

SFFS - 2 décembre 2008

## Le multimédia embarqué :

ANDROID

Laurent Bonnet  
Fabien Devos  
Manuel Garnier  
Pierre Viau



Encadrant :  
Marie-Jo Pedrono

INSA Rennes - Département informatique

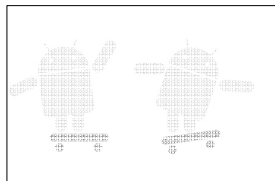
## Plan

- Systèmes embarqués
  - Généralités
  - L'embarqué multimédia
  - Les smartphones
- Android
  - Présentation
  - Technologies
- Programmer sous Android

Le multimédia embarqué : Android

2

## Les systèmes embarqués



Les systèmes embarqués

## Généralités (1/2)

- Définition
  - Electronique et informatique
  - Autonome
  - Mobile
- Caractéristiques
  - Ressources limitées
    - Mémoire
    - Puissance de calcul
    - Source d'énergie
  - Entrées/Sorties spécifiques
    - écran, touches, ...



Le multimédia embarqué : Android

4

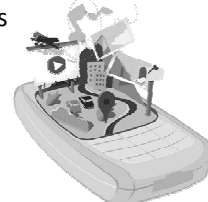
## Domaines d'applications

- Industrie
    - Transports
    - Militaire
    - Robotique
  - Grand public
    - Consoles de jeu portables
    - PDA
    - Téléphonie
- Vers le multimédia



## La problématique de l'embarqué multimédia

- Applications multiples
  - Internet
  - GPS
  - Video
  - Musique
  - Communication
- ... dans le creux de la main !



## La problématique de l'embarqué multimédia

- Contraintes matérielles
  - Evolution rapide
  - Ressources restent limitées
    - Mémoire vive : dizaines de Mo
    - Processeurs : centaines de Mhz
    - Résolution d'écran : 300x500 pixels environ
- Bien moins que pour un PC
  - Mais demande comparable !

## La problématique de l'embarqué multimédia

- Contraintes logicielles
  - SDK propriétaires
  - Langage souvent imposé et non standard
  - Bas niveau



## Exemple : console nintendo DS (1/2)

- Matériel
  - Deux CPUs (66 et 33 Mhz)
  - Mémoire
    - RAM : 4 Mo
    - RAM Video : plusieurs banques (3D : 3x128 Mo max)
  - Deux écrans (256x192 pixels chacun) dont un tactile
  - Wifi, microphone



## Exemple : console nintendo DS (2/2)

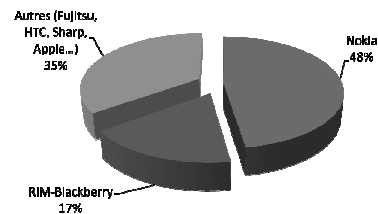
- Logiciel
  - SDK propriétaire de Nintendo, bibliothèques relativement bas niveau
  - Kit de développement et documentation payante
  - Langage C ou C++
- Exemple de contrainte : chargement en Vram
  - Chargement des données en temps limité (Vblank)
  - « Vector » trop lent : utilisation d'un tableau

## Les smartphones (1/5)

- Téléphone portable couplé avec un PDA
  - Agenda, navigation Web, messagerie instantanée, musique, vidéo, GPS...
- Pour l'entreprise à l'origine
  - Vers le grand public (iPhone)
- 2005 : 5% des téléphones portables
- 2009 : 25% prévu

## Les smartphones (2/5)

- Le marché des smartphones (2008)



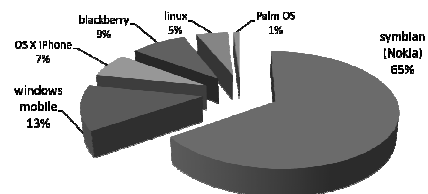
## Les smartphones (3/5)

- OS dédiés
  - Nokia & co. : Symbian
  - Palm : Palm OS
  - Apple : iPhone OS
- OS génériques
  - Microsoft : Windows Mobil
  - Google : Android
    - Le seul open source et gratuit !



## Les smartphones (4/5)

- Le marché des OS pour smartphones (2007)

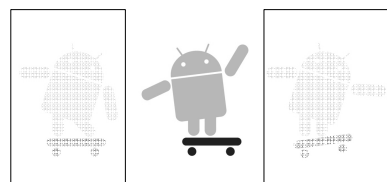


## Les smartphones (5/5)

- L'évolution matérielle
  - Sony Ericsson P800 (2002)
    - 200 Mhz – 12 Mo RAM
  - Palm Treo 650 (2005)
    - 312 Mhz – 32 Mo RAM
  - HTC G1 (2008)
    - 528 Mhz – 192 Mo RAM
    - Le premier sous Android !



## Android



## Historique

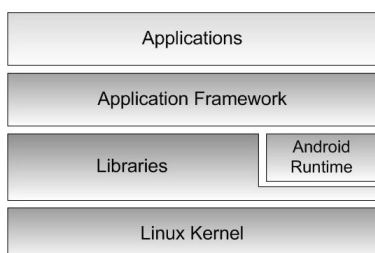
- 2005 : Google rachète la start-up Android Inc.
- 2007 : Open Handset Alliance
  - Motorola, Samsung, HTC, Intel, Nvidia, TI...
  - Promouvoir Android, système d'exploitation ouvert
- Sept. 2008 : SDK 1.0
- Oct. 2008 : sortie du HTC G1

## Un OS ouvert

- License majoritairement Apache 2.0
- Open source
  - <http://source.android.com/download>
- Forte communauté de développeurs
  - SDK mis à disposition
  - Concours d'applications
    - 10 M\$ pour 50 gagnants



## Architecture (1/8)



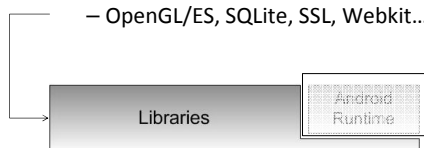
## Architecture (2/8)

- Pilotes
  - Wifi, USB, audio, video, affichage...
- Gestion de l'alimentation



## Architecture (3/8)

- Bibliothèques
  - OpenGL/ES, SQLite, SSL, Webkit...



## Architecture (4/8)

- Système d'exécution Android
  - Bibliothèque principale
  - Machine virtuelle (Dalvik)



## Architecture (5/8)

- Gestionnaires
  - Fenêtres, ressources, téléphonie, contenu, localisation...



## Architecture (6/8)

- Pour l'utilisateur final
  - Agenda, navigateur web, communication, lecteur multimédia...

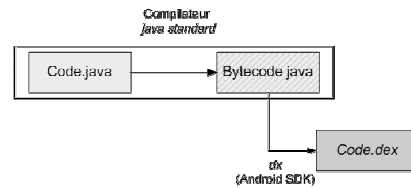


## Architecture (7/8)

- Machine virtuelle Dalvik
  - Pour exécuter du code java
  - Réduite pour libérer de l'espace en mémoire
  - Optimisée pour demander moins de ressource
  - Pas de compilateur *just-in-time*
  - Plusieurs instances de VM à la fois

## Architecture (8/8)

- Machine virtuelle Dalvik vs machine virtuelle java



- Permet à Google d'utiliser java standard
  - ...mais sans la VM *mobile edition* de Sun

## Exemples d'application (1/3)



## Exemples d'application (2/3)



## Exemples d'application (3/3)

**enkin** 



Le multimédia embarqué : Android

29

## Le côté obscur

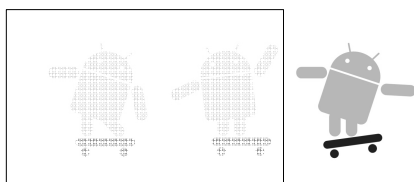
- Un autre tentacule pour la pieuvre Google ?



Le multimédia embarqué : Android

30

## Programmer sous Android



## Moyens techniques

- Technologies
  - Java restreint (java.util, .lang, .security...)
  - XML pour l'interface graphique
  - Bibliothèques (base de données, 3D...)
- SDK
  - Plug-in pour Eclipse
  - Outils de debug
  - Emulateur complet (Internet, carte SD, GPS...)

Le multimédia embarqué : Android

32



## Une application Android

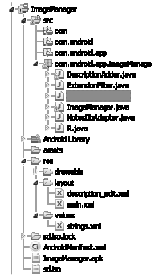
- Architecture d'un projet Android

- Java

- Code de l'utilisateur
    - Fichier R.java (ressources)

- XML

- Description de l'interface
    - Description de styles



## Activity

- Une *chose* unique et précise que peut faire l'utilisateur

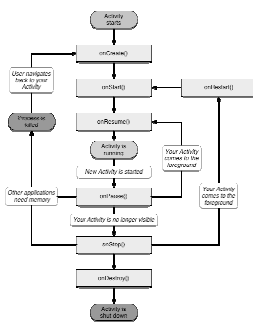
- Affichage d'un élément graphique (fenêtre, fenêtre flottante)
  - Cycle de vie

- Arborescence d'activités

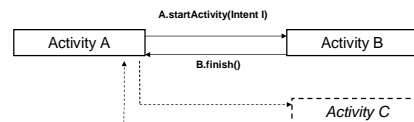
- Une seule activité à la main
  - Peut lancer une sous-activité qui prend la main

## Activity

- Cycle de vie



## Sub-Activity & Intent

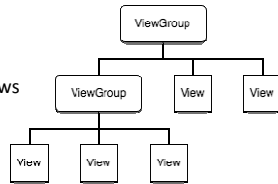


- *Intent*:

Structure contenant toutes les informations (nom, paramètres...) nécessaires au lancement d'une activité subordonnée

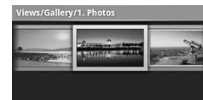
## View

- **View**
  - Structure de donnée stockant un *layout* et des composants
  - Pour une zone rectangulaire de l'écran
  - Gère le dessin, le focus, les touches ou mouvements sur la zone concernée
- **ViewGroup**
  - View contenant d'autres views
  - IG = Structure arborescente
  - ViewGroup = Layouts



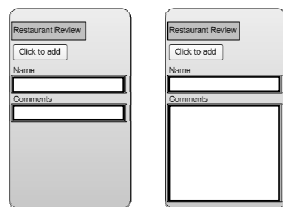
## Layouts et composants

- **Layout**
  - Définit la stratégie de placement des composants
  - Selon un axe, par des coordonnées etc.
  - Propriétés contenues dans une classe interne *LayoutParams*
- **Composants**
  - Pré-implémentés : les *Widgets*
  - Possibilité de créer ses propres composants



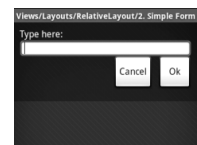
## Layouts (1/2)

- **Linear Layout**
  - Aligne tous ses fils selon une direction (H ou V)



## Layouts (2/2)

- **Absolute Layout**
  - Coordonnées par rapport au coin haut gauche
- **Relative Layout**
  - Composants identifiés par un ID
  - Placement par rapport
    - au composant parent  
*AlignParentRight(true), etc.*
    - à un autre composant :  
*below(myButton), toLeft(myTextBox), etc.*



## Question & TP

N'hésitez pas si vous avez des questions,  
on est là pour ça !

?

*A tout de suite pour le TP !*