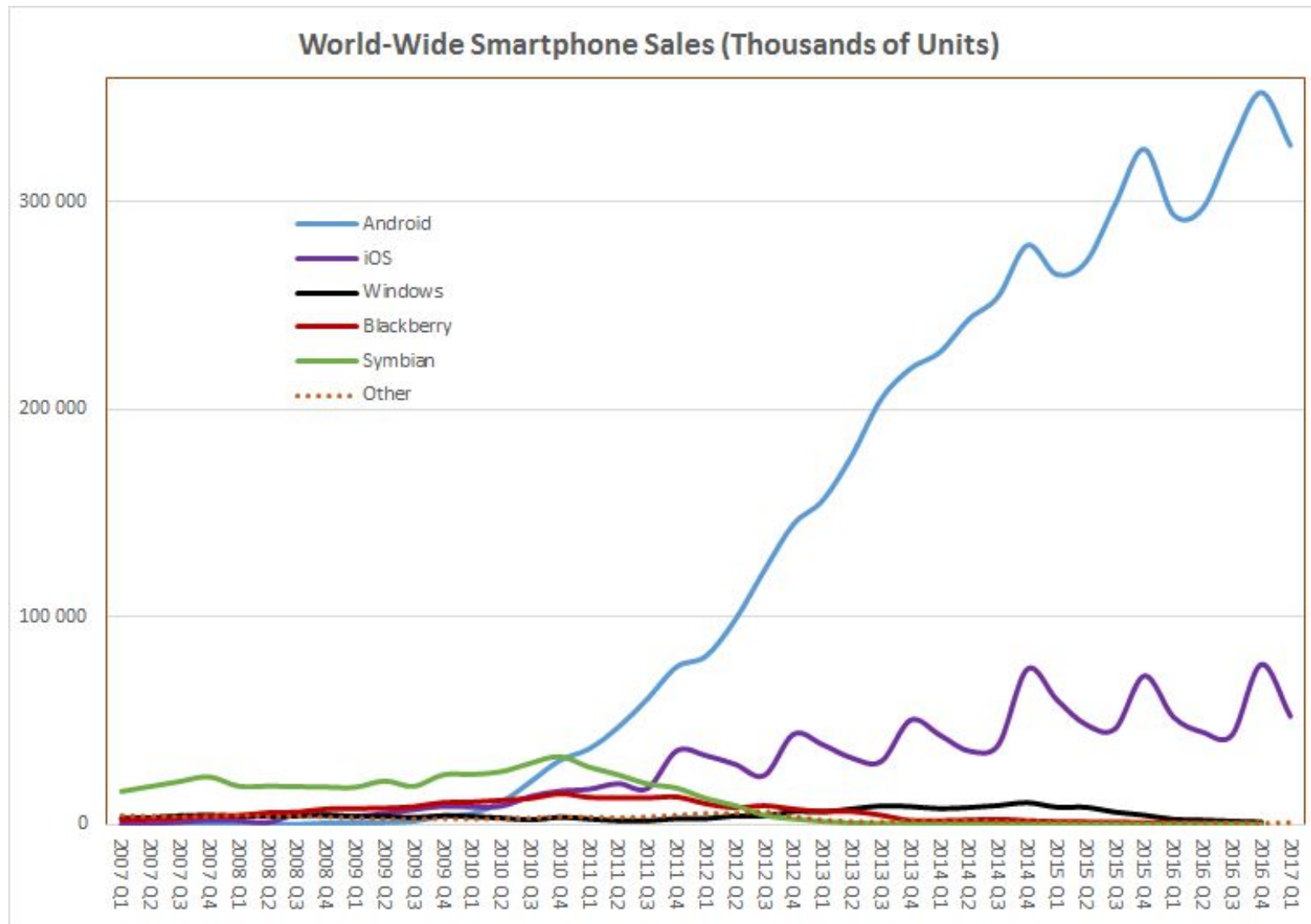
Historique des appareils intelligents

- Croissance rapide des ventes des appareils iOS et Android, avec Android qui remplace Symbian en première place depuis 2010
- App Store a été longtemps numéro 1 en revenus depuis sa création, mais Google Play domine maintenant en nombre de téléchargements.
- Google Play a une croissance rapide et, en 2016, était à 50% des revenus du App Store, mais il est prédit que le revenu de toutes les sources Android (Google Play, Amazon, etc.) combinées surpassera celui du App Store.

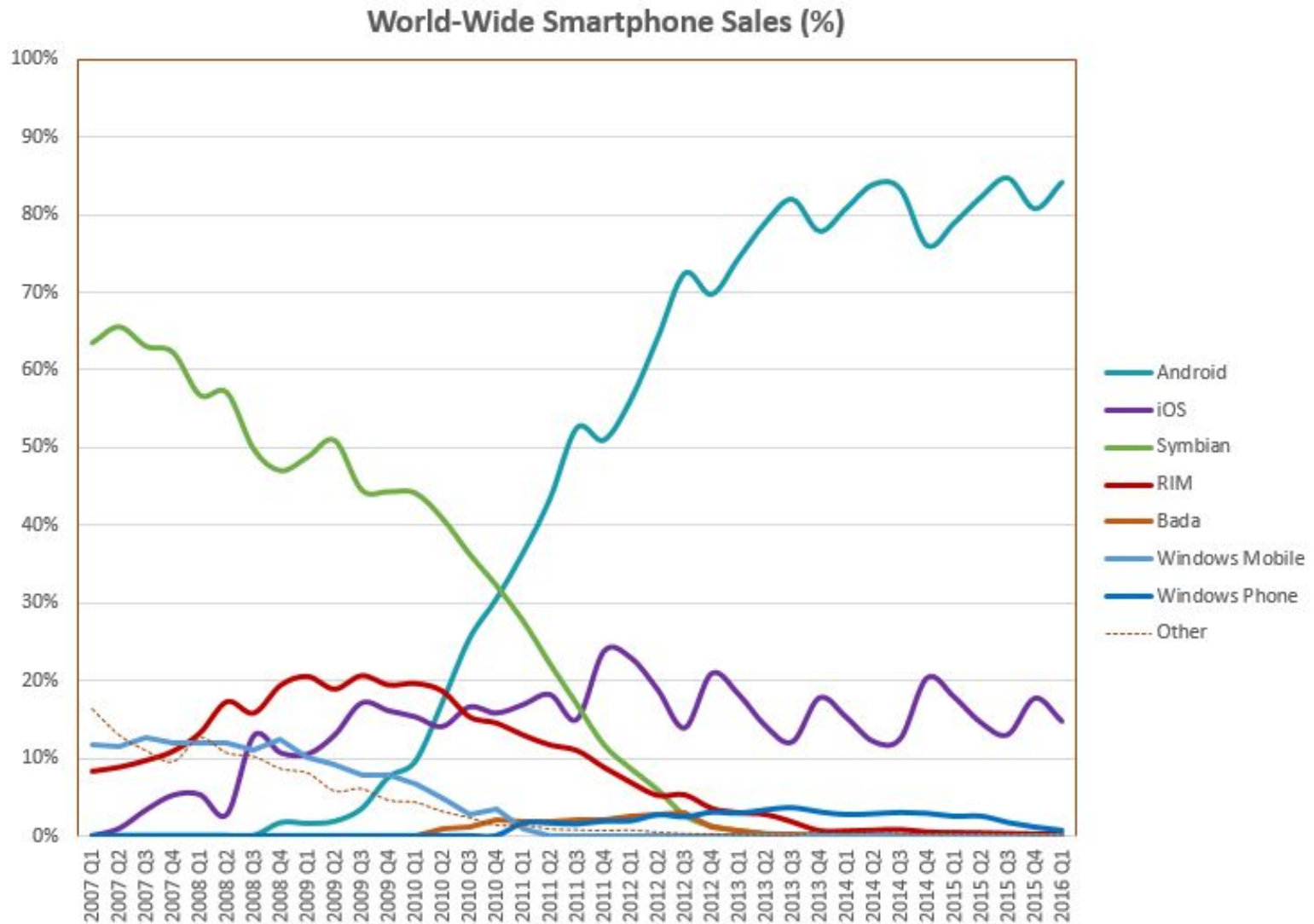
Ventes mondiales 2017Q1



Source : http://en.wikipedia.org/wiki/Mobile_operating_system



Source : http://en.wikipedia.org/wiki/Mobile_operating_system



Cours 1

Introduction à la plate-forme Android



Qu'est-ce que Android?

- Système d'exploitation de Google et du Open Handset Alliance (84 membres incluant Google, HTC, Sony, Dell, Intel, Motorola, Samsung, LG, etc.)
- En développement depuis 2003
- Premier téléphone en 2008 (HTC Dream)

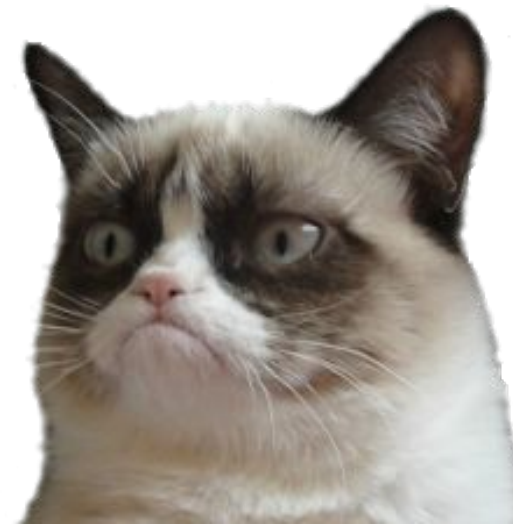


Caractéristiques

- Noyau Linux 2.6 (et 3.x depuis *Ice Cream Sandwich*)
- Architecture ARM principalement
- Inclut des bibliothèques systèmes en C et C++
- « *Runtime* » principalement en Java, se servant de JNI pour optimisation et accès au système

Dalvik

- Machine virtuelle servant à l'exécution du Java sur Android
- Utilise un format de fichier binaire (DEX) spécialisé pour les contraintes mobiles
- Architecture par registres plutôt que par pile



ART

- *Android Runtime* (ART) disponible expérimentalement depuis KitKat et officiel depuis Lollipop.
- Tout le code Java pour Dalvik est désormais recompilé en natif (comme une application C/C++) et s'exécute dans une émulation de l'environnement Java.
- Il n'est donc plus nécessaire de programmer en C/C++ pour la performance. Ceci est par contre encore préférable pour la portabilité.

Partitions principales

- **system** : installation du système d'exploitation en lecture seule
- **data** : partition où sont principalement les données de l'utilisateur, ainsi les installations d'applications
- **cache** : partition dédiée à des données temporaires
- **sdcard** : stockage externe, généralement sur carte SD ou émulée à partir de **data** à l'aide de FUSE.

Applications

- Distribuées sous la forme APK (*Android Package*)
- Installées dans un « *sandbox* » avec des utilisateurs uniques
- Exécutées dans leurs propres instances de machine virtuelle
- Découpées en une ou plusieurs composantes

Composantes

- **Activités** : représentent un écran avec interface utilisateur
- **Services** : composantes en arrière-plan permettant d'effectuer des tâches continues
- **Fournisseurs de contenu** : permet le partage de ressources communes entre applications
- **Récepteurs de « *broadcast* »** : composantes réagissant à des messages système

Versions du SDK Android

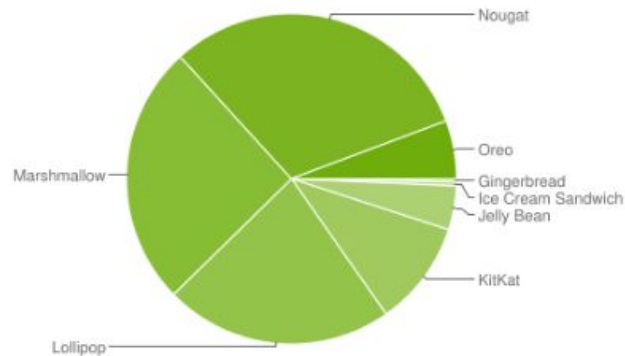
- Nommées avec des noms de pâtisseries et gâteries (*Eclair*, *Gingerbread*, *Icecream Sandwich*, ...)
- Deux désignations numériques:
 - Exemple: *Froyo*, Android version 2.2.x, API Level 8
- Applications peuvent avoir des niveaux d'API cible et minimum différents

Niveau	Version	Nom de code	Lancement	Remarques
1	1.0		23 septembre 2008	HTC Dream
2	1.1	“Petit Four”	9 février 2009	
3	1.5	Cupcake	30 avril 2009	HTC Magic, ajout des widgets
4	1.6	Donut	15 septembre 2009	
5	2.0	Eclair	26 octobre 2009	Motorola Droid, support multi-touch
6	2.0.1	Eclair	3 décembre 2009	
7	2.1	Eclair	12 janvier 2010	
8	2.2.x	Froyo / “Frozen Yogurt”	20 mai 2010	Nexus One, beaucoup d’innovations, ajout de Flash
9	2.3 - 2.3.2	Gingerbread	6 décembre 2010	Nexus S, gains de performance
10	2.3.3 - 2.3.7	Gingerbread	9 février 2011	
11	3.0	Honeycomb	22 février 2011	Motorola Xoom, tablette seulement, améliorations multi-core, changement total de l’interface
12	3.1	Honeycomb	10 mai 2011	
13	3.2 - 3.2.6	Honeycomb	15 juillet 2011	Google TV
14	4.0 - 4.0.2	Ice Cream Sandwich	19 octobre 2011	Galaxy Nexus, les avantages de 3.x pour les téléphones

Niveau	Version	Nom de code	Lancement	Remarques
15	4.0.3 - 4.0.4	Ice Cream Sandwich	16 décembre 2011	Dernière version à supporter Flash
16	4.1 - 4.1.2	Jelly Bean	9 juillet 2012	Nexus 7
17	4.2 - 4.2.2	Jelly Bean	13 novembre 2012	Nexus 4, Nexus 10
18	4.3 - 4.3.1	Jelly Bean	24 juillet 2013	Nexus 7 deuxième génération, OpenGL ES 3.0
19	4.4 - 4.4.4	KitKat	31 octobre 2013	Nexus 5, Android Runtime (ART)
20	4.4W - 4.4W.2	Android Wear	25 juin 2014	Montres intelligentes
21	5.0 - 5.0.2	Lollipop	12 novembre 2014	Nexus 6, Android TV, Material Design, ART, OpenGL ES 3.1, plus de 5000 nouveaux API
22	5.1- 5.1.1	Lollipop	9 mars 2015	
23	6.0-6.0.1	Marshmallow	5 octobre 2015	Nouveau système de permissions
24	7.0	Nougat	22 août 2016	Multi-fenêtre, Vulkan 3D, compilateur JIT avec optimizateur
25	7.1-7.1.2	Nougat	4 octobre 2016	Mises-à-jour A/B, gestionnaire de sauvegarde manuelle
26	8.0	Oreo	21 août 2017	Plus rapide, picture-in-picture, support autre que Google Play

Source : <https://developer.android.com/about/dashboards/index.html>

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.5%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.3%
5.0	Lollipop	21	4.8%
5.1		22	17.6%
6.0	Marshmallow	23	25.5%
7.0	Nougat	24	22.9%
7.1		25	8.2%
8.0	Oreo	26	4.9%
8.1		27	0.8%



Data collected during a 7-day period ending on May 7, 2018.
Any versions with less than 0.1% distribution are not shown.

Cours 1

Le manifeste Android



- Fichier XML obligatoire, situé à la racine de l'application et nommé `AndroidManifest.xml`.
- Ce fichier dit à l'environnement Android comment interagir avec les composantes de votre application.
- Il contient aussi la liste des pré-requis à l'installation de l'application, ainsi que les permissions nécessaires à l'exécution.
- À partir de Android Studio, plusieurs de ses propriétés sont spécifiées par Gradle.

manifest

- Contient l'identifiant de paquet Java dans lequel toutes les classes de l'application sont principalement situées. Il doit être unique d'une application à l'autre.
- **versionCode** est un entier correspondant à la version de l'application. Il est utilisé pour déterminer si une nouvelle version est disponible.
- **versionName** est la valeur affiché à l'utilisateur pour nommer la version.

uses-sdk

- Maintenant déterminé par Gradle.
- Détermine les niveaux d'API Android compatible avec l'application.
 - **targetSdkVersion** est le niveau utilisé pour compiler l'application (présentement 26 pour Oreo).
 - **minSdkVersion** est le minimum de compatibilité que l'application supporte (souvent 19 pour Kit Kat).

uses-permission

- Liste les permissions nécessaire par l'application. L'utilisateur sera présenté avec la liste au moment de l'installation et devra accepter ou non. L'utilisateur sera aussi averti lorsqu'une mise-à-jour a modifié la liste de permissions requises.
- Certaines permissions sont considérées plus dangereuses et seront indiquées dans ce dialogue.
- Depuis Marshmallow, les permissions dangereuses doivent être acceptées à l'exécution.

application

- Le bloque **application** est obligatoire et liste toutes les composantes de l'application.
- **debuggable: true** ou **false** pour permettre le débogage de l'application.
- **icon**: référence à l'icône à utiliser pour l'application.
- **label**: référence à une String contenant une description courte de l'application.
- **theme**: thème globale à utiliser dans l'application.

activity

- Chaque activité de votre application DOIT avoir un bloque **activity** à l'intérieur de **application**.
- **name**: Le nom de la classe.
- **icon, label, theme**: Permettent de spécifier des valeurs spécifiques pour cette activité.
- **configChanges**: Permet de spécifier les types de changement gérés par l'application.
- **screenOrientation**: Permet de spécifier les orientations supportées de l'activité.

intent-filter

- Spécifié dans une activité pour l'exposer à des "*Intents*" particuliers.
- Sert à déterminer quand l'activité sera exécutée
- Doit obligatoirement contenir un élément **action**.
- Peut contenir des éléments **category** et **data**.

- Un **Intent** est similaire à une ligne de commande. C'est un message envoyé à Android qui résulte dans le lancement d'une composante d'application.
- Android détermine quelle activité, service ou groupe de récepteurs, les instancie et démarre.
- C'est de cette façon que les applications sont lancées à partir du menu principale.

- **Un Intent peut contenir:**
 - **Composante:** Le nom de la composante à invoquer (optionnel).
 - **Action:** Le nom de l'action à exécuter.
 - **Données:** Un URI à interpréter.
 - **Catégorie:** Le nom d'un type de composantes pouvant exécuter le message.
 - **Extras:** Un **Bundle** contenant des données optionnelles.
 - **Drapeaux:** Différents drapeaux booléens.

- Par exemple, pour déterminer l'activité qui répondra à l'action d'ouverture à partir du menu principal:

```
<intent-filter . . . >  
  <action android:name="android.intent.action.MAIN" />  
  <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>
```



- Structure constamment en évolution.
- Pour plus d'informations, se référer à:

<http://developer.android.com/guide/topics/manifest/manifest-intro.html>

