

Programmation Nomade (Android)

Dominique Archambault

Master "Technologie et Handicap"

Université Paris 8

Cours 1 – Introduction – 2012/2013

1 Introduction

1.1 Les systèmes d'exploitation mobiles

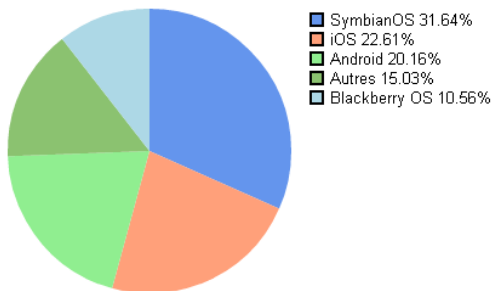
Les systèmes d'exploitation mobiles

Une concurrence féroce

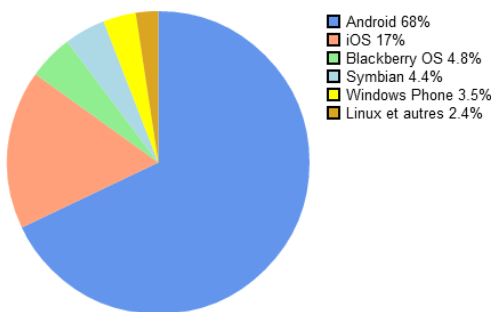
1. Android (Google Inc., **open source**)
2. Bada (Samsung Electronics)
3. BlackBerry OS (RIM)
4. HP webOS (Palm Inc.)
5. iOS (Apple Inc.)
6. MeeGo (Linux Foundation, **open source**)
7. S40 – *Series40* (Nokia)
8. Symbian OS (Nokia and Accenture)
9. Windows Phone (Microsoft)

Parts de marché

Systèmes d'exploitations des téléphones mobiles (monde) – février 2012



Ventes mondiales de smartphones selon le système d'exploitation – 2nd trimestre 2012



Les "Boutiques d'applications"

Principaux OS mobiles

OS	Boutique	Nombre d'applications	Date
IOS	App Store	700 000	(09/2012)
Android	Google Play (<i>Android Market</i>)	675 000	(09/2012)
Bada	Samsung App	2 930	(10/2012)
Blackberry OS	BlackBerry App World	10 000	(02/2012)
Symbian	Nokia Store	30 000	(05/2012)
Windows Phone	Marketplace	100 000	(06/2012)

Développement d'applications

- **IOS** : Objective C, Support d'OpenGL Plate-forme : XCode sur Mac Autre : Apple Developer Program
- **Windows phone** : C# ou VB.net Plate-forme : Silverlight ou XNA
- **Blackberry OS** : Java
- **Bada** : C++, Lib C++ STL, Support d'OpenGL

- **Android** : Java, Support d'OpenGL Plateforme : ligne de commande ou Eclipse

Développement d'Applications mobiles

Difficultés

- Taille réduite des écrans
- Taille des claviers (réels ou virtuels)
- Dispositifs de pointage peu pratiques ou imprécis
- Puissance (vitesse du processeur, stockage) limités

Écueils à éviter

- L'application utilise tellement de ressources qu'on ne peut pas recevoir d'appels
- L'application ne passe pas bien en arrière plan, ou s'arrête (et les tâches en cours sont perdues).
- L'application provoque un plantage du téléphone.

Points forts d'Android

- Langage de programmation très classique, disposant de nombreuses bibliothèques (Java)
- Outils de développement intégrés à Eclipse (plugin)
- Framework stable, en mode protégé (les applications ne peuvent pas interférer entre elles ou avec le système d'exploitation).

Développement d'Applications mobiles

Fonctionnalités disponibles

- Stockage interne et sur carte SD (fichier ou BD)
- Multimédia (musique, vidéo, photo, mémos vocaux)
- Réseau (stockage dans le *cloud*, applications Web)
- Géolocalisation (via puce GPS ou via réseau)
- Différents capteurs : Accéléromètre, Gyroscope, Boussole, Capteur de proximité, Baromètres...
- Écrans multitouch
- Lecteur de RFID
- Services téléphoniques (appels voix/vidéo, SMS)

Le système d'exploitation Android

Android

C'est un système d'exploitation pour terminaux mobiles (smart-phones, tablettes, PDA, etc.), basé sur le noyau Linux.

- Licence open source (Apache v2)
- Développé par Google Inc.

Conçu pour intégrer les application Google

- Gmail
- Google Maps
- Google Agenda
- Google Talk
- YouTube
- Google Latitude

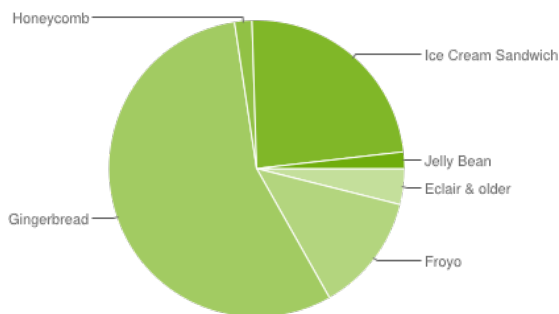
Le système d'exploitation Android

Versions

Version		Sortie	API level	
1.0	<i>Apple Pie</i> (α) Tarte aux pommes	fin 2007	1	
1.1	<i>Banana Bread</i> (β) Cake à la banane	fin 2008	2	
1.5	<i>Cupcake</i> Petit gâteau	30/04/2009	3	0,1 %
1.6	<i>Donut</i> Beignet	15/09/2009	4	0,4 %
2.0 - 2.1	<i>Eclair</i> Éclair	26/10/2009	5-6-7	3,4 %
2.2.x	<i>Froyo</i> (<i>Frozen Yogourt</i>) Yaourt glacé	20/05/2010	8	12,9 %
2.3 - 2.3.2	<i>Gingerbread</i> Pain d'épices	06/12/2010	9	0,3 %
2.3.3-7			10	55,5 %
3.0	<i>Honeycomb</i> Rayon de miel	22/02/2011	11	
3.1.x			12	0,4 %
3.2.x			13	1,5 %
4.0.x	<i>Ice Cream Sandwich</i> Sandwich à la glace	19/10/2011	14-15	23,7 %
4.1.x	<i>Jelly Bean</i> Dragibus	09/07/2012	16	1,8 %
4.2	<i>Key Lime Pie</i> Tarte au citron	?		

Le système d'exploitation Android

Versions



<http://developer.android.com> [About > Dashboards]

Le système d'exploitation Android

Problèmes

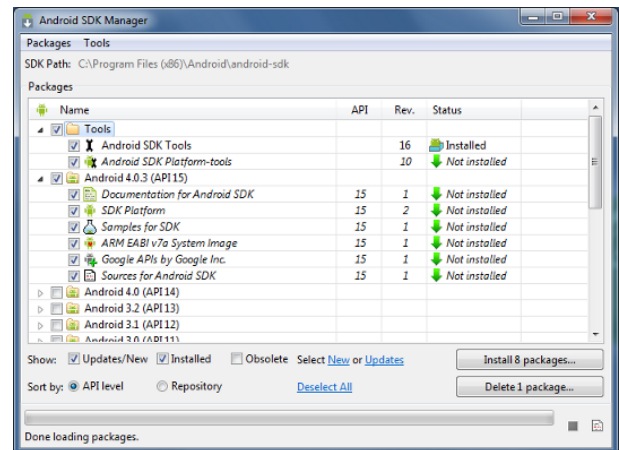
- Nombreuses versions
- ROMs modifiés par les constructeurs
- ROMs modifiés par les opérateurs
- ROMs alternatives

1.2 Développer des applications Android

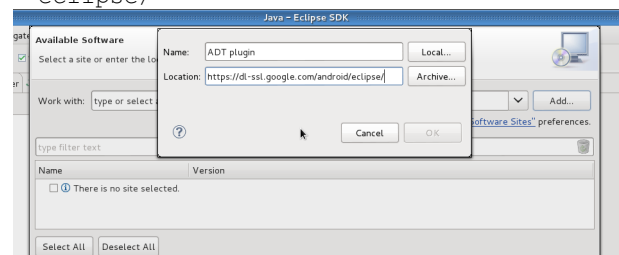
Développement d'application Android

Outils de développement

- le **SDK** à télécharger sur <http://developer.android.com> [Develop > Tools > Download]
- les **plateformes et paquets** Dans le répertoire du SDK, lancer le **Android SDK Manager**. Les paquets recommandés sont sélectionnés automatiquement, choisir ceux qu'on veut installer et lancer l'installation.



- le **plugin Eclipse** Dans Eclipse, menu [Help > Install New Software]
Ajouter le **Repository** : **ADT Plugin** avec l'URL suivante : <https://dl-ssl.google.com/android/eclipse/>



Développement d'application Android

Langages de développement

- **XML** pour la définition des écrans Un outil graphique est disponible
- **JAVA** pour l'implémentation

Développement d'application Android

Contenu du programme Android

- **Activités** (*Activities*)
- **Services** (*Services*)
- **Fournisseurs de contenu** (*Content providers*)
- **Intentions** (*Intents*)

Annexe 1 : Application "Hello World !"

Code java : HelloWorldActivity.java

```

package net.chezdom.android.cours_000;

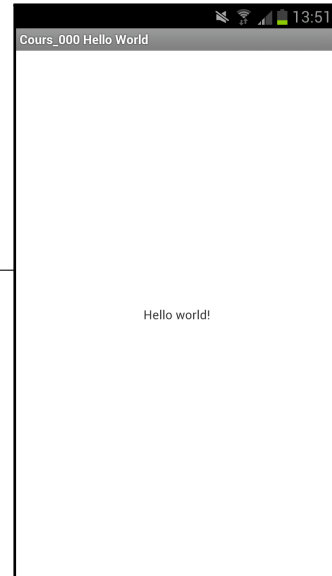
import android.os.Bundle;

public class HelloWorldActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hello_world);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_hello_world, menu);
        return true;
    }
}

```



Code XML (écran) : HelloWorldActivity.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:padding="@dimen/padding_medium"
        android:text="@string/hello_world"
        tools:context=".HelloWorlsActivity" />

</RelativeLayout>

```

Code XML (chaînes) : strings.xml

```

<resources>

    <string name="app_name">Cours_000 Hello World</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_hello_world">Cours_000 Hello World</string>

</resources>

```

Assistant “New Android Project”

New Android Application

⊗ Enter an application name (shown in launcher)

Application Name:

Project Name:

Package Name:

Build SDK: Choose...

Minimum Required SDK:

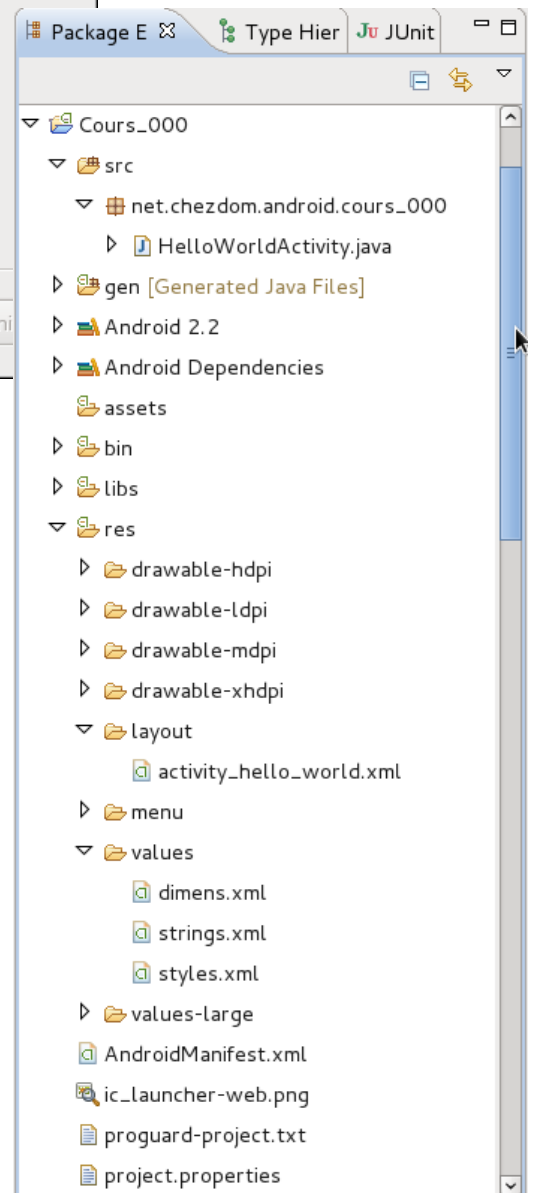
Create custom launcher icon

Mark this project as a library

Create Project in Workspace

Location: Browse...

Projet Android



2 Interfaces utilisateurs

Créer une interface utilisateur

Étapes

- Créer un "layout" en utilisant l'assistant graphique
- Créer les objets correspondants aux widgets qu'on souhaite manipuler dans le programme JAVA
- Associer ces objets aux éléments du "layout" par l'intermédiaire de leurs IDs
- Faire implémenter les interfaces de manipulation de ces objets à la classe *Activity* de l'application (ou créer des classes "contrôleurs" dédiées).
- Associer les objets correspondants aux widgets à la classe implémentant leurs interfaces de manipulation (this, si on a utilisé la classe *Activity*).
- Implémenter les méthodes de manipulation.

Quelques éléments d'interface

Views (widgets)

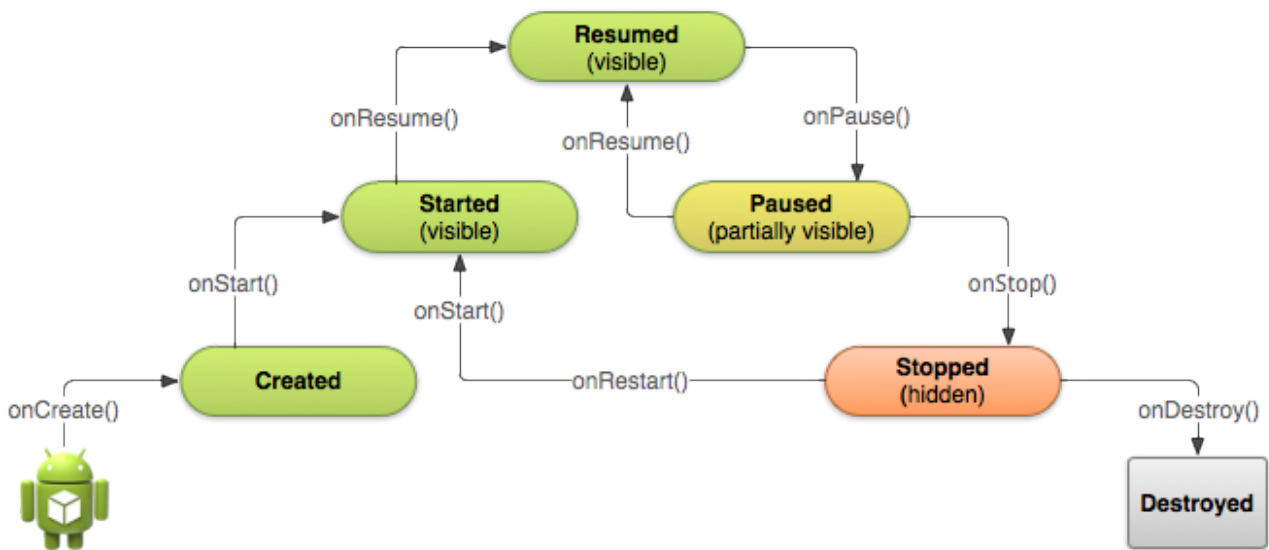
- Button
- CheckBox
- EditText
- RadioButton
- Toast

- DatePicker
- ImageButton
- SlidingDrawer

Layouts (conteneurs)

- LinearLayout
- RelativeLayout
- GridLayout
- FrameLayout

Cycle de vie d'une activité

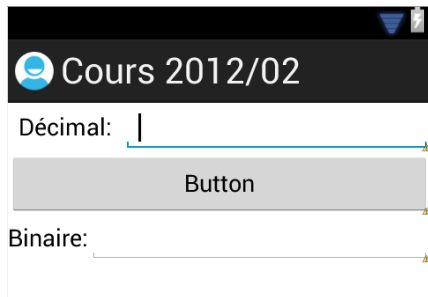


View.OnClickListener	onClick(View v)
View.OnLongClickListener	onLongClick(View v)
View.OnFocusChangeListener	onFocusChange(View v, boolean hasFocus)
View.OnKeyListener	onKey(View v, int keyCode, KeyEvent event)
View.OnTouchListener	onTouch(View v, MotionEvent event)

TP 1

Nous allons créer une première application, dont le but est de convertir des nombres entiers en binaire.

1. Créer un layout avec l'assistant graphique



Indication : Le layout ci-dessus est formé d'un **LinearLayout** vertical, contenant :

- un **LinearLayout** horizontal, contenant lui-même :
 - une étiquette (**TextView**)
 - et un champ de saisie **EditText**.
- un **Button**,
- et un second **LinearLayout** horizontal, contenant aussi une étiquette et un champ de saisie.

On pourra aussi (dans l'éditeur graphique), utiliser un champ de saisie qui n'accepte que les nombres.

2. Créer les objets correspondants et les associer aux éléments du layout

```
import android.widget.Button;

public class MainActivity extends Activity {

    private Button bGO;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bGO = (Button) this.findViewById(R.id.button1);
    }
}
```

NB : Attention à bien importer la classe **Button** du package android.widget (voir la première ligne du code ci-dessus). On trouvera l'ID du bouton dans le fichier XML "layout".

```
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button" />
```

3. La classe Activity implémente l'interface de manipulation

```
import android.view.View.OnClickListener;

public class MainActivity extends Activity
    implements OnClickListener {
    //...

    @Override
    public void onClick(View v) {
        // ...
    }
}
```

NB : Attention à bien importer l'interface **OnClickListener** du package android.widget.View.

4. Associer les objets widgets à leur interface de manipulation

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    bGO = (Button) this.findViewById(R.id.button1);
    bGO.setOnClickListener(this);
}
```

5. Implémenter les méthodes de manipulation

```
import android.widget.Toast;

//...

@Override
public void onClick(View v) {
    if (v.getId()==R.id.button1)
        Toast.makeText(getApplicationContext(),
            "Conversion!",
            Toast.LENGTH_SHORT).show();
}
```

6. Récupérer une valeur dans un champ de texte

On associe l'objet EditText avec son champ de saisie, en utilisant l'ID.

```
import android.widget.EditText;

public class MainActivity extends Activity
    implements OnClickListener {
    //...
    private EditText tfDecimal, tfBinaire;

    public void onCreate(Bundle savedInstanceState) {
        //..
        tfDecimal=(EditText) this.findViewById(
            R.id.editText1);
    }
}
```

```
String chDecimal=tfDecimal.getText().toString();
int d=Integer.parseInt(chDecimal);
```

7. Écrire dans un champ de texte

Ne pas oublier d'associer l'objet EditText avec son champ de saisie, en utilisant l'ID comme précédemment.

```
tfBinaire.setText(""+convertInBinary(d));
```

Rappel : Conversion en binaire

(voir TD intensifs)

```
1 bin=""
2 tantque n>0
3 faire
4     si n%2=0
5     alors
6         bin = "0" + bin
7     sinon
8         bin = "1" + bin
9     finis
10    n=n/2
11 fait
```