

Chapitre 1

Retour en arrière

1. Difficultés du développement pour les systèmes embarqués

Quelques contraintes à prendre en compte :

- Mémoire: il y a moins de mémoire dans un téléphone intelligent que dans un PC.
- La résolution de l'écran et la qualité de l'affichage.
- Gestion des ressources: appel téléphonique pendant le déroulement d'une partie de jeu.
- Optimisation pour une meilleure efficacité.
- Taille de l'écran.
- Disposition des éléments et accès à ces éléments (stylet, gros doigts, etc.).
- Une variété de langues.

2. Pourquoi Android?

Écouter les explications des concepteurs d'Android :

<http://www.youtube.com/watch?v=6rYozIZOgDk>

Le but de départ était de développer un système d'exploitation pour des caméras numériques. Vu que le marché n'était pas assez large, le projet a été détourné pour les téléphones intelligents.

L'idée était donc de créer un système d'exploitation capable d'avoir des outils pouvant interagir avec l'environnement de l'utilisateur (prendre une photo et dire exactement l'endroit où elle a été prise).

3. La philosophie d'Android

- Android est basé sur le système d'exploitation Linux.
- Android est un projet "Open source", distribué gratuitement.
- Le système est développé par l'alliance OHA (Open Handset Alliance) et Google.

<http://www.openhandsetalliance.com/>

http://fr.wikipedia.org/wiki/Open_Handset_Alliance

- Cette alliance regroupe :
 - * Opérateurs de téléphonie mobile
 - * Fabricants de semi-conducteurs
 - * Fabricants d'appareils
 - * Concepteurs de logiciels
 - * Sociétés responsables de la commercialisation

- Android permet le traitement d'une panoplie d'applications natives : téléphonie, courriel, agenda, GPS, multimédia, messagerie, etc.
- L'API Android est à la disponibilité de la communauté des développeurs. Ceci permet l'extension des applications.
- Google Play (Android Market) : est la plateforme sur laquelle un développeur peut vendre son application.

<https://support.google.com/googleplay/android-developer/answer/112622?hl=en>

Si le développeur souscrit à un programme spécial de Google, il perçoit pour le premier million de dollars, 85% du prix de vente et Google 15%. Pour les montants subséquents, le ratio est de 70/30. Si le développeur n'est pas membre de ce programme, le ratio est de 70/30.

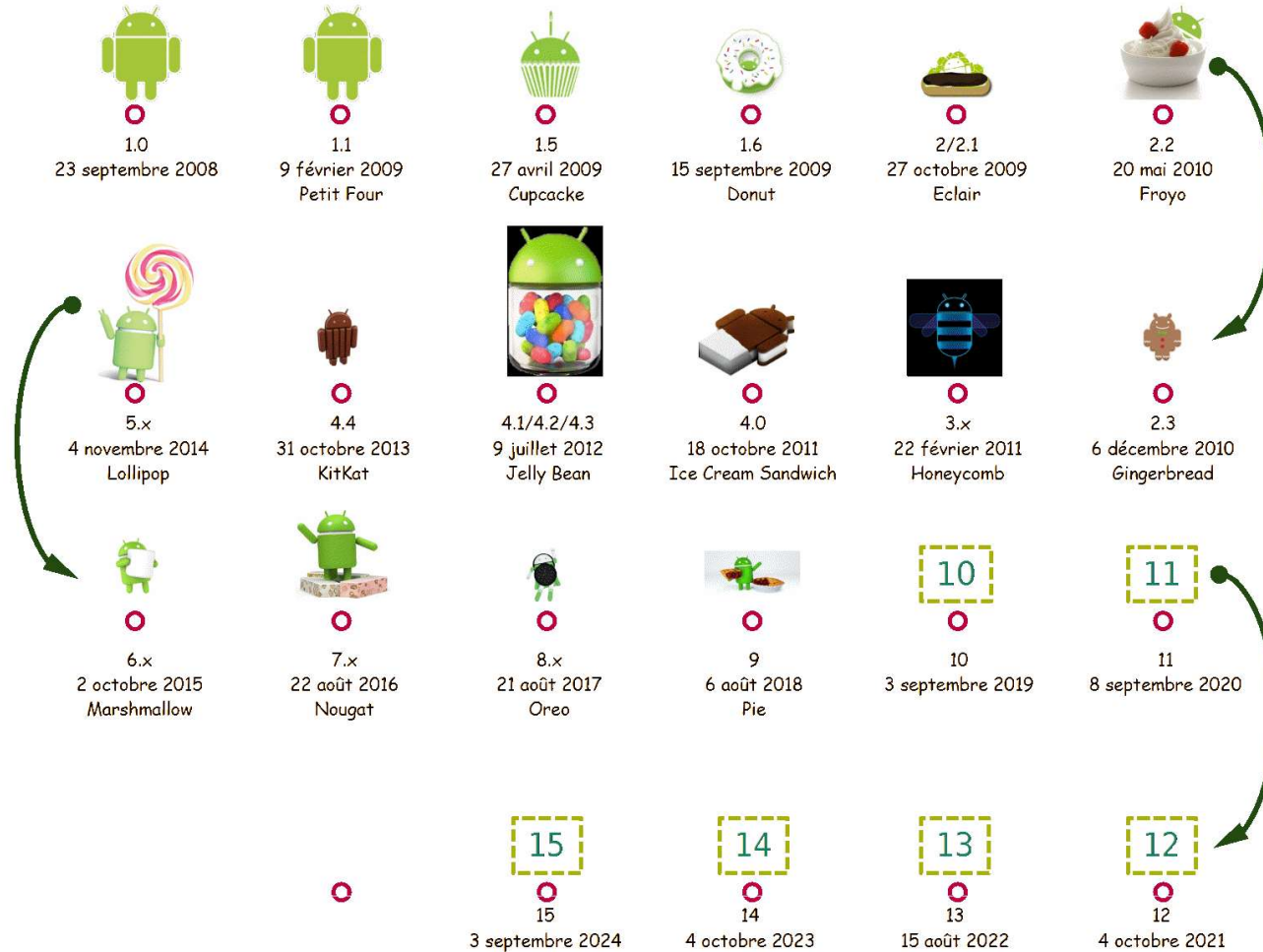
- Ce n'est pas tous les pays qui peuvent vendre des applications sur Google Play :

http://fr.wikipedia.org/wiki/Google_Play

<https://support.google.com/googleplay/answer/2843119?hl=en&rd=1>

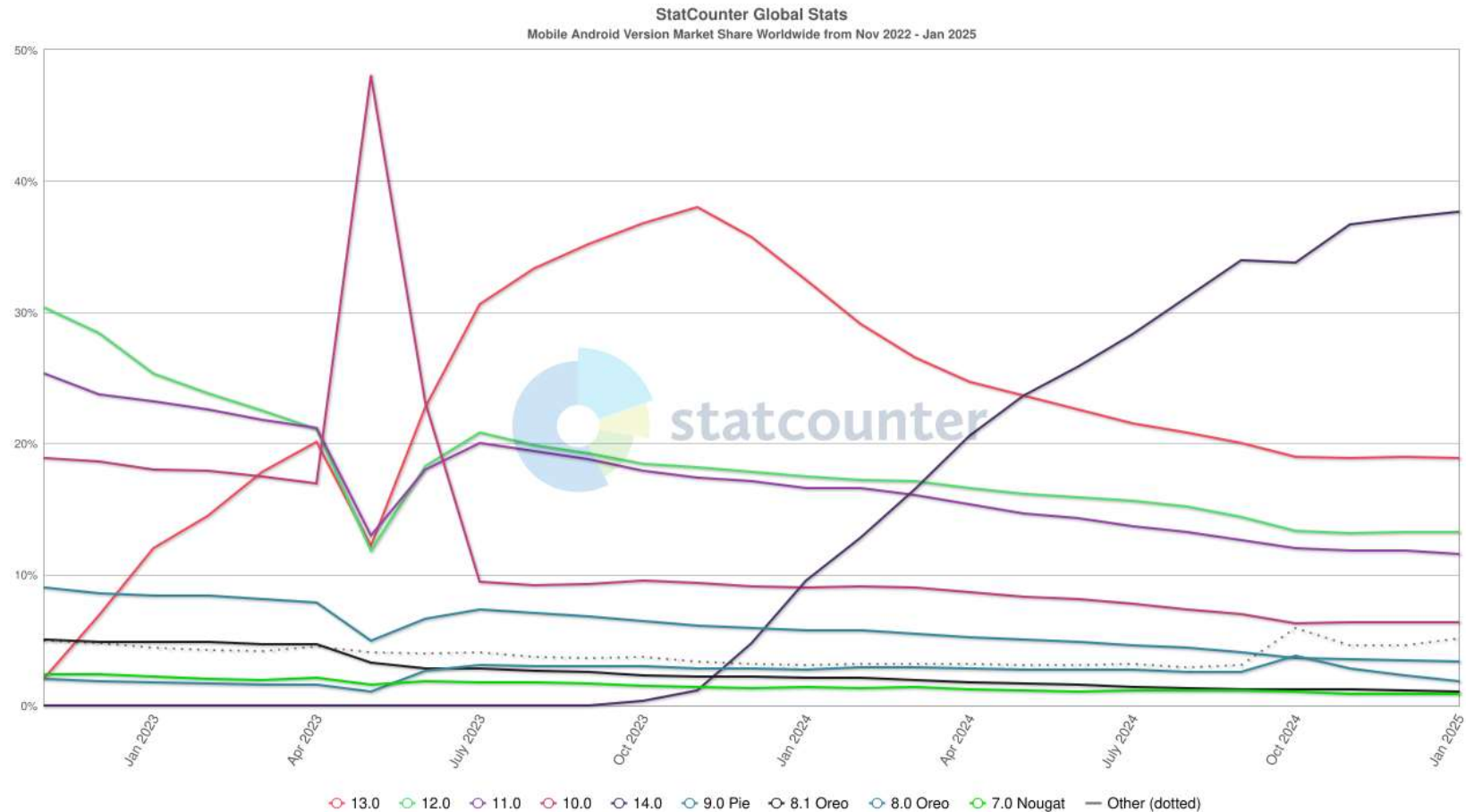
Des contraintes de copyright (films, séries de télévision, livres, etc.) ou logistique (taxes, institutions financières, etc.).

4. L'historique d'Android



https://en.wikipedia.org/wiki/Android_version_history

La répartition de l'installation des versions d'Android :



<https://gs.statcounter.com/android-version-market-share/mobile/worldwide/#monthly-202211-202501>

Pour des statistiques sur un mois :

<http://www.appbrain.com/stats/top-android-sdk-versions>

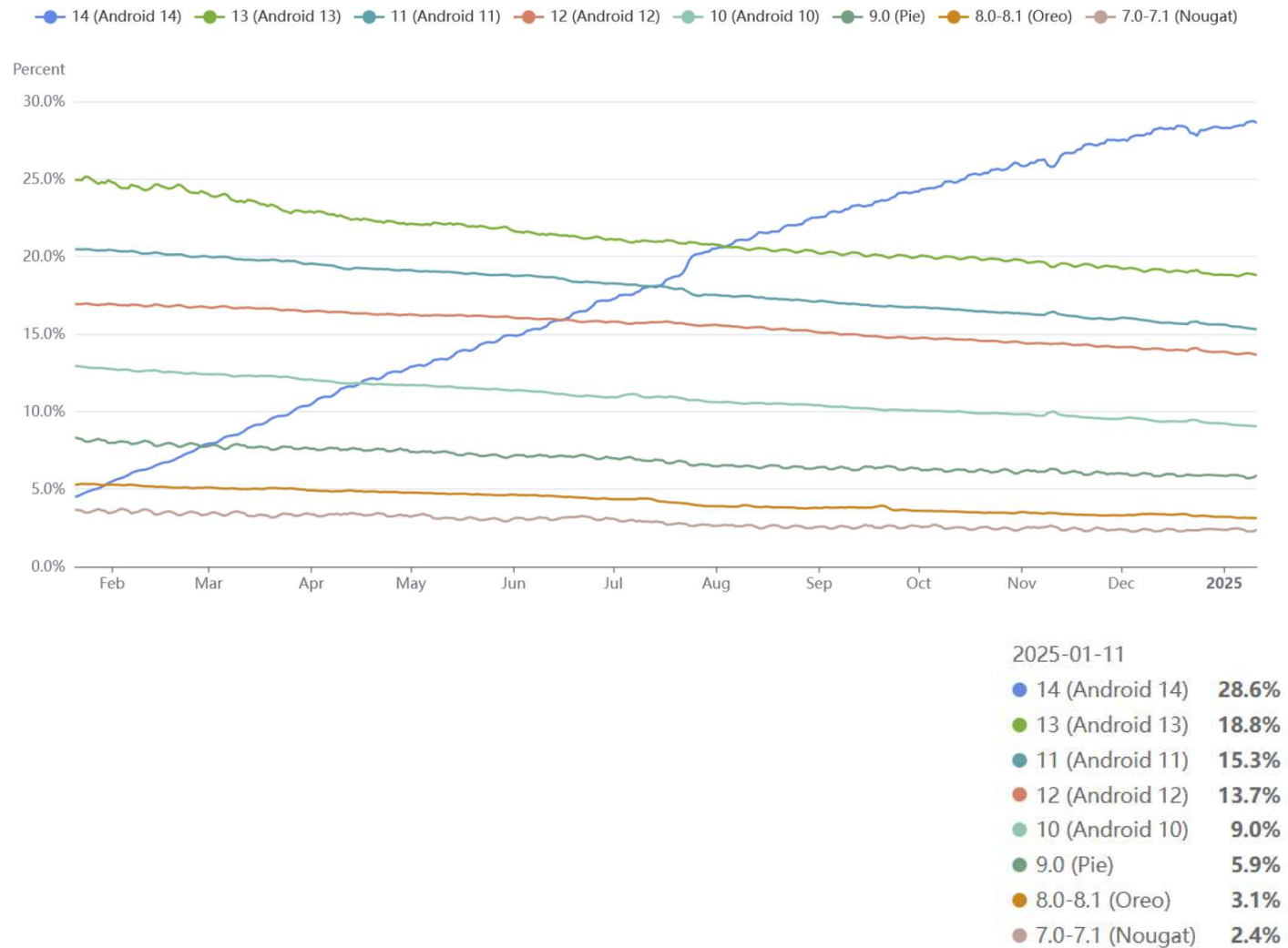
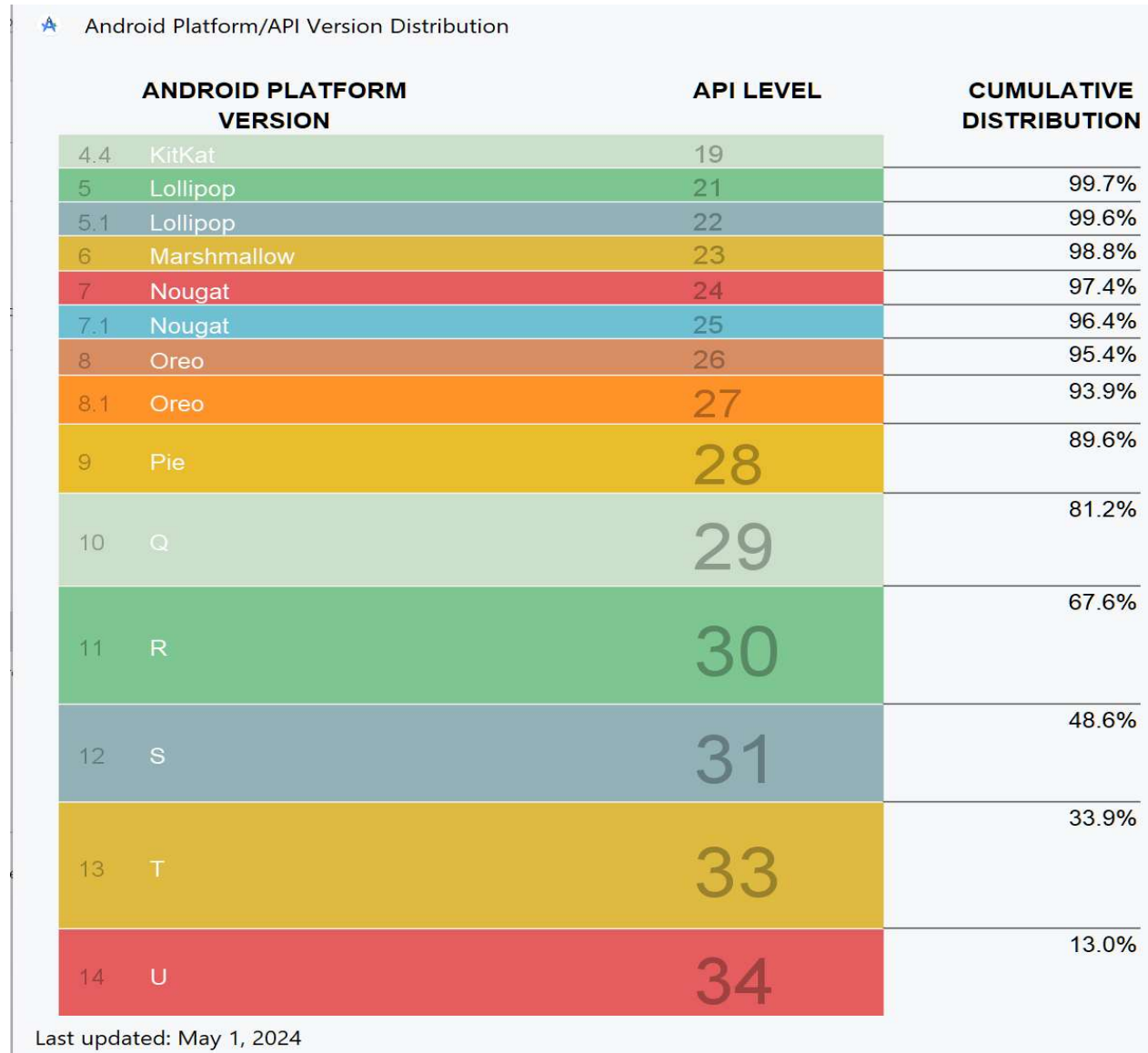


Image capturée le 12/01/2025



Taille et résolution de l'écran

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

Les données ont été collectées sur 1 semaine, le 23 avril 2024.

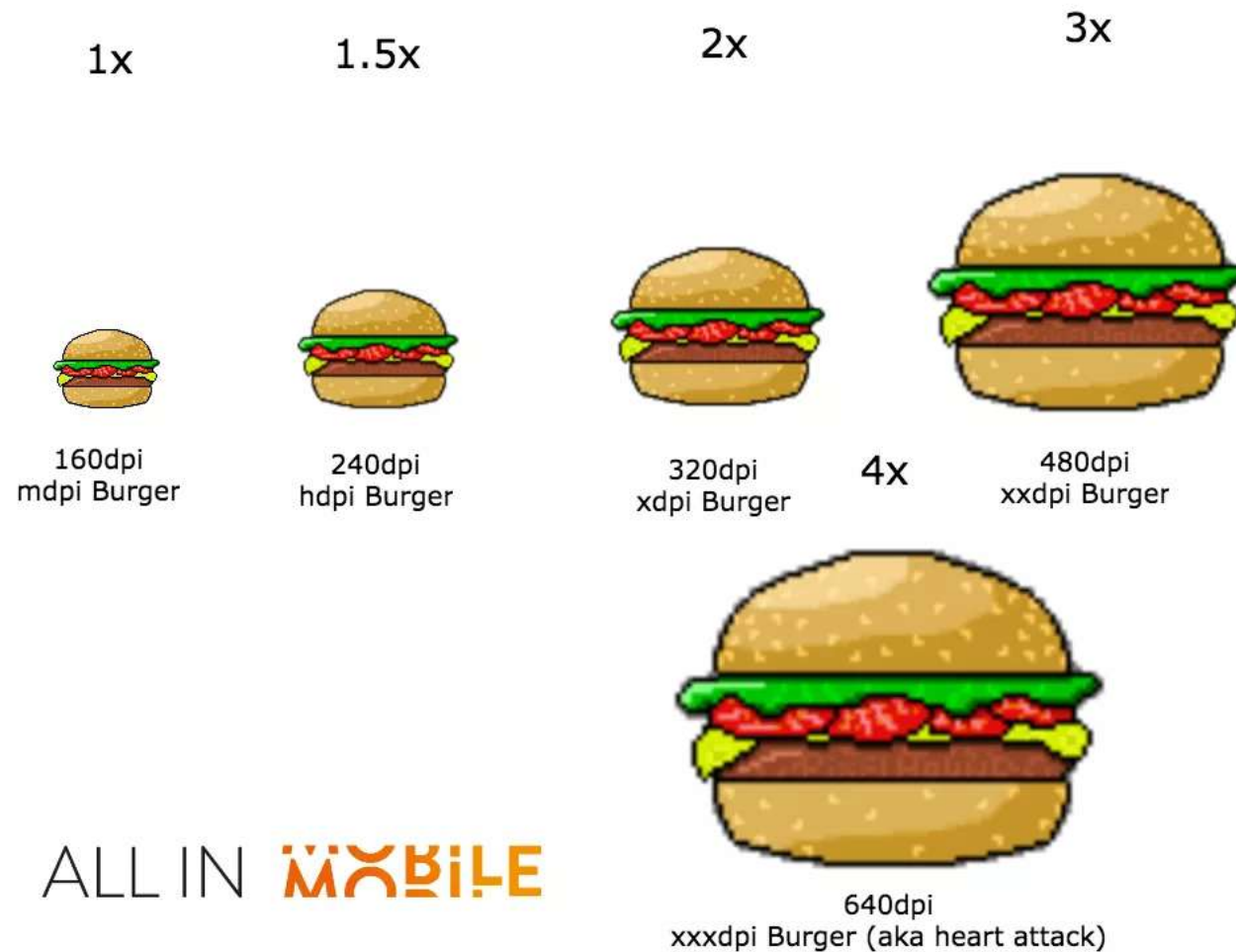
	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total 2017	Total 2018	Total 2019	Total 2022	Total 2023	Total 2024	Total 2025
Small					0.5%	0.1%	1.2%	0.6%	0.4%	0.3%	0.3%	0.4%	0.6%
Normal		0.1%	0.3%	4.3%	44.6%	23.8%	87.6%	89.8%	92.8%	80.6%	79.3%	76.9%	73.1%
Large		1.0%	4.0%	1.0%	9.1%	1.2%	7.2%	6.4%	4.5%	11.5%	12.6%	14.2%	16.3%
Xlarge		5.6%	0.1%	4.0%	0.3%		4.0%	3.2%	2.3%	7.7%	7.8%	8.5%	10.0%
Total	0.0%	6.7%	4.4%	9.3%	54.5%	25.1%							

Normal : 470x320 dp, 480x800 hdpi (720x1200), 640x960 xhdpi (1280x1920)

<http://stefan222devel.blogspot.ca/2012/10/android-screen-densities-sizes.html>

https://developer.android.com/guide/practices/screens_support.html

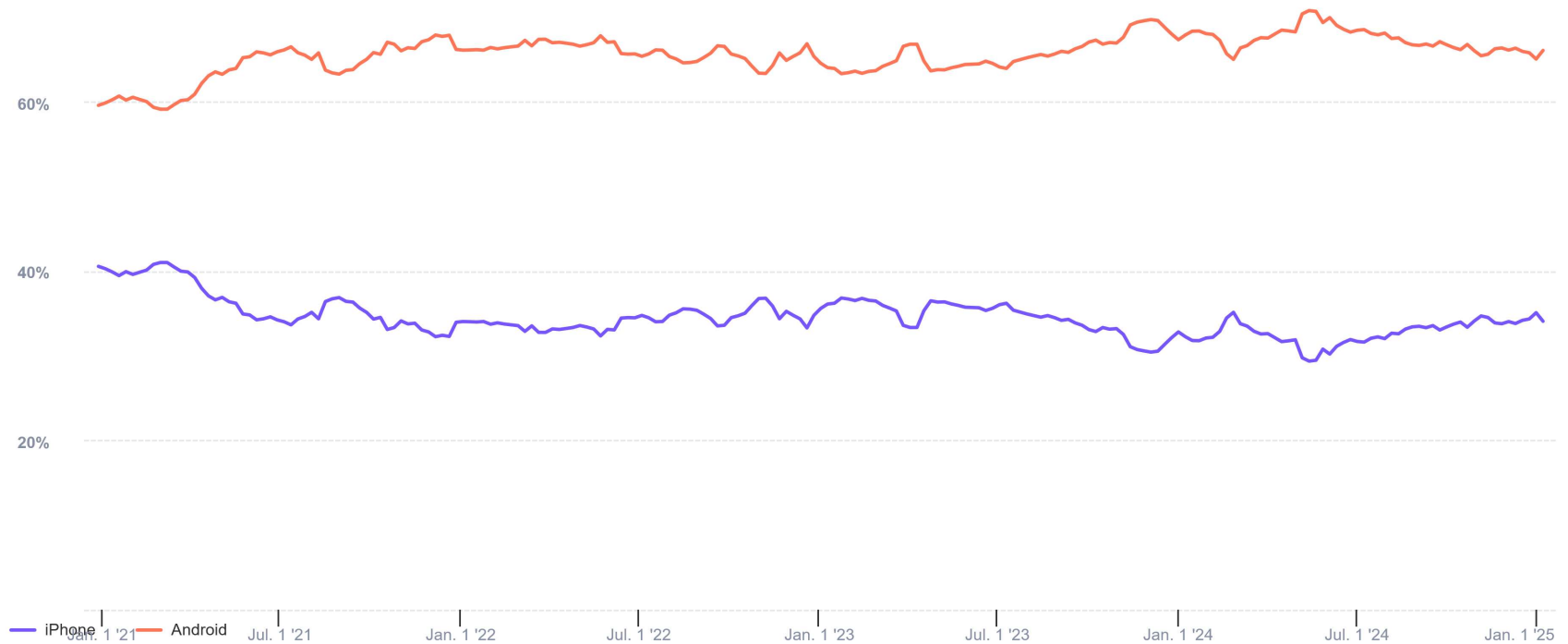
<https://material.io/devices/>



(allinemobile)

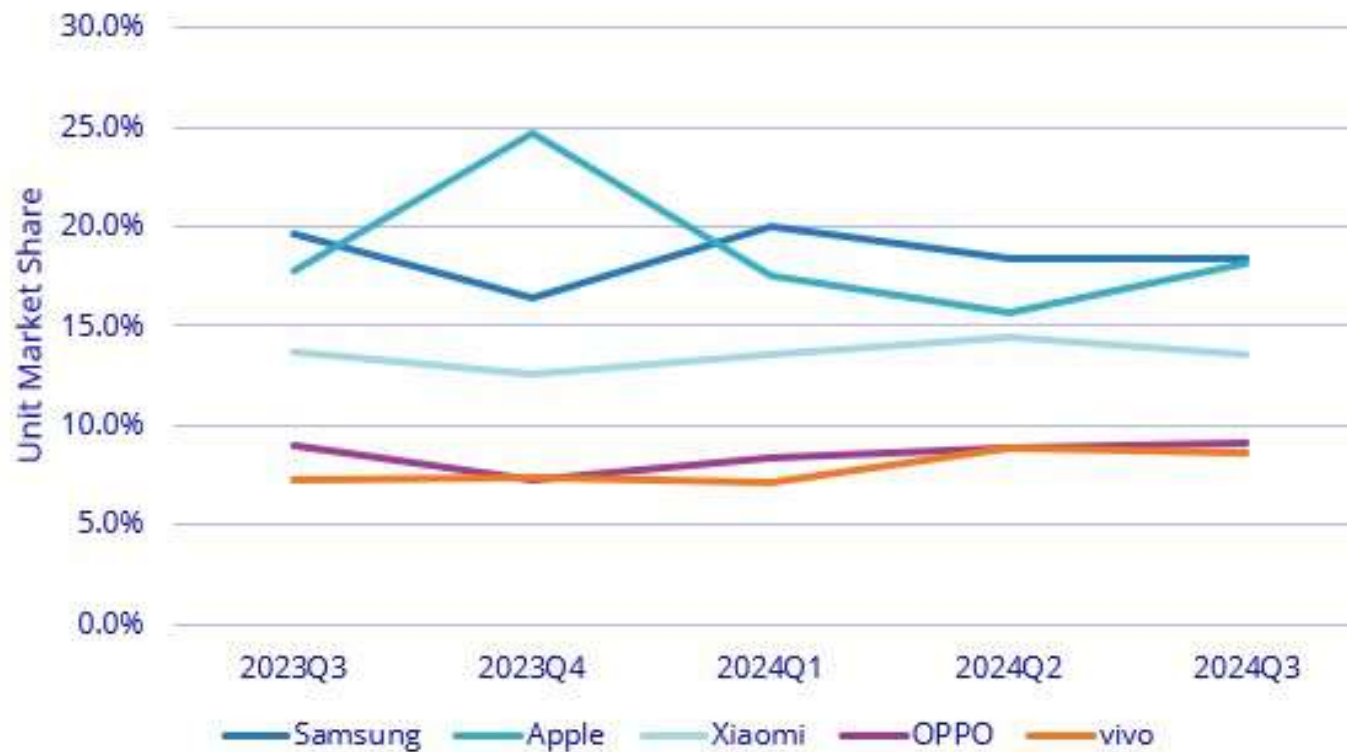
La part de marché pour chaque OS :

https://mixpanel.com/trends/#report/android_vs_ios



La part de marché pour chaque vendeur :

<http://www.idc.com/prodserv/smartphone-market-share.jsp>



6. Android et les outils de développement

- Android Studio

<https://developer.android.com/studio>

- Android SDK

<https://developer.android.com/studio>

L'installateur d'Android Studio par défaut va installer aussi le dernier SDK disponible.

- Ajouter les paquets

<https://developer.android.com/studio/intro/update>

- Cibler la bonne version d'Android

<https://developer.android.com/google/play/requirements/target-sdk>

When you upload an APK, it must meet Google Play's [target API level requirements](#).

Starting August 31 2024:

- New apps and app updates must target Android 14 (API level 34) or higher to be submitted to Google Play; except for Wear OS and Android TV apps, which must target Android 13 (API level 33) or higher.
- Existing apps must target Android 13 (API level 33) or higher to remain available to new users on devices running Android OS higher than your app's target API level. Apps that target Android 12 (API level 31) or lower (Android 10 (API level 29) or lower for Wear OS and Android 11 (API level 30) or lower for Android TV), will only be available on devices running Android OS that are the same or lower than your app's target API level.

You will be able to request an extension to November 1, 2024 if you need more time to update your app. You'll be able to access your app's extension forms in Play Console later this year.

Vous pouvez choisir juste la dernière API avec le strict minimum (plateforme et image API x86), et Android Studio va demander de compléter l'installation au besoin (au moment de l'utilisation). Sinon, si l'espace disque n'est pas un problème, vous pouvez installer dès le départ les API suivantes : 35 (Android 15), 34 (Android 14).

Name	API Le...	Rev...	Status
Android 15.0 ("VanillaIceCream")			
☒ Android SDK Platform 35	35	2	Installed
☒ Sources for Android 35	35	1	Installed
☐ AOSP ATD ARM 64 System Image	35	1	Not installed
☐ AOSP ATD Intel x86_64 Atom System Image	35	1	Not installed
☐ ARM 64 v8a System Image	35	2	Not installed
☒ Intel x86_64 Atom System Image	35	2	Installed
☐ Google APIs ARM 64 v8a System Image	35	8	Not installed
☒ Google APIs Intel x86_64 Atom System Image	35	8	Installed

Name	Version	Status
▼  Android SDK Build-Tools 36-rc3		
☐ 36.0.0-rc3	36.0.0 rc3	Not installed
☐ 36.0.0-rc1	36.0.0 rc1	Not installed
☒ 35.0.0	35.0.0	Installed

Name	Version	Status
▼  Android SDK Command-line Tools (latest)		
☒ Android SDK Command-line Tools (latest)	17.0	Installed

```
[=====] 100% Computing updates...
Installed packages:
```

Path	Version	Description	Location
build-tools;34.0.0	34.0.0	Android SDK Build-Tools 34	build-tools\34.0.0
build-tools;35.0.0	35.0.0	Android SDK Build-Tools 35	build-tools\35.0.0
cmake;3.31.1	3.31.1	CMake 3.31.1	cmake\3.31.1
cmdline-tools;17.0	17.0	Android SDK Command-line Tools	cmdline-tools\17.0
cmdline-tools;latest	17.0	Android SDK Command-line Tools (latest)	cmdline-tools\latest
emulator	35.3.11	Android Emulator	emulator
extras;android;m2repository	47.0.0	Android Support Repository	extras\android\m2repository
extras;google;Android_Emulator_Hypervisor_Driver	2.2.0	Android Emulator hypervisor driver (installer)	extras\google\Android_Emulator_Hypervisor_Driver
extras;google;auto	2.0	Android Auto Desktop Head Unit Emulator	extras\google\auto
extras;google;google_play_services	49	Google Play services	extras\google\google_play_services
extras;google;instantapps	1.9.0	Google Play Instant Development SDK	extras\google\instantapps
extras;google;m2repository	58	Google Repository	extras\google\m2repository
extras;google;market_apk_expansion	1	Google Play APK Expansion library	extras\google\market_apk_expansion
extras;google;market_licensing	1	Google Play Licensing Library	extras\google\market_licensing
extras;google;simulators	1	Android Auto API Simulators	extras\google\simulators
extras;google;usb_driver	13	Google USB Driver	extras\google\usb_driver
extras;google;webdriver	2	Google Web Driver	extras\google\webdriver
ndk-bundle	22.1.7171670	NDK	ndk-bundle
ndk;27.2.12479018	27.2.12479018	NDK (Side by side) 27.2.12479018	ndk\27.2.12479018
patcher;v4	1	SDK Patch Applier v4	patcher\v4
platform-tools	35.0.2	Android SDK Platform-Tools	platform-tools
platforms;android-34	3	Android SDK Platform 34	platforms\android-34
platforms;android-34-ext10	1	Android SDK Platform 34-ext10	platforms\android-34-ext10
platforms;android-34-ext8	1	Android SDK Platform 34-ext8	platforms\android-34-ext8
platforms;android-35	2	Android SDK Platform 35	platforms\android-35
platforms;android-35-ext14	1	Android SDK Platform 35-ext14	platforms\android-35-ext14
skiaparser;2	3	Layout Inspector image server for API S	skiaparser\2
skiaparser;3	5	Layout Inspector image server for API 31-36	skiaparser\3
sources;android-34	2	Sources for Android 34	sources\android-34
sources;android-35	1	Sources for Android 35	sources\android-35
system-images;android-34-ext10;google_apis_playstore;x86_64	2	Google Play Intel x86_64 Atom System Image	system-images\android-34-ext10\google_apis_playstore\x86_64
system-images;android-34-ext8;google_apis_playstore;x86_64	2	Google Play Intel x86_64 Atom System Image	system-images\android-34-ext8\google_apis_playstore\x86_64
system-images;android-34;default;x86_64	4	Intel x86_64 Atom System Image	system-images\android-34\default\x86_64
system-images;android-34;google_apis;x86_64	14	Google APIs Intel x86_64 Atom System Image	system-images\android-34\google_apis\x86_64
system-images;android-34;google_apis_playstore;x86_64	14	Google Play Intel x86_64 Atom System Image	system-images\android-34\google_apis_playstore\x86_64
system-images;android-35-ext14;google_apis_playstore;arm64-v8a	1	Google Play ARM 64 v8a System Image	system-images\android-35-ext14\google_apis_playstore\arm64-v8a
system-images;android-35-ext14;google_apis_playstore;x86_64	1	Google Play Intel x86_64 Atom System Image	system-images\android-35-ext14\google_apis_playstore\x86_64
system-images;android-35;default;x86_64	2	Intel x86_64 Atom System Image	system-images\android-35\default\x86_64
system-images;android-35;google_apis;x86_64	8	Google APIs Intel x86_64 Atom System Image	system-images\android-35\google_apis\x86_64

On vous recommande d'ajouter ces chemins dans la variable « path » de l'utilisateur (et non pas dans celle du système) :

- Raccourcis pour la gestion de l'émulateur :

C:\Users\Votre_Nom_Utilisateur\AppData\Local\Android\Sdk\emulator

Cette commande permet d'afficher la liste des émulateurs.

`emulator -list-avds`

- Raccourcis pour les outils tels que sdkmanager :

C:\Users\Votre_Nom_Utilisateur\AppData\Local\Android\Sdk\cmdline-tools\latest\bin

Cette commande permet d'afficher la liste des librairies installées et celles disponibles.

`sdkmanager --list`

- Raccourcis pour l'outil adb :

C:\Users\Votre_Nom_Utilisateur\AppData\Local\Android\Sdk\platform-tools

Si vous avez besoin du compilateur, et de l'interpréteur Java, installés dans Android Studio et vous n'avez pas une installation locale de Java d'Oracle :

- Raccourcis pour les outils Java :

C:\Program Files\Android\Android Studio\jre\bin

Il vous faut ajouter ces deux variables d'environnement dans votre profil d'utilisateur (et non pas celle du système) :

1-

[JAVA_HOME](#)

C:\Program Files\Android\Android Studio\jre

Si cette variable n'a pas été configurée par une installation locale de JDK, vous pouvez la configurer pour pointer celle d'Android Studio.

2-

[ANDROID_SDK_ROOT](#)

C:\Users\Votre_Nom_Utilisateur\AppData\Local\Android\Sdk

7. Émulateur

Avantage : on peut simuler ce que l'on veut.

Inconvénient: on ne peut pas juger les performances (GPS?) ... ce n'est qu'un émulateur après tout. Il ne peut pas simuler non plus « Bluetooth », « NFC », « carte SD » et les écouteurs.

Pour des tests réels, il n'y a pas mieux qu'un vrai appareil, mais faites attention à la version d'Android qu'il contient pour éviter d'être piégé.

Prendre en exemple :

Medium Phone Google API 35

AVD name:

	Medium Phone	6.4 1080x2400 420dpi	<button>Change...</button>
	VanillalceCream	Android 15.0 x86_64	<button>Change...</button>

VanillalceCream

API Level

35

Type

Google APIs

Android

15.0

Google Inc.

System Image

x86_64 (translated: arm64-v8a)

Pour améliorer les performances de l'émulateur et profiter ainsi de la puissance matérielle de votre machine, assurez-vous d'avoir l'un de ces utilitaires en fonction de l'architecture du processeur :

<https://developer.android.com/studio/run/emulator-acceleration>

Pour lister les émulateurs disponibles (configurés) :

```
emulator -list-avds
```

```
C01_MP_G_API_35
```

Pour démarrer un émulateur (sous Windows uniquement, pour activer tous les paramètres, il faut lancer la ligne de commande en tant qu'administrateur) :

```
emulator -netdelay none -netspeed full -avd C01_MP_G_API_35
```

Si on veut un démarrage à froid (cold boot) :

```
emulator -netdelay none -netspeed full -avd C01_MP_G_API_35 -no-snapshot-load
```

8. Connecter son appareil

Besoin d'un driver?

<https://developer.android.com/studio/run/oem-usb>

9. Se connecter à distance

- Ouvrir un terminal et se positionner dans le répertoire « android-sdks/platform-tools ».
- La commande « adb devices » liste tous les terminaux connectés à votre machine.

```
terminal:~$ adb devices
List of devices attached
emulator-5554    device
```

- Ouvrir un shell à distance sur le terminal dont le nom est « emulator-5554 » :

```
terminal:~$ adb -s emulator-5554 shell
```

- La commande « ls » pour connaître les différentes commandes :

```
C:\Users\lokbani>adb -s emulator-5554 shell
emu64xa:/ $ ls
acct          cache         dev           metadata      proc          system
adb_keys      config        etc           mnt           product       system_dtkm
apex          d            init          odm           sdcard        system_ext
bin           data         init.environ.rc odm_dtkm      second_stage_resources tmp
bootstrap-apex data_mirror  linkerconfig  oem           storage       vendor
bugreports    debug_ramdisk lost+found    postinstall  sys           vendor_dtkm
emu64xa:/ $
```

- La commande « id » pour lister des informations sur l'utilisateur courant :

```
emu64xa:/ $ id
uid=2000(shell) gid=2000(shell) groups=2000(shell),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),
3001(net_bt_admin),3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:shell:s0
emu64xa:/ $ cd lost+found/
/system/bin/sh: cd: /lost+found: Permission denied
2|emu64xa:/ $
```

On n'a pas tous les droits, nous allons demander plus de privilèges, mais attention ...

```
2|emu64xa:/ $ su
emu64xa:/ # id
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),3001(net_bt_
admin),3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:su:s0
emu64xa:/ # cd lost+found/
emu64xa:/lost+found # ls
emu64xa:/lost+found #
```

- Pour obtenir les privilèges « root » sur un émulateur, il faut installer sur ce dernier une image « Google APIs ... » et non pas « Google Play ... ». Cette dernière utilise une image de production d'un appareil donné. Les privilèges « root » sont verrouillés sur une telle image. Vous allez obtenir ce message d'erreur :

```
1|generic_x86:/ $ su
/system/bin/sh: su: not found
```

- Il est possible d'accéder au privilège « root », en utilisant la commande « adb root ». Ajouter « -s » si vous avez connecté plusieurs appareils.

-

```
C:\Users\lokbani>adb -s emulator-5554 root
restarting adbd as root
```

- La prochaine utilisation « adb » se fera en tant que l'utilisateur « root » :

```
C:\Users\lokbani>adb -s emulator-5554 shell
emu64xa:/ # id
uid=0(root) gid=0(root) groups=0(root),1004(input),1007(log),1011(adb),1015(sdcard_rw),1028(sdcard_r),1078(ext_data_rw),1079(ext_obb_rw),3001(net_bt_admin),3002(net_bt),3003(inet),3006(net_bw_stats),3009(readproc),3011(uhid),3012(readtracefs) context=u:r:su:s0
emu64xa:/ #
```

-

- Pour sortir de cet état, il faut faire « adb unroot ».

```
C:\Users\lokbani>adb unroot
restarting adbd as non root
```

- On se connecte à distance sur le terminal avec la commande « telnet » ...

Sous Windows, le client « telnet » n'est pas installé par défaut. Il faut l'activer.

« telnet localhost 5554 » (où 5554 est l'identificateur du port de communication associé à votre terminal).

```
C:\Users\lokbani>telnet localhost 5554
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Android Console: Authentication required
Android Console: type 'auth <auth_token>' to authenticate
Android Console: you can find your <auth_token> in
'C:\Users\lokbani\.emulator_console_auth_token'
OK
```

Mettre la valeur fournie dans le fichier « .emulator_console_auth_token ». Par exemple :

auth 123456789ABCdefZ

Vous allez obtenir ce qui suit :

```
Android Console: type 'help' for a list of commands
OK
help
Android console commands:
  help|h|?
  help-verbose
  ping
  automation
  event
  geo
  gsm
  cdma
  crash
```

L'utilisation d'un jeton caché est nécessaire depuis la version « 25.1.6 » :

[SDK Tools, Revision 25.1.6 \(May 2016\)](#)

Dependencies:

- Android SDK Platform-tools revision 23 or later.

General Notes:

- To improve the security of the Android Emulator and to address a reported security vulnerability, the Android Emulator Console now requires [authentication](#) before you can enter commands. Enter the `auth auth_token` command after you `telnet` to an emulator instance. `auth_token` must match the contents of the `.emulator_console_auth_token` file in your home directory.

- On envoie un message...

```
sms send 5143436111 ceci est un test
```

- On peut même l'appeler...

```
gsm call 5143436111
```

Il y a quelqu'un?

- Si je veux transférer un fichier ... patience!

10. Android Studio et quelques perspectives

Voir le chapitre qui décrit dans les détails l'interface d'Android Studio.

11. Constructeur de phrases « Phrazibus »

- Nous allons construire une application Android et la déployer sur l'émulateur choisi pour cette session.
- L'exemple à utiliser est disponible ici :
<http://www.iro.umontreal.ca/~lokbanicours/ift1176/communs/Cours/ZIP/gui.zip>
- La version « swing » de l'application est le fichier « Phrazibus11.java ». Profitez de l'occasion pour jeter un œil attentif aux notes du cours associé à cet exemple.
- Une nouvelle version de ce fichier est disponible, compilée sous Java 11, sur le site web du cours « IFT1155 ».
- L'application « Phrazibus » a pour but de générer un ensemble de phrases.
- Chaque phrase doit-être composée obligatoirement d'un sujet, d'un verbe et d'un complément d'objet direct.

- Si cette contrainte n'est pas respectée, la phrase ne sera pas affichée.
- Nous ajoutons aussi un bouton qui permettra de remettre l'application à zéro afin de réaliser un test de nouveau.
- Voici le résultat obtenu en Java :



- Voici le résultat obtenu dans un émulateur Android :



- Nous allons d'abord étudier les 3 approches algorithmiques associées à cette application avant d'aborder le développement de l'interface Android. L'ensemble des fichiers est disponible dans le fichier compressé associé à ce chapitre.