



# Systemes d'exploitation

## (2) Du cote utilisateur

### 1. Structuration d'un S.E.

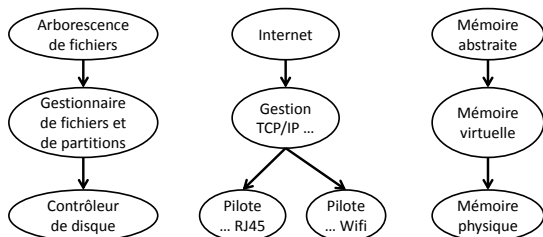
#### Les deux extremes :

- Cote utilisateur :
  - + Interface  
(metaphores de la machine a ecrire, du bureau, ...)
  - + Mecanisme d'abstraction du materiel  
(arborescence de fichiers, ...)
  - + Outils de traitement et de programmation  
(API multimedia, ...)
- Cote machine :
  - + Gestion des peripheriques  
← pilotes
  - + Gestion des ressources materielles  
(processeur, MEV, taches (processus, threads,...))

#### Communication cloisonnee :

- Utilisateur → S.E. : appels systemes
- S.E. → Machine : interruptions

#### Abstraction : exemples



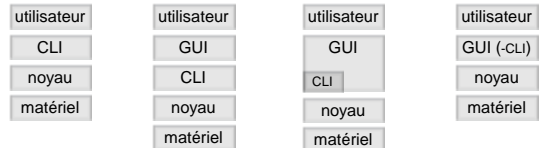
#### S.E. unitaire ou dual :

##### UNITAIRE :

- Un seul programme global pour les deux niveaux  
exemple : MSDOS

##### DUAL :

- Un (des) programme(s) tourné vers l'utilisateur
  - Interface en mode texte (CLI) ou graphique (GUI)
    - + Texte : DOS, Shell Unix/Linux, ...
    - + Graphique : Explorateur Windows, X Window, Gnome/KDE/...
- Un (des) programme(s) tourné vers la machine
  - Noyau : partie assurant les fonctions essentielles du S.E.



## 2. Interface utilisateur en mode texte (CLI)

### 2.1. Unix™/Linux : le shell

Interpréteur (*Command-Line Interpreter*) = superviseur = shell

Choix du shell : **sh** (Bourne: std Posix), csh, tcsh, ksh, **bash** (Bourne again), zsh

Shell convivial : ←, →, BACK, DEL, TAB (autocomplétion), ...

Multi-utilisateur → utilisateurs, groupes et gestion des droits

Multi-tâche → gestion des processus, parallélisme

Console : CLI n plein écran (généralement sur serveur)

Terminal : périphérique, client CLI

Console virtuelle = Emulateur de terminal : Fenêtre CLI dans un GUI

→ TUI (*Text\* User Interface*) : « CLI du point de vue utilisateur »

\* Code ASCII uniquement !

### Commandes std :

**man** : manuel (man *nomcommande*) **info**

whoami : nom de l'utilisateur

who : qui est connecté

cd : changer de dossier

chmod : modifier les droits (fichiers/dossiers)

chown : changer le propriétaire

chgrp : changer le groupe

ln : créer un lien (ln -s *fichieroudossier lien*)

ls : afficher la liste (fichiers/dossiers, options -a -i ...)

pwd : afficher dossier courant

mkdir : créer répertoire

cp : copier (fichiers/dossiers)

mv : déplacer (fichiers/dossiers)

rm : supprimer fichiers ou lien

rmdir : supprimer dossiers

cat : afficher contenu d'un fichier

more, less : affichage paginé

echo : afficher un texte (redirigeable)

### Commandes std (suite) :

env : lister variables d'environnement

set : créer/modifier une variable d'environnement

su : se connecter comme superutilisateur ("root")

→ sudo su

alias : création d'une abréviation

at : exécuter commandes à un moment précis

cmp : comparer deux fichiers

cut : sélectionner une partie de chaque ligne d'un fichier

date : afficher la date (quel format ?)

expr : évaluer une expression logique ou arithmétique

find : chercher dans une arborescence de fichiers

grep : chercher un motif dans des fichiers

paste : fusion de fichier (sortie stdout)

sed : transformation lexicale d'un flux (voir aussi awk)

sort : trier des lignes d'un fichier (ou plusieurs)

touch : modifier la date d'accès et de modification de fichiers

tr : modifier des caractères (de stdin vers stdout)

uniq : éliminer les doublons de lignes successives

wc : compter octets, mots, lignes.

### Commandes std (suite) :

ps : lister les processus actifs

kill : gérer les processus (dont suppression)

fork : créer des processus (fils d'un processus père)

nice, nohup, sleep, tee, trap, wait, ...

gzip : compresser un fichier

gunzip : décompresser

tar : archiver un dossier (cf compression)

← tar-gz : export délicat (dont Windows)

### Entrées/Sorties :

stdin : entrée standard (#0),  
stderr : sortie standard des messages d'erreurs (#1)  
stdout : sortie standard (#2)  
N.B. question de l'ordre des sorties...

### Opérateurs de redirection :

> : redirige stdout vers un fichier (écrase)  
>> : redirige stdout vers un fichier (ajoute)  
< : redirige stdin depuis un fichier  
| : dirige la sortie d'une commande vers l'entrée de la suivante

### Opérateurs d'exécution conditionnelle :

&& : réalise 2<sup>e</sup> commande si 1<sup>e</sup> commande a réussi  
|| : réalise 2<sup>e</sup> commande si 1<sup>e</sup> commande a échoué

### Opérateurs de parallélisation :

& : en fin de ligne de commande

### Métacaractères :

? : un caractère quelconque  
\* : toute suite (éventuellement vide) de caractères  
; : délimiteur de retour de commande  
' : délimiteur de texte  
" : délimiteur de texte  
\$ : récupération de valeur  
\ : caractère d'échappement (\n, \t, ...)  
& : séparateur de commandes  
; : séparateur de cas

### Gestion des dossiers :

/ : séparateur de dossier  
~ : dossier \$HOME de \$USER  
.: dossier courant  
.. : dossier père

### Éditeur CLI de fichier :

- vi : rustique = pas convivial + robuste (successeur de ed), vim, ...
- emacs : peu convivial
- utilisation d'un éditeur GUI sur CLI ← &

serveur sans GUI ?  
→ éditeur CLI originel

### Scripts :

- lot de commandes
- nom de fichier conventionnel : .sh
- en-tête : #! (\*shebang)
- structures de contrôle (if)

### Fichiers de script std :

/etc/profile : ouverture de session, tous utilisateurs  
~/profile  
~/bashrc  
~/bash\_login  
~/bash\_profile  
~/bash\_logout  
...

← utilisation des alias. Exp : alias lsa="ls -lia"

### Utilisation de VI :

Échape : passe du mode insertion au mode commande  
I : passe du mode commande au mode insertion

#### Commandes :

A : insérer texte à la fin de la ligne courante  
D : supprimer la fin de la ligne  
G : curseur vers le fin du fichier  
H : curseur vers le haut de l'écran  
L : curseur vers le bas de l'écran  
O : ouvre une nouvelle ligne au dessus de la ligne courante  
P : paste avant le curseur  
  
a : insérer texte après le curseur  
c : efface des caractères (cv : efface fin mot courant puis passe en mode insertion)  
d : effacer  
dd : effacer ligne courante  
dd! : supprime la chaîne de caractères après le curseur (jq avant !)  
dw : effacer (out) mot après curseur  
I : mode insertion (avant le curseur)  
  
h : curseur vers la gauche  
j : curseur vers le haut  
k : curseur vers le bas  
l : curseur vers la droite  
  
o : ouvre une nouvelle ligne sous la ligne courante  
p : paste après le curseur (voir C)  
r : remplace caractères sous curseur  
u : undo  
w : avancer d'un mot (dw : effacer un mot, 'dhw : 3 mots)  
x : efface caractère sous le (ou à droite du) curseur  
  
!cmd : exécuter la commande shell cmd  
  
q : quitter VI  
q! : quitter, même si modifications non enregistrées  
: :  
:w : enregistrer le fichier (possible : !wq)  
  
:set {set nv} affiche les numéros de ligne / nom / ignorecase / autoident / ruler / showmode / all  
:set {set nv} : création d'une macro



### Variables d'environnement standard :

USER, UID, HOME, PATH, SHELL, ...

### Variables créées :

Mavariab= valeur

### Valeur d'une variable : \$

\$HOME ~  
\$PATH exp : PATH="\$HOME/bin:\$PATH"  
\$\* \$1,...

Exp : ajouter un dossier pour la recherche d'exécutables :

PATH=\$PATH:\$HOME/bin

### Délimiteurs : apostrophe, accent-grave, guillemet

...  
→ Quelle différence de comportement ? d'usage ?  
essayer : echo "\$\n" ; echo "\n" ; echo "\$'\n'" ; echo '\n'

### Shell Unix : gestion des fichiers

#### Reconnaissance du type :

- en-tête « nombre magique » (extension de fichier inutile)  
#, ELF, PK, %PDF, ...
- exécutable : ELF (avant : .OUT)
- mémorisation des nombres magiques dans un fichier /etc/...

#### Gestion des caractères :

- UFT-8 (ou ISO-8859-15...) → comme HTML5
- Retour-ligne : #10 (LF)

#### Arborescence des dossiers : std Posix\* + SUS (Single Unix Specification)

- \* POSIX : documentation et certification payantes → Unix, pas Linux
- Linux : LSB (Linux Standard Base)
- Symbole / → comme le Web  
N.B. : . (dossier courant) .. (dossier père)
- A la racine : /bin, /dev, /etc, /home, /lib, /proc, /temp, /usr, /var, ...  
exp : /var/spool/mail,
- Principe du "tout est fichier" → stderr, stdout, stdin
- ← importance du langage C

### Gestion des droits : (cf super-utilisateur)

- chmod, chown : utilisateur/groupe/tous

### BOM ou pas ?

- Indicateur de l'ordre des octets → UTF-16, UTF-32
- Absent des anciens Unix
- Inutile pour UTF-8 mais...
- Début de fichier (cf nombre magique)  
EFBBBF (utf-8), FEFF (utf-16 big endian), FFFE (utf-16 little endian), ...

### Shell Unix : gestion des utilisateurs

#### Mode super-utilisateur :

nom= root  
- su,  
- sudo

#### Utilisateurs :

- nom : *login* et mot de passe : *passwd*
- Identificateur : *UID* (su: 0)
- espace réservé : ~ = \$HOME (dans /home/... ou /usr/home/...)
- fichiers du S.E. "dot" : .login, ....

### Shell Unix : gestion des processus

**Processus** : unité de programme en cours d'exécution,  
chargée en MEV (instructions + données, dont pile)

#### Shell :

- PID
- ps, kill

#### Programmation :

- fork : processus père → processus fils

### Shell Unix et réseaux

#### Connexion à distance :

- telnet, ssh, ...

#### Outils :

- ping, traceroute, ...

#### Transfert de fichiers :

- ftp

#### Courriel :

- mail
- ~/.forward, /var/spool/mail/\$USER

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.3. OS X

L'ancêtre Mac OS : métaphore du bureau

- Lisa (1983), Apple II (1977), Macintosh (1984)
  - interface graphique avant les PC
  - multitâche coopératif
- processeurs Motorola 680x0 puis PowerPC

Evolution linéaire : 10.0 (2001) ... 10.11 ...

- multitâche préemptif et protection mémoire
- processeurs PowerPC puis Intel-64 (≥2006)
  - ← "universal binaries"
- Apple Store, nombreux outils

Interface graphique : Aqua (+Spotlight + QuickLook +...)

- une constante : l'ergonomie
- fonctionnalités [à interface] graphiques :
  - dock, finder, ...
- Moteur graphique : Quartz (Core Framework + Graphics & Media Layer)
  - ← OpenGL pour 3D & QuickTime pour multimedia

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.4. iOS ← iPhone OS (XXI<sup>e</sup> s.)

Tablettes et smartphones : bureau tactile avec icônes

- Héritages d'OS X (Dock, Safari, ...) et convergences (Siri, ...)
- Une constante : l'ergonomie
- Des contraintes et des a priori :
  - pas d'accès à la hiérarchie des dossiers
  - isolation des applications
  - applications autorisées seulement (pas Flash Player)

Interface graphique : Springboard

#### 3.5. Android (2005)

Tablettes et smartphones :

- un Linux à interface graphique iOS-like ?
- Bibliothèque 2D/3D : OpenGL
- Diversité : surcouche propriétaire selon appareil
- (Google) Store

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.6. Quelle interface graphique ?

Homogénéité :

- métaphore du bureau (tactile ? "Modern UI" ?)
- gestion des fenêtres : ouvrir, fermer, agrandir, maximiser, réduire, ...
- fenêtres filles
- apparence plus élaborée (transparence, ...)

Diversité :

- position des boutons (haut-gauche / haut-droit)
- taille des bordures
- souris et/ou tactile (une période charnière ?)
- positionnement et taille automatique ?



## Shell Unix : scripts

### Boucle précontrôlée : TANTQUE

```
while condition
do ...
done
```

### Boucle postcontrôlée : REPETER

```
until condition
do ...
done
```

### Sortie prématurée de boucle :

`break`

### Saut en fin de boucle :

`continue`

## Shell Unix : scripts

### Choix multiples : SELON

```
#
x=$1
case $x in
'Zaza')
echo "a"
;;
'Zoe')
echo "a"
;;
'Zefir')
echo "i"
;;
*)
echo "y"
;;
esac
```

```
# Application : operations inductives
# d'apres Baranger & Schomaker
#
if [ $# -lt 2 ]; then
echo "nombre d'arguments insuffisant"
echo "syntaxe : $0 operateur nombre1 ... nombreN"
exit 1
fi
else
operateur=$1
shift
case $operateur in
'x')
resultat=$1
echo $resultat
shift
while [ $1 ]; do
resultat=expr $resultat + $1
shift
done
echo "somme : $resultat"
;;
'y')
resultat=$1
shift
while [ $1 ]; do
resultat=expr $resultat \ * $1
shift
done
echo "produit : $resultat"
;;
esac
fi
```

## Shell Unix : scripts - éviter les boucles...

### Chercher dans un fichier : grep

```
grep Exercice tp1.htm
```

```
<br>Exercice 0.<br>
<br>Exercice 1.<br>
<br>Exercice 2.<br>
<br>Exercice 3.<br>
<br>Exercice 4.<br> (sort, uniq, wc)
```

```
grep -E "^<br>$^" tp1.htm
```

```
<br>
<br>
<br>
...
```

métacaractères : ^ \$ [] | ? \* + . ()  
options : -i, -n, -r

### Chercher des fichiers/dossiers : find

```
find -/.. -type d -name a"
```

```
alsigayr
```

```
...
```

### Sélectionner des champs : cut

```
find ~ | cut -d"/" -f4
```

```
find ~ -type f -printf "%p -%u"
```

```
grep -E Exercice `find - -name *.htm`
```

### Remplacer des caractères : tr

```
tr a-z AZ < pays.txt
```

## Shell Unix : scripts - entrée/sortie

```
echo bonjour
echo "bonjour"
echo -n bonjour
```

→ Création de menus

```
echo "votre nom : "
read nom
echo "votre prenom : "
read prenom
echo "bonjour $prenom $nom"
```

→ gestion de fichiers

```
echo "1ere ligne de texte" > texte.txt
echo "2eme ligne de texte" >> texte.txt
```

```
read ligne < texte.txt
```

```
for mot in `more texte.txt` ; do
echo mot
done
```

```
...
```

## Shell Unix : scripts – fonctions

```
function existence {  
  if [ -f $1 ]  
  then  
    echo "$1 existe"  
  else  
    echo "$1 n'existe pas"  
  fi  
}
```

## Shell Unix : scripts – Commandes ++

```
basename,  
clear,  
cmp,  
crypt,  
dd,  
df,  
diff,  
dirname,  
du,  
dump,  
file,  
fg,  
head,  
hostname,  
jobs,  
line,  
logname,  
lp,  
mail,  
mesg,  
...
```

## Shell Unix : scripts – Commandes ++

```
...  
mktmp,  
newgrp,  
news,  
nice,  
nohup,  
od,  
paste,  
passwd,  
pr,  
renice,  
sleep,  
split,  
stty,  
tabs,  
tail,  
tee,  
test,  
top,  
trap,  
...
```

## Shell Unix : scripts – Commandes ++

```
...  
tty,  
umask,  
uname,  
which,  
wait,  
write,  
...
```

## 2. Interface utilisateur en mode texte (CLI)

### 2.2. MSDOS : (Microsoft Disk O.S.) 1981-2000 (+Win XP Dsk)

#### Fichiers systèmes :

- IO.SYS charge MSDOS.SYS et traite CONFIG.SYS pour charger un interpréteur de commande (shell=COMMAND.COM)

#### Fichiers :

- Noms en lettres majuscules (et chiffres) 8.3
- Type de fichier : déterminé par l'extension du nom
- Symbole de dossiers : \ ('/' paramètres de commande)
- Attribut : système / caché / lecture seule
- Table de caractères : page 437 (US), 850 (FR), ... → pas ISO !
- Retour à la ligne : CR+LF (#13#10)

#### Commandes :

- cd, chkdsk, copy, del, deltree, dir, edit, fdisk, format, help, keyb, md, type, xcopy, rd, ...
- + symbole de paramètres / au lieu de - s

#### Scripts :

- Fichiers .BAT (dont autoexec.bat)
- Rôle de Basic

## 2. Interface utilisateur en mode texte (CLI)

### 2.3. OS X :

#### GUI-oriented O.S. :

- héritage de Mac OS (pas CLI) et de NextStep (proche Unix)

#### Terminal :

- Menu Applications/Utilitaires/Terminal
- Noyau avec conformité Unix-Posix
  - commandes Unix : man, ls, ...
  - scripts Unix + AppleScript
  - codage\* UTF-8, retour ligne LF (#10, N.B. Mac OS : CR)
  - ← Mais exécutable Mach-O et pas ELF

\* "Décoder BOM des fichiers extérieurs, générer BOM pour utf16 seulement, utiliser les types natifs en interne."

### 2.4. IOS :

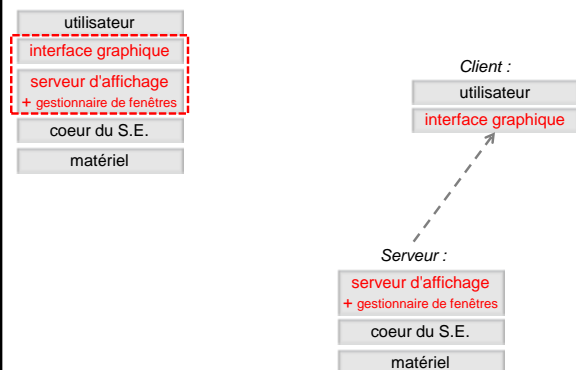
#### GUI + Tactile :

- Issu de OS X (même structure que OS X)

#### Mais :

- Fermeture (sources) et restrictions
- PAS de CLI → jailbreak ?

## 3. Interface utilisateur en mode graphique (GUI)



## 3. Interface utilisateur en mode graphique (GUI)

### 3.1. Unix

#### X Window System (=X11) :

- sous Linux : X.org (successeur de Xfree86)
- Mode client-serveur : logiciel client délègue affichage à serveur X11
  - possibilité de terminal X ("client léger") + serveur central
- Multi-plateforme
  - serveur X.org pour Cygwin sous Windows
- Moteur graphique basé sur X Protocol
- Bibliothèque de fonctions Xlib gérant X Protocol
- Toolkits (API) utilisant Xlib : Qt, GTK+, X-Toolkit...

#### Interface graphique :

- KDE (basé sur Qt)
  - apparence Windows ?
- Gnome, Xfce, ... (basés sur GTK+)
  - Gnome 3 : la rupture ?



Lancer l'interface graphique en mode CLI : startx



### 3. Interface utilisateur en mode graphique (GUI)

#### 3.2. Windows

##### Trois lignées :

- 1<sup>o</sup> Win. 3.1 (1985) : 16 b + MSDOS 16 b
  - dualité problématique
- 2<sup>o</sup> Win. 9x (1995) : 32 b + MSDOS 16 b → 95, 95, 98, Me
  - accord d'exclusivité avec constructeurs de PC
- 3<sup>o</sup> Win NT (1993) : multi-tâche+préemptif+multi-utilisateur+multi-processeur
  - ← OS/2
  - Win. NT → NT3, NT4, 2000 (NT5)
  - Win. Server → 2003 (NT5), 2008 (NT6), 2012...
  - Win. Xp (NT5), Vista (NT6), 7, 8, 8.1
    - ← Win. 7 : le début de la maturité ? mais ...
    - ← Win. 8 : interface orientée tactile, Windows Store, PAS menu Démarrer
- Win. 10 : nouveau noyau
  - convergence ordinateur-smartphone, voix, MAJ continue obligatoire, ...
  - Windows Mobile / Win. Phone / (Win. CE)

+ versions successives (OSR/SP/R2/...)

+ versions segmentées (familial, professionnel, entreprise, ...)

+ processeurs Intel (récemment : ARM)

© 2016 - A. Sigayret

33

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.2. Windows

##### Interface graphique : "explorateur Windows" (ancien : Program Manager)

- Bureau avec icônes, barre des tâches, (menu Démarrer), ...
- console : cmd.exe (lignée Windows NT)
- Moteur graphique (Bibliothèque 2D3D) : Direct3D (≠DirectX)
- Question de l'ergonomie discutable
  - Exp. Positions des fenêtres
  - Internet Explorer (réseau) ↔ Explorateur de fichiers (local)
- Question des réglages par défaut
  - Exp. Masquer les extension des fichiers

##### Accessoires obligés :

- Internet Explorer → Edge ?

##### Initiatives :

- Par défaut : txt → Notepad, rtf → Wordpad
- + installation de MS Office : rtf → Word

© 2016 - A. Sigayret

34

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.2. Windows

##### Windows et MSDOS :

- page de code windows-1252 ≠ MSDOS
- Exp. cmd → dir > tmp.txt → éditer tmp.txt

##### Contenu du dossier :

```
Dossiertemp
tmp.txt
zipit.gif
aaaç.txt
ééèè.txt
```

##### Affichage :

Le volume dans le lecteur C s'appelle Win8  
Le num,ro de s,rie du volume est 0407-FFFF

```
R,pertoire de C:\Users\lrv\_data
29/02/2016 09:55 <DIR> .
29/02/2016 09:55 <DIR> ..
29/02/2016 09:55 0 tmp.txt
27/01/2014 18:33 7y331 zipit.gif
29/02/2016 09:02 <DIR> Dossiertemp
29/02/2016 09:54 0 ..f,.t.txt
29/02/2016 09:55 0 ..$%.t.txt
4 fichier(s) 7y331 octets
4 R,p(s) 397y893y947y392 octets libres
```

© 2016 - A. Sigayret

35

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.2. Windows

##### Console Linux sous Windows : Cygwin

- Linux RedHat
- commandes shell Linux
- GUI X11 possible
- exécutable Windows (pas Linux)

© 2016 - A. Sigayret

36

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.3. OS X

L'ancêtre Mac OS : métaphore du bureau

- Lisa (1983), Apple II (1977), Macintosh (1984)
  - interface graphique avant les PC
  - multitâche coopératif
- processeurs Motorola 680x0 puis PowerPC

Evolution linéaire : 10.0 (2001) ... 10.11 ...

- multitâche préemptif et protection mémoire
- processeurs PowerPC puis Intel-64 (≥2006)
  - ← "universal binaries"
- Apple Store, nombreux outils

Interface graphique : Aqua (+Spotlight + QuickLook +...)

- une constante : l'ergonomie
- fonctionnalités [à interface] graphiques :
  - dock, finder, ...
- Moteur graphique : Quartz (Core Framework - Graphics & Media Layer)
  - ← OpenGL pour 3D & QuickTime pour multimedia

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.4. iOS

← iPhone OS (XXI<sup>e</sup> s.)

Tablettes et smartphones : bureau tactile avec icônes

- Héritages d'OS X (Dock, Safari, ...) et convergences (Siri, ...)
- Une constante : l'ergonomie
- Des contraintes et des a priori :
  - pas d'accès à la hiérarchie des dossiers
  - isolation des applications
  - applications autorisées seulement (pas Flash Player)

Interface graphique : Springboard

#### 3.5. Android (2005)

Tablettes et smartphones :

- un Linux à interface graphique iOS-like ?
- Bibliothèque 2D/3D : OpenGL
- Diversité : surcouche propriétaire selon appareil
- (Google) Store

### 3. Interface utilisateur en mode graphique (GUI)

#### 3.6. Quelle interface graphique ?

Homogénéité :

- métaphore du bureau (tactile ? "Modern UI" ?)
- gestion des fenêtres : ouvrir, fermer, agrandir, maximiser, réduire, ...
- fenêtres filles
- apparence plus élaborée (transparence, ...)

Diversité :

- position des boutons (haut-gauche / haut-droit)
- taille des bordures
- souris et/ou tactile (une période charnière ?)
- positionnement et taille automatique ?

