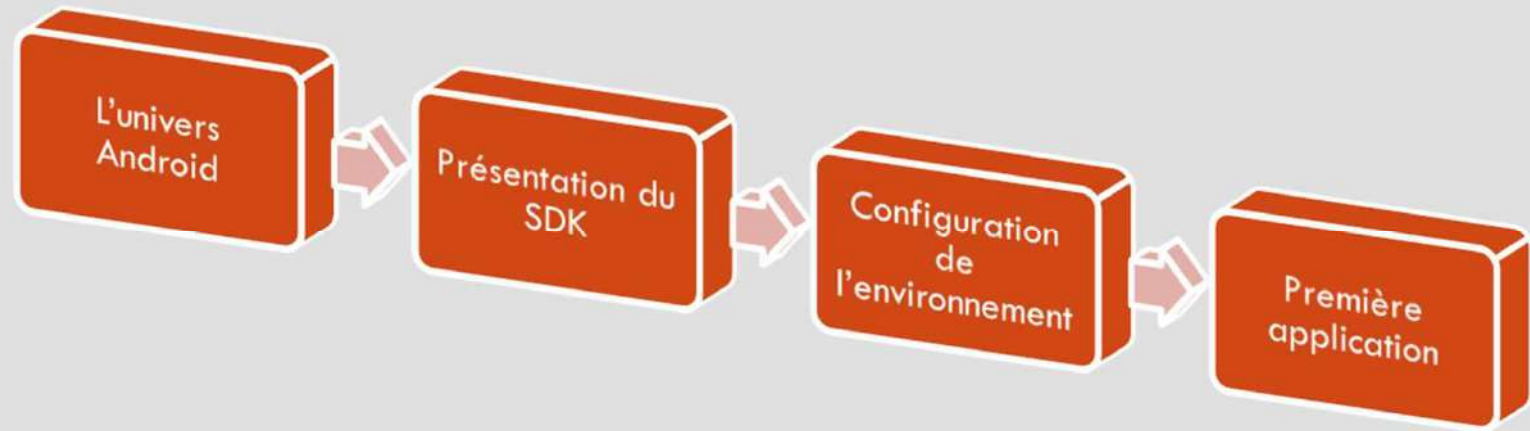




PROGRAMMATION ANDROID

Ana ROXIN

Partie 01 – Introduction à Android



2

Plan du cours

Que vous débutiez en développement ou que vous soyez déjà un développeur Java aguerri, Android est une excellente opportunité pour appréhender le développement d'applications mobiles ambitieuses. Ce premier cours vous permettra de vous familiariser avec la plate-forme Android et l'environnement de développement Eclipse.

Fichiers source

3

- Sur ufrsciencestech.u-bourgogne.fr
- Répertoire UE5/AR
 - Cours
 - TD: énoncé + fichiers source

Installation

4

- Connexion sous Linux
 - ADT Bundle pour Linux 64-bit - [adt-bundle-linux-x86_64.zip](#)
 - A télécharger, puis à extraire sur le bureau

 - Ouvrir Eclipse
 - Dossier_ADT/eclipse/eclipse.exe

 - Lancer le Android SDK Manager
 - Vérifier les mises à jour du SDK
 - Versions obligatoires: 1.5, 1.6, 2.1, 4.0 à 4.1

**1^e
partie**

La plate-forme Android

Points clés de la plate-forme Android

6

Innovante

- Intègre toutes les dernières technologies de téléphonie mobile : écran tactile, accéléromètre, GPS, appareil photo numérique etc.

Accessible

- Pas d'achat de matériel spécifique;
- Pas de connaissance d'un langage peu utilisé ou spécifique : Java;

Ouverte

- Fournie sous licence open source
 - Permet de consulter les sources et d'effectuer des modifications.

La plate-forme Android

7

- Android = système d'exploitation pour mobiles open source développé par la société Google
- Juin 2007 – création de OHA (Open Handset Alliance)
 - ▣ Fabricants de téléphones: HTC, Sony Ericsson, Samsung ou Motorola
 - ▣ Opérateurs de téléphonie: Sprint, T-Mobile ou NTT DoCoMo
 - ▣ Société Internet: Google, eBay
 - ▣ Constructeurs de puces électroniques: Intel, nVidia
 - ▣ Autres acteurs: Garmin
- Open Handset Alliance et Android sont des marques déposées.

Plus de renseignements

- ☑ site de l'OHA: <http://www.openhandsetalliance.com/>

Les membres de l'OHA

8



Quelques offres des constructeurs

9

~~Android = Google Phone~~



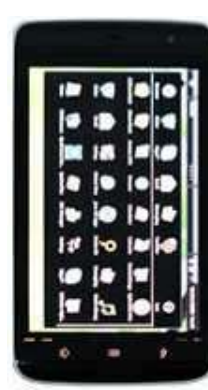
HTC
Desire HD



Huawei
U8230



Google
Nexus 7



Acer
Stream



Motorola
Razr HD



Sony
Xperia X10

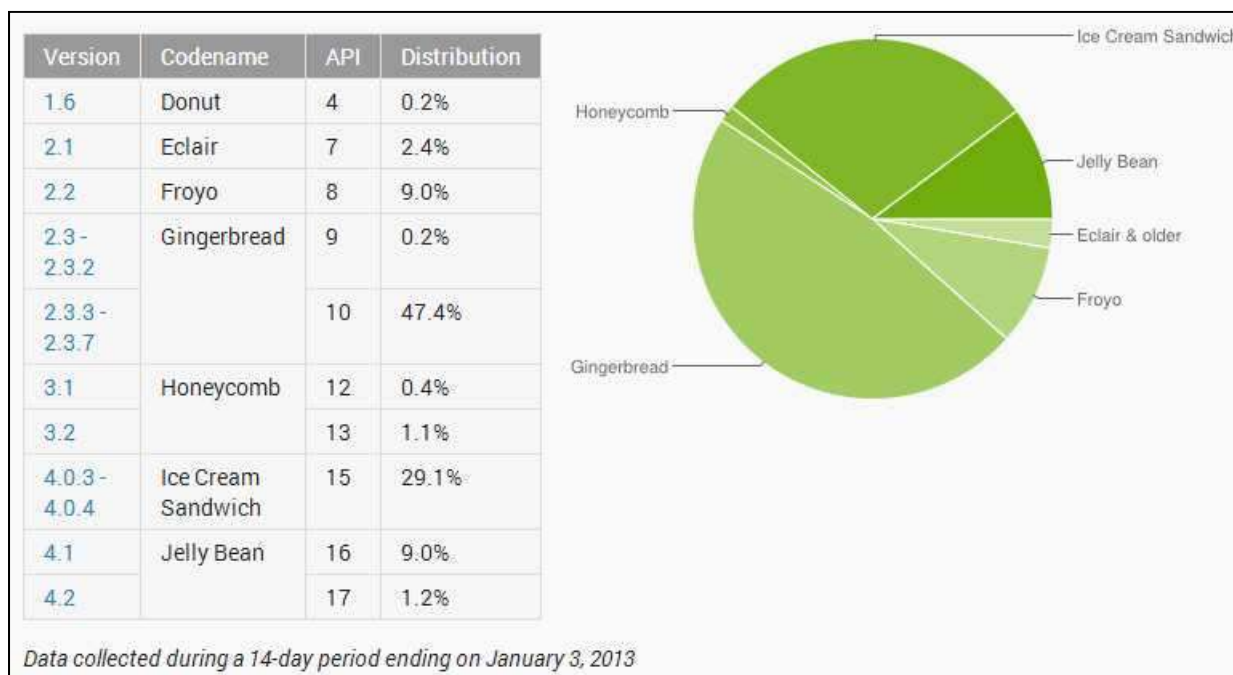
Les versions de la plate-forme (1)

10

Version	Date	Nom	Remarques
Android 1.0	09/2008		Android Market, sites Web en (X)HTML, appareil photo, dossiers d'applications, synchro Gmail, etc.
Android 1.1	02/2009	Petit Four	Mise en ligne pour le HTC Dream
Android 1.5	04/2009	Cupcake	Basée sur le noyau Linux 2.6.27, support pour les claviers virtuels, les widgets, copier/coller, etc.
Android 1.6	09/2009	Donut	Basée sur le noyau 2.6.29, mäj de la recherche et de la recherche vocale, interface Android Market améliorée
Android 2.0	10/2009	Eclair	Vitesse hardware optimisée; support de plus de taille d'écran et résolutions, réorganisation de l'UI, nouvelle interface du navigateur et support de l'HTML5
Android 2.2	05/2010	FroYo	Basée sur le noyau 2.6.32, optimisations générales de la vitesse, de la mémoire et des performances de l'OS
Android 2.3	12/2010	Gingerbread	Basée sur le noyau 2.6.35, mäj interface utilisateur, support de nouveaux capteurs (gyroscope et baromètre), support du développement en code natif amélioré, etc.
Android 3.0	02/2011	Honeycomb	Basée sur le noyau 2.6.36 et réservée aux tablettes tactiles
Android 4.0.1	10/2011	Ice Cream Sandwich	Basée sur le noyau 3.0.1 Boutons virtuels intégrés à l'interface, création facilitée des dossiers, amélioration du copier/coller, etc.
Android 4.1.1	07/2012	Jelly Bean	Basée sur le noyau Linux 3.0.31 Amélioration des fonctionnalités (recherche vocale, appareil photo, usb audio, Google Chrome, etc.) Interface utilisateur plus fluide (synchronisation Vsync, triple buffering)
Android 4.2.1	11/2012		Amélioration de la stabilité et des performances Correction du bug du mois de décembre 2012 Bluetooth: Résolution de problèmes, ajout de GamePads et Joystick Bluetooth
Android 5.0		Key Lime Pie	

Les versions de la plate-forme (2)

11

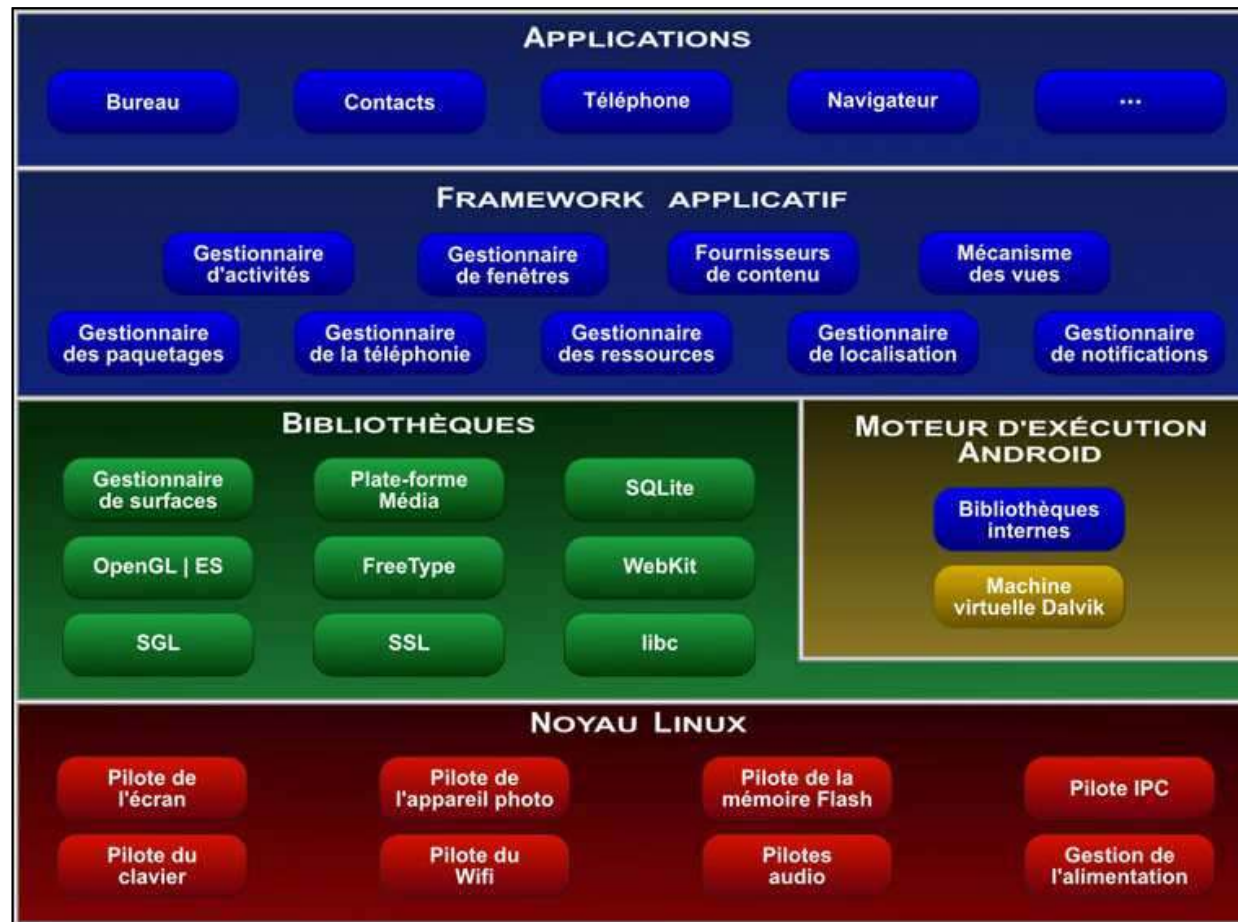


Plus de renseignements

- ✓ <http://developer.android.com/about/dashboards/index.html>
- ✓ http://fr.wikipedia.org/wiki/Historique_des_versions_d'Android

L'architecture

12



La licence

13

- La licence principale du projet Android est la licence Apache 2.0.
 - Autorise la modification et la distribution du code sous forme libre ou non;
 - Permet, comme toute licence vraiment libre, d'en faire un usage commercial.
 - Oblige le maintien du copyright lors de toute modification (et également du texte de la licence elle-même).

- D'autres parties du projet peuvent être sujettes à d'autres types de licences
 - Parties logicielles proches du noyau Linux qui sont concernées par la licence GPL

Plus de renseignements

☑ <http://www.gnu.org/licenses/license-list.html>

Pratique: découverte des tablettes (1)

14

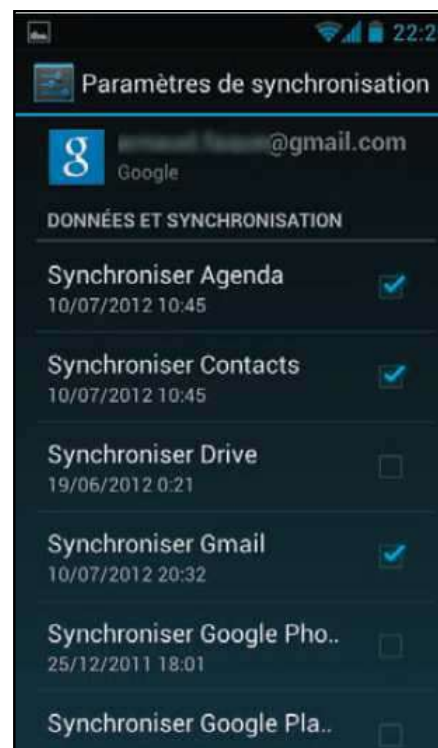
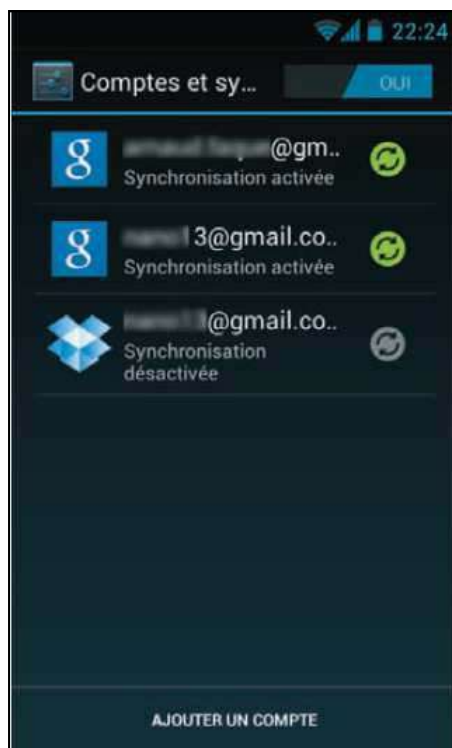
- ❑ Connexion au réseau WiFi: IEMInfo
- ❑ Avec vos login et mdp pour les salles de TP
- ❑ Pour connaître les caractéristiques de la tablette **Android Info System**



Pratique: découverte des tablettes (2)

15

- Identification avec un compte google
 - Gestion du compte Google: Paramètres système > Comptes et synchro



Pratique: découverte des tablettes (3)

16

- Savoir qui est connecté à votre compte Gmail

Activité sur ce compte

Cette fonctionnalité fournit des informations concernant la dernière activité sur ce compte de messagerie et toute autre activité simultanée. [En savoir plus](#)

Il ne semble pas que ce compte soit ouvert ailleurs. Néanmoins, il se peut que certaines sessions n'aient pas été fermées.

Activité récente :

Type d'accès [?] (Navigateur, mobile, POP3, etc.)	Lieu (adresse IP) [?]	Date/heure (Affiché dans votre fuseau horaire)
Navigateur	* France (82.229. [redacted])	22:13 (il y a 0 minutes)
Mobile	France (82.229. [redacted])	20:32 (il y a 1,5 heures)
Mobile	France (92.90. [redacted])	13:03 (il y a 9 heures)
Mobile	France (82.229. [redacted])	2 juin (il y a 1 jour)
Mobile	France (82.229. [redacted])	2 juin (il y a 1 jour)
Mobile	France (92.90. [redacted])	2 juin (il y a 1 jour)
Mobile	France (82.229. [redacted])	1 juin (il y a 2 jours)
Mobile	France (92.90. [redacted])	1 juin (il y a 2 jours)
Navigateur	France (82.235. [redacted])	1 juin (il y a 2 jours)
Mobile	France (82.235. [redacted])	1 juin (il y a 2 jours)

Dashboard Système Tasks Apps Logs

Max: 1007.89Mo Libre: 346.87Mo

Carte SD
42%
Max: 13624.34Mo Libre: 5855.51Mo

RAM
12%
Max: 334Mo Libre: 41Mo

Uptime
0 jours, 16 heures, 7 minutes, 7 secondes

Type de réseau
mobile

adresse IP
10.74.134.182

Pratique: découverte des tablettes (4)

17

- Deux solutions s'offrent à nous:
- Si l'appareil est neuf ou si l'on vient d'installer une nouvelle ROM (d'origine ou alternative), il suffit de ne pas saisir l'identifiant Gmail, et de ne pas le créer, lorsque celui-ci vous est demandé.
- Si le compte est déjà actif sur votre smartphone, il s'agit de le supprimer. Il faut également le supprimer de toutes les applications utilisant ce compte.
 - Rendez-vous dans **Paramètres > Applications > Gérer les applications**. Sélectionnez **Gmail**.
 - Là, cliquez sur **Effacer les données**. Il faudra répéter ces opérations pour les autres applications recourant au fameux compte : Gtalk, Google Documents, Google Maps, Contacts, Galerie, etc.
- À ce stade, un message « Warning » devrait apparaître dans la zone de notification.
 - Pour le faire disparaître, il est nécessaire d'enregistrer un compte Google valide !

Pratique: découverte des tablettes (5)

18



Pratique: découverte des tablettes (6)

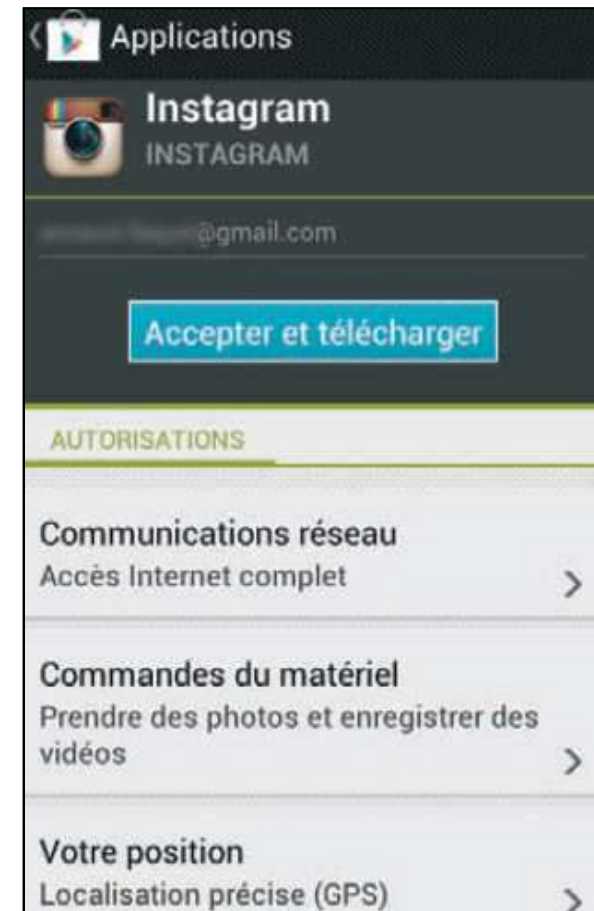
19



Pratique: découverte des tablettes (7)

20

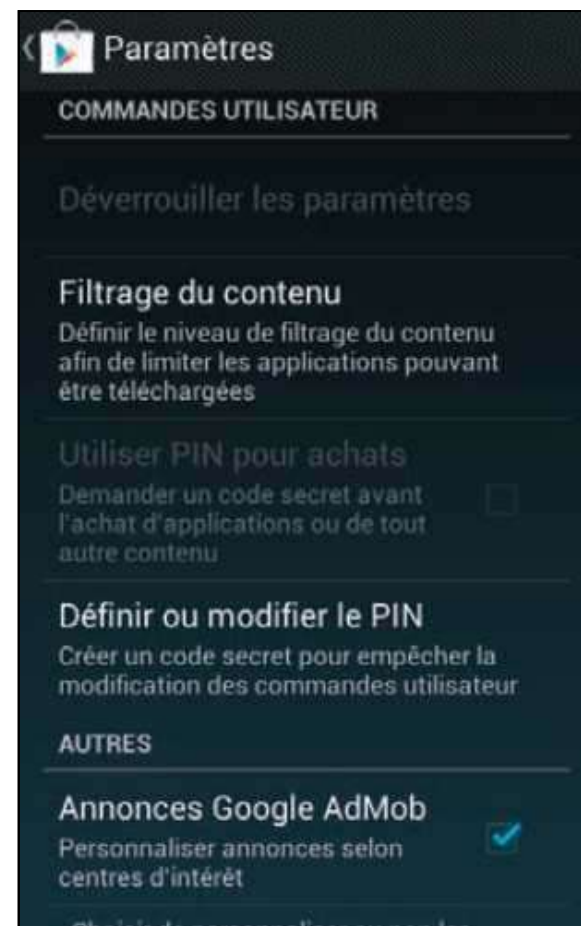
- **CyanogenMod : ROM alternative**
 - L'utilisation de ROM alternatives, comme CyanogenMod, permet de gérer les droits pour chaque application.
 - CyanogenMod agit en quelque sorte comme un pare-feu (*firewall*). Même si une application veut un droit, on peut l'empêcher de l'avoir ; elle restera tout de même opérationnelle.



Pratique: découverte des tablettes (8)

21

- L'application Google Play installée sur le smartphone permet dans ses options d'appliquer un filtrage sur le contenu (**Paramètres > Filtrage des contenus**).
 - Aucun lien avec la sécurité,
 - Permet de créer un filtre des applications qui seront proposées au téléchargement en fonction des critères définis
- Ce paramètre est à vérifier si certaines applications refusent de s'afficher dans le module de recherche.



Pratique: découverte des tablettes (9)

22

- <http://fr.appbrain.com>
 - Cette application peut avantageusement remplacer Google Play.
 - Elle se décompose en deux parties :
 - un site Internet
 - une application sur l'appareil.
- <http://yaam.mobi>
- <http://f-droid.org>

Welcome to the AppBrain Android market

- ✓ Découvrez les meilleures applications pour Android via la recherche, le classement et les catégories.
- ✓ Easily install and manage your apps directly from the web browser and on the phone with our free Android app.
- ✓ Partager les applications installées sur votre téléphone avec vos amis.

Join AppBrain today

Featured by AppBrain

Dernières Evaluations

AppBrain App Market
Swiss Codemoneys
90 Free 1,000,000+

AppBrain Ad Detector
Swiss Codemoneys
90 Free 100,000+

Face Changer
Sibompa
87 Free 1,000,000+

Make Me Bald - Video
Swiss Codemoneys
79 Free 100,000+

SPES Backup & Restore
Good appl. :)
par christophe

YouTube Music
par lafayette53

Age of Empires
Very good game, nice development, many events, news, and functionalities... You can have a very good start bonus if you ...
par DAMIER

QuickMark Barcode Scanner
works perfect product it easy and has alot of current info
par thewhistler

More >

Pratique: découverte des applications

23

- ❑ L'antivirus avast! Free Mobile Security

- ❑ Divers
 - ❑ Timeriffic, pour minuter les périodes de veille ou mode Avion
 - ❑ eBuddy Messenger pour tchatter
 - ❑ Evernote : un bloc-notes sur le cloud

- ❑ Pour l'administrateur système et réseau
 - ❑ Hacker's Keyboard: clavier avec les touches Ctrl et Alt
 - ❑ ConnectBot : se connecter à distance via SSH
 - ❑ TeamViewer : bureau à distance
 - ❑ Fing - Network Tools : utilitaire réseau

**2^e
partie**

Le kit de développement Android

Le SDK

25

- Google fournit, en plus du système d'exploitation, un kit de développement comprenant:
 - des API (interfaces de programmation) ;
 - des exemples de code ;
 - de la documentation ;
 - des outils – parmi lesquels un émulateur – permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application.

- Le SDK Android est disponible gratuitement sur le site de Google.

Plus de renseignements

✓ <http://developer.android.com>

La documentation

26

- La documentation du SDK Android est scindée en deux parties bien distinctes :
 - Le guide du développeur:
 - Disponible en HTML
 - Dans le répertoire `/docs/guide/index.html`.
 - La documentation des API
 - Au format javadoc – générée automatiquement à partir des commentaires du code source formatés avec une syntaxe précise
 - Dans le répertoire `/docs/reference/packages.html`.

Les exemples d'applications

27

- Le kit de développement est accompagné de quelques exemples illustrant les possibilités du SDK Android.
 - dans le répertoire `platforms\android-2.0\samples`
- Parmi ces exemples, on peut citer:
 - un jeu du serpent (répertoire Snake) ;
 - un projet qui couvre l'utilisation de plusieurs pans de l'API Android comme les alarmes, les notifications, les menus, etc. (répertoire APIDemos) ;
 - une application pour sauvegarder des notes (répertoire NotePad).

Les outils de développement du SDK

28

- Le SDK est livré avec un certain nombre d'outils couvrant différents aspects du cycle de développement d'une application Android.
 - Dans le répertoire tools du SDK
- Parmi ces programmes, on peut citer:
 - DDMS – outil de débogage puissant ;
 - mksdcard – outil de création de cartes mémoire logicielles utilisables avec l'émulateur ;
 - sqlite3 – outil pour accéder aux fichiers de données au format SQLite.
 - ADT (Android Development Tools Plugin) – outil intégrable à Eclipse, proposant des interfaces et des assistants pour la création et le débogage d'applications.

**3^e
partie**

Configurer l'environnement de développement

Quelques liens utiles...

30

- Récupérer les sources d'Android
 - <http://source.android.com/>
- Sites d'informations sur Android:
 - Site officiel d'Android:
 - <http://www.android.com/>
 - Site pour les développeurs Android:
 - <http://developer.android.com/index.html>
 - Sites pour suivre l'actualité Android en français:
 - <http://www.pointgphone.com/>
 - <http://android-france.fr/>
 - Site pour rechercher des applications Android:
 - <http://www.androlib.com/>

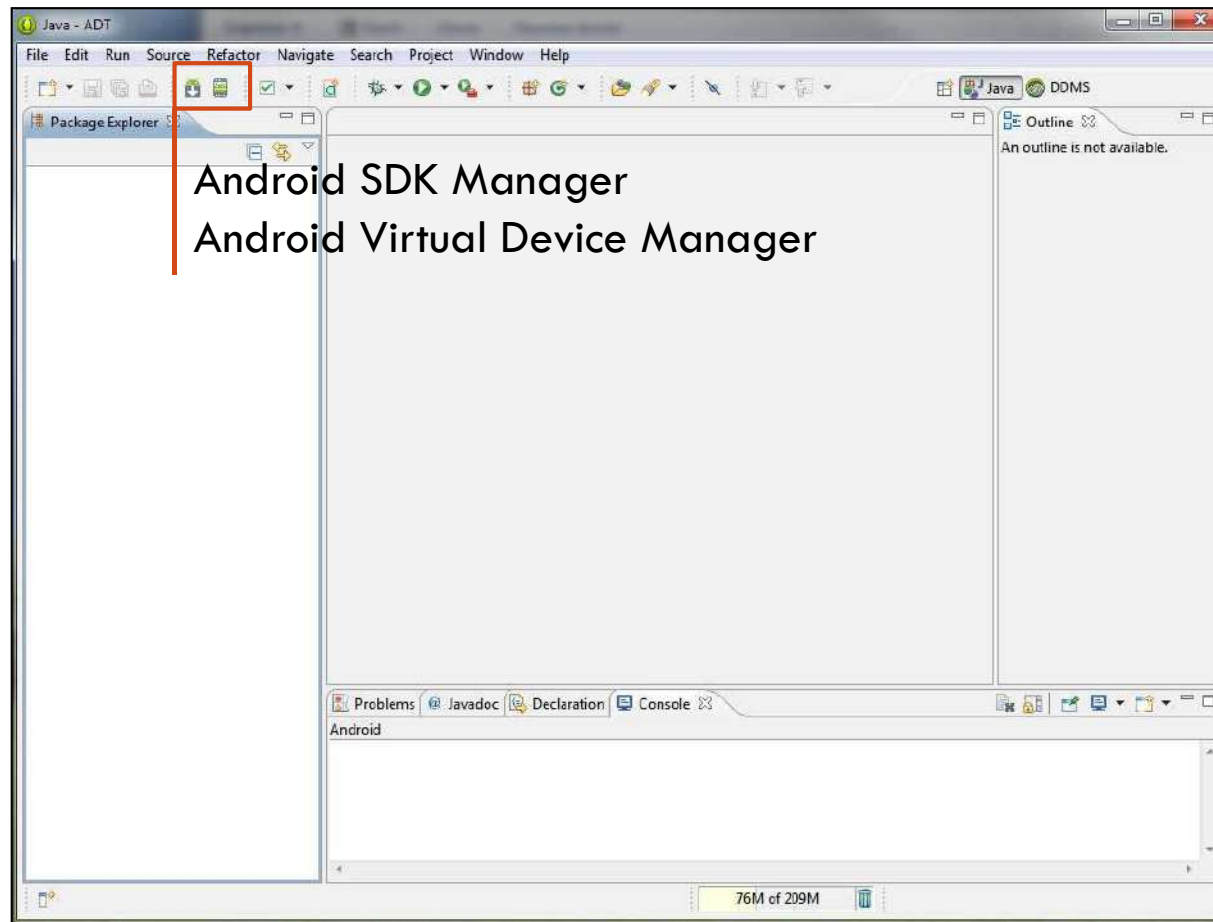
Les outils indispensables

31

- Eclipse
 - <http://www.eclipse.org/>
 - Eclipse IDE for Java EE Developers (la plus complète)
- SDK Android + plugin ADT Eclipse
 - <http://developer.android.com/sdk/index.html>
- Java Runtime Environment (JRE)
 - <http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

Ecran d'accueil d'Eclipse Java

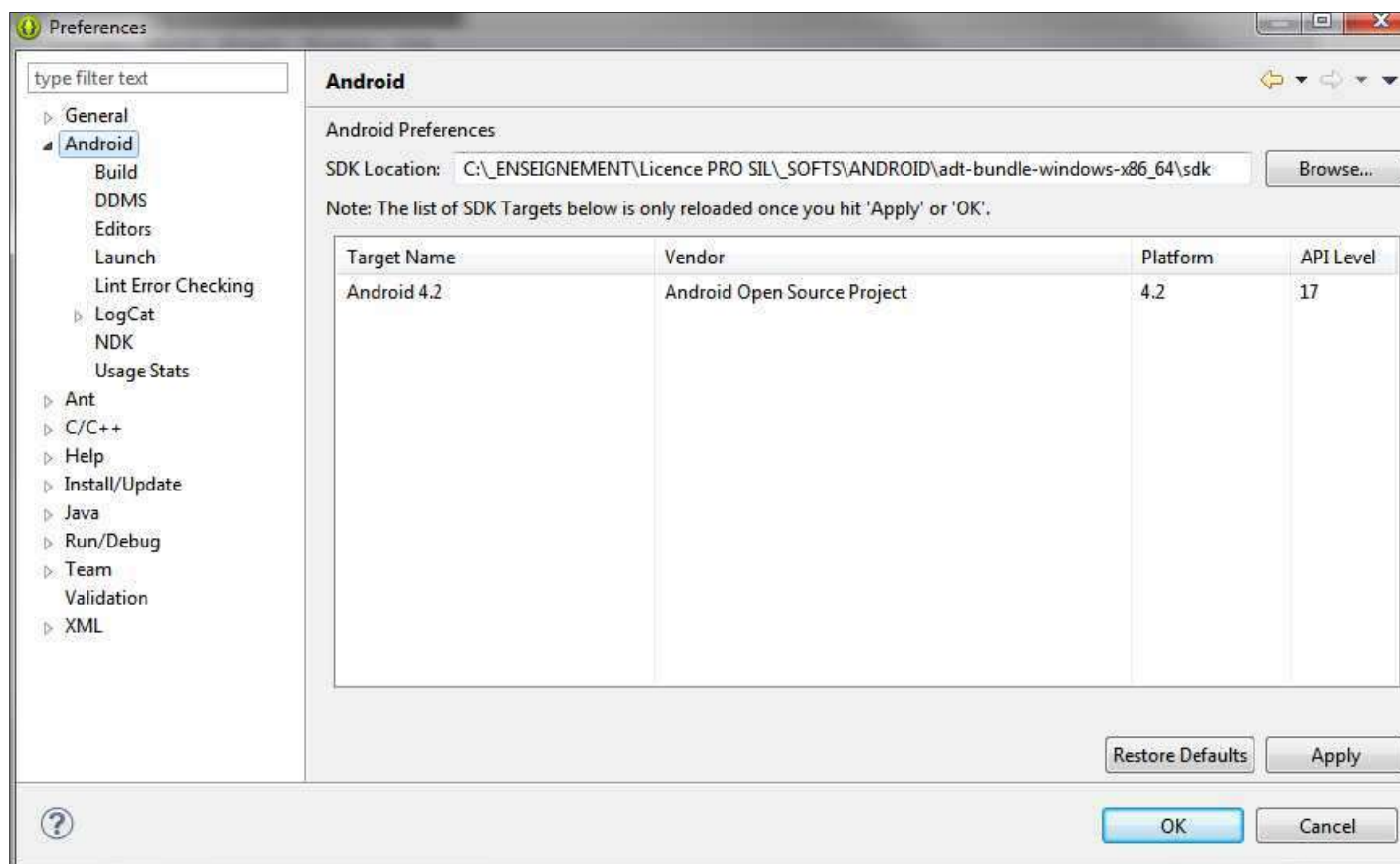
32



Lien avec le SDK Android

33

- Menu Window > Lien Preferences

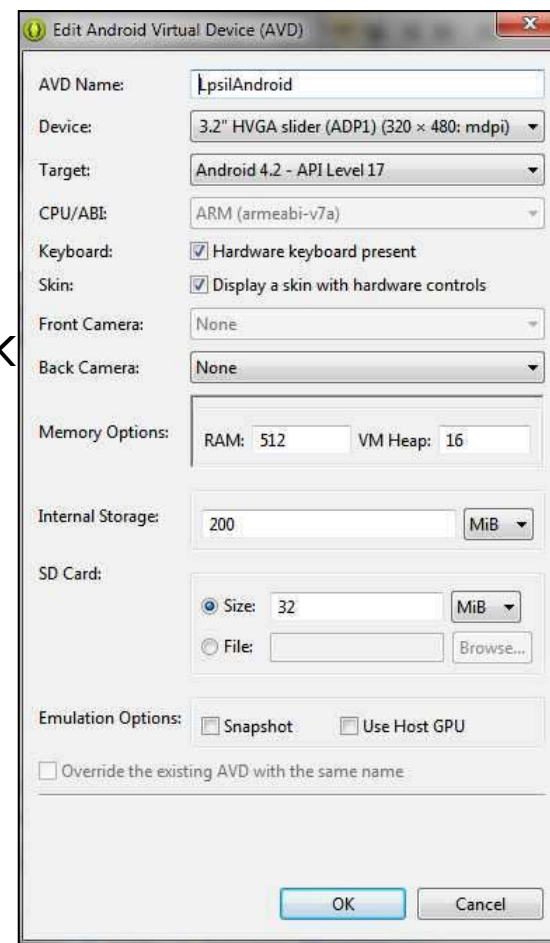


Créer un émulateur

34

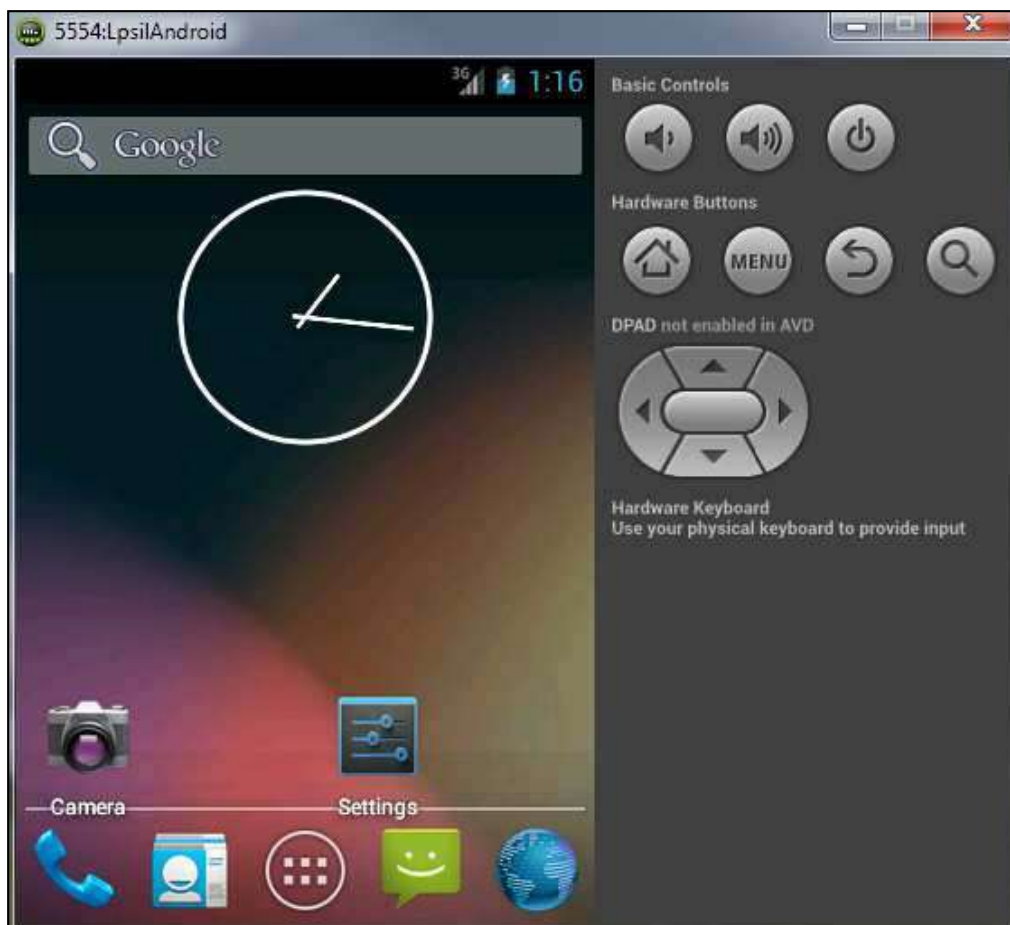
- ❑ Ouvrir l'AVD (Android Virtual Device Manager)
- ❑ Cliquer sur **New**, puis remplir les champs:
 - ❑ AVD Name: LpsilAndroid
 - ❑ Device 3,2" HVGA slider
 - ❑ Target: si possible 1.5, sinon dernière version du SDK
- ❑ Cliquer sur **Start...**

- ❑ Lancement de l'émulateur:
 - ❑ Invite de commandes
 - Aller dans le dossier d'installation du SDK, puis dans le répertoire `tools`
 - `emulator.exe -avd LpsilAndroid`



Découverte de l'émulateur

35



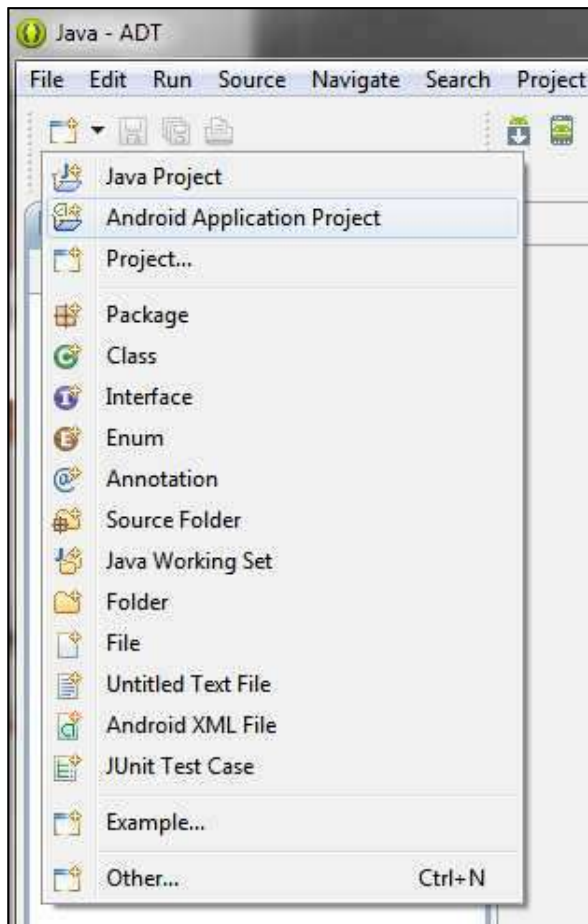
Touche	Description
Touche 9 pavé numérique	Passer en mode paysage / repasser en mode portrait.
Touche +	Augmente le volume
Touche -	Diminue le volume
F8	Activer/désactiver le réseau cellulaire.
Alt + Entrée	Activer/désactiver le mode plein écran.
F6	Activer/désactiver le trackball.

**4^e
partie**

Une première application

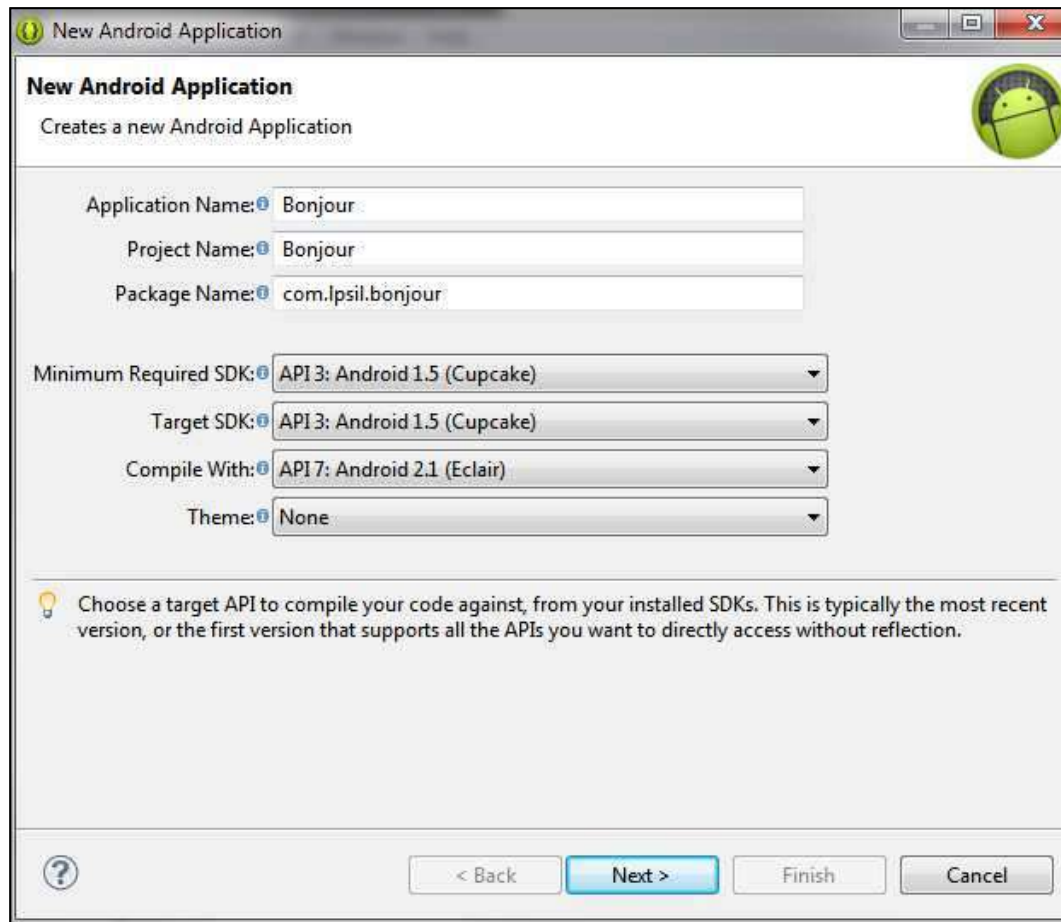
Créer un nouveau projet Android (1)

37



Créer un nouveau projet Android (2)

38



New Android Application
Creates a new Android Application

Application Name:

Project Name:


Package Name:

Minimum Required SDK:

Target SDK:

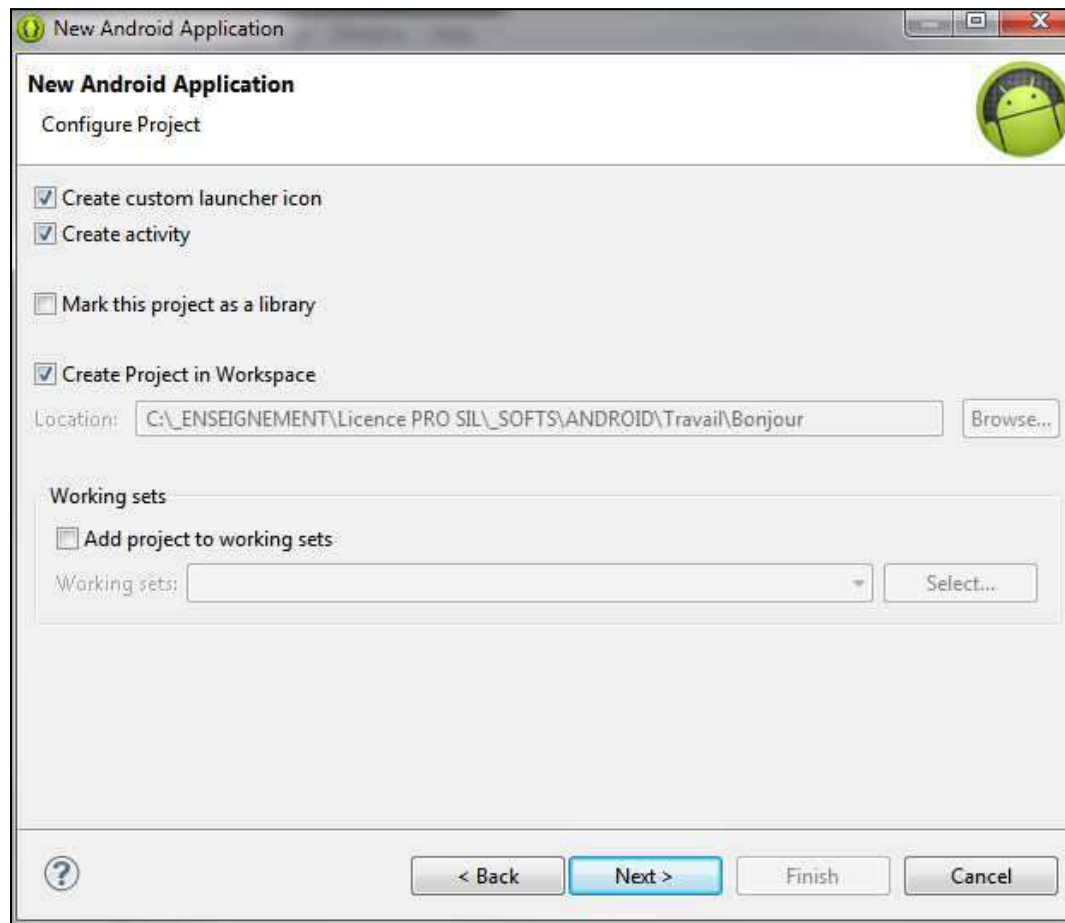
Compile With:

Theme:

 Choose a target API to compile your code against, from your installed SDKs. This is typically the most recent version, or the first version that supports all the APIs you want to directly access without reflection.

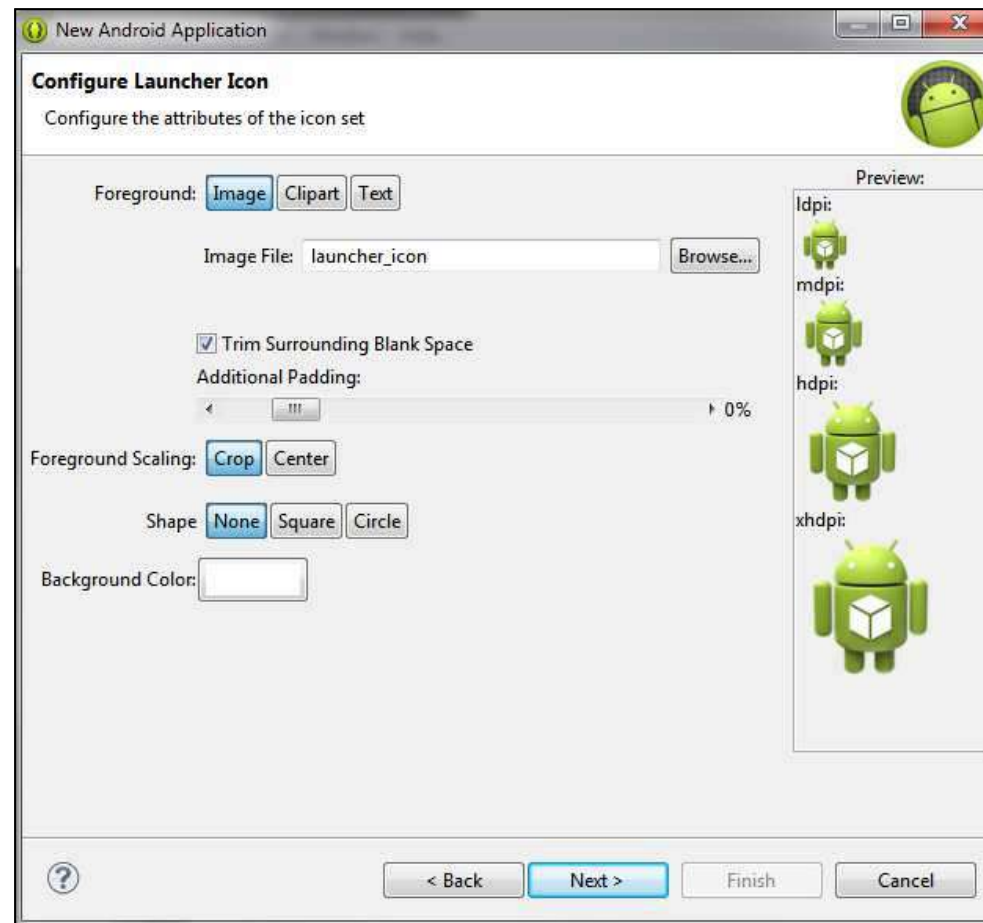
Créer un nouveau projet Android (3)

39



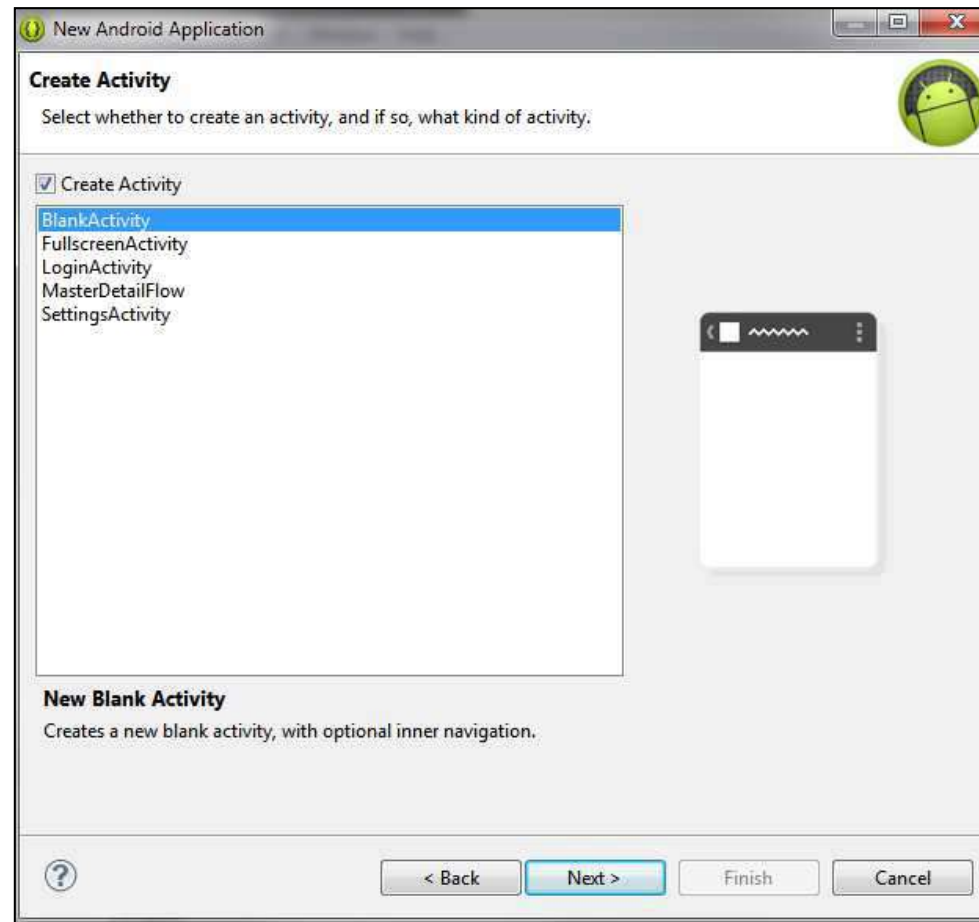
Créer un nouveau projet Android (4)

40



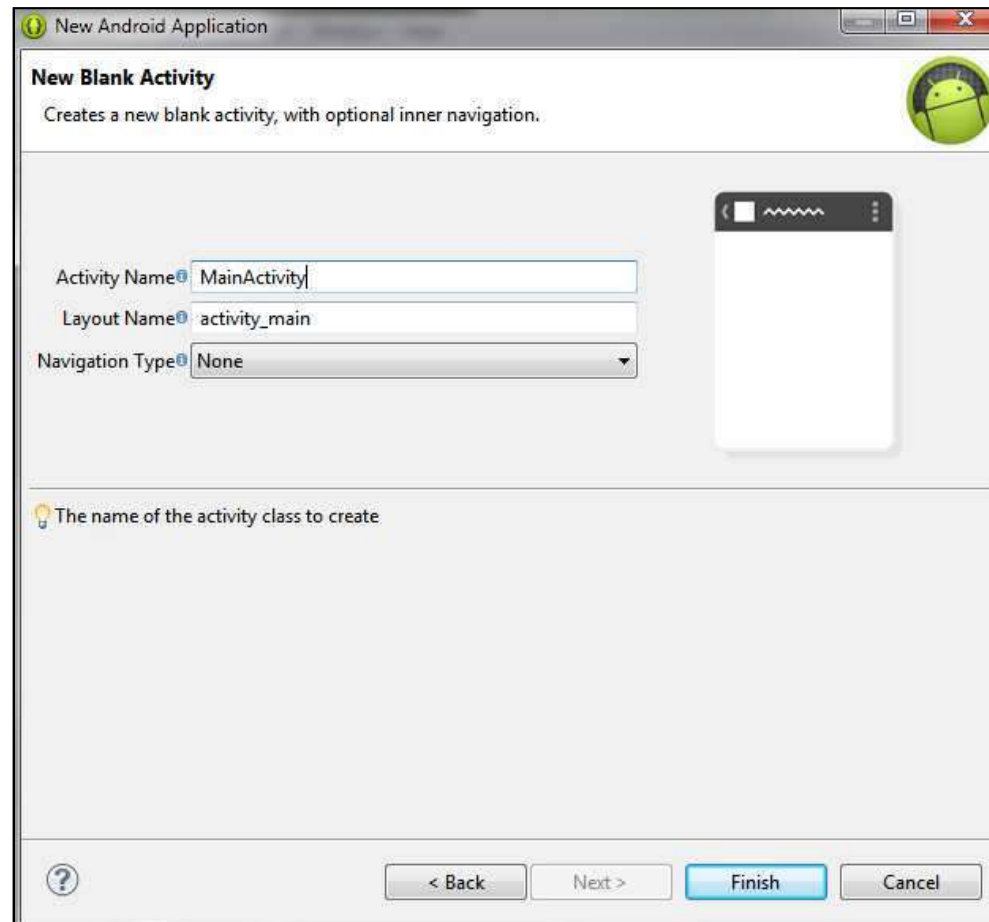
Créer un nouveau projet Android (5)

41



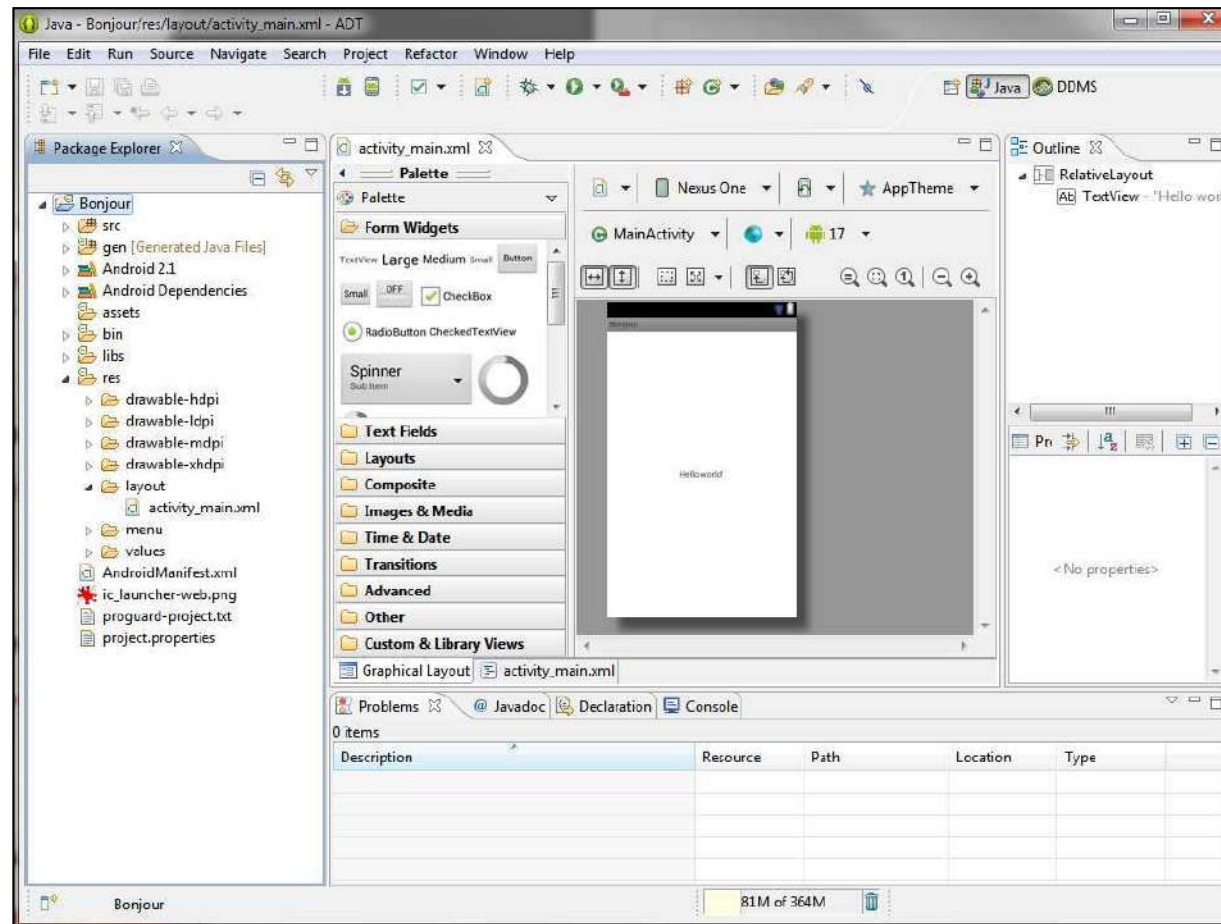
Créer un nouveau projet Android (6)

42



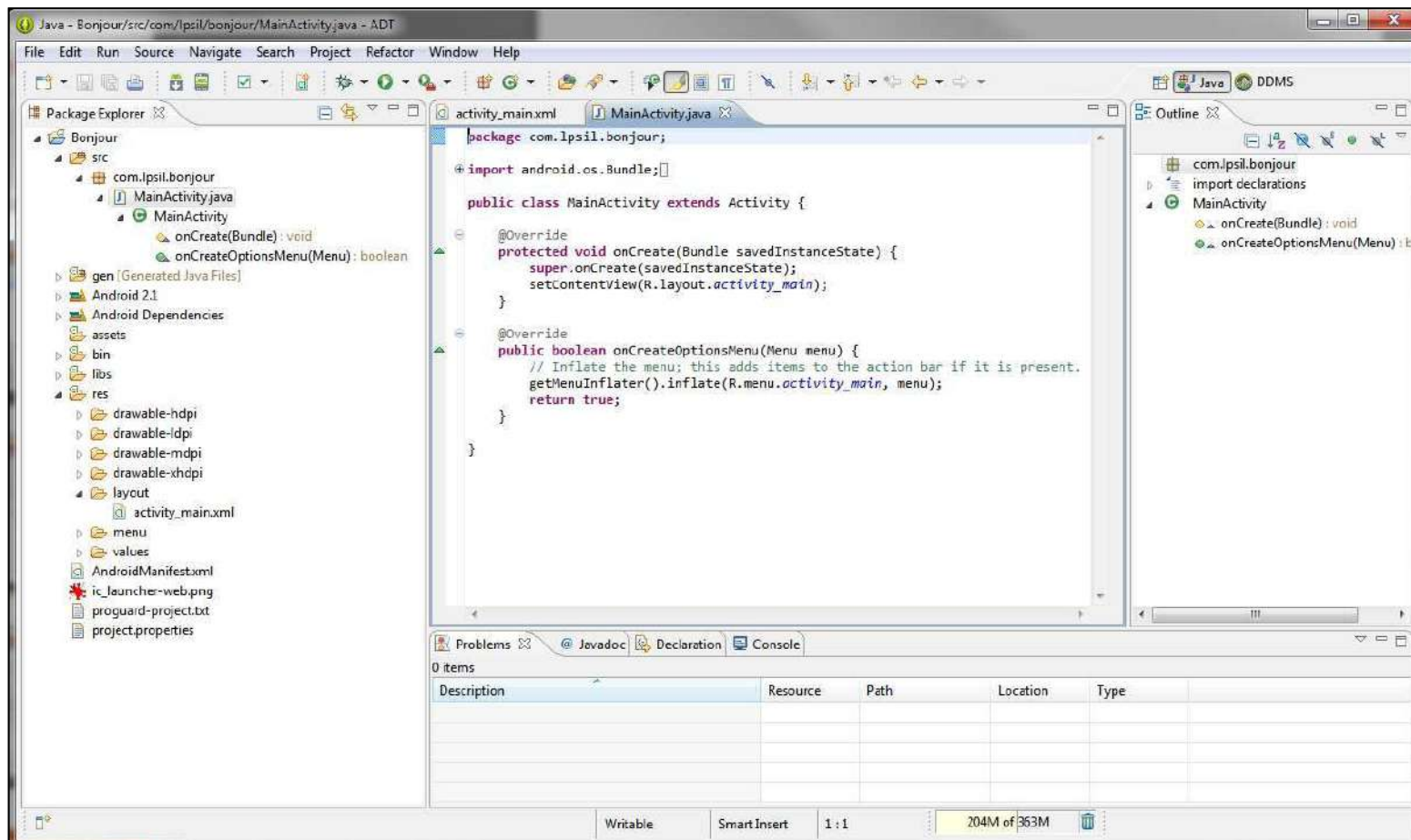
Aperçu du premier projet (1)

43



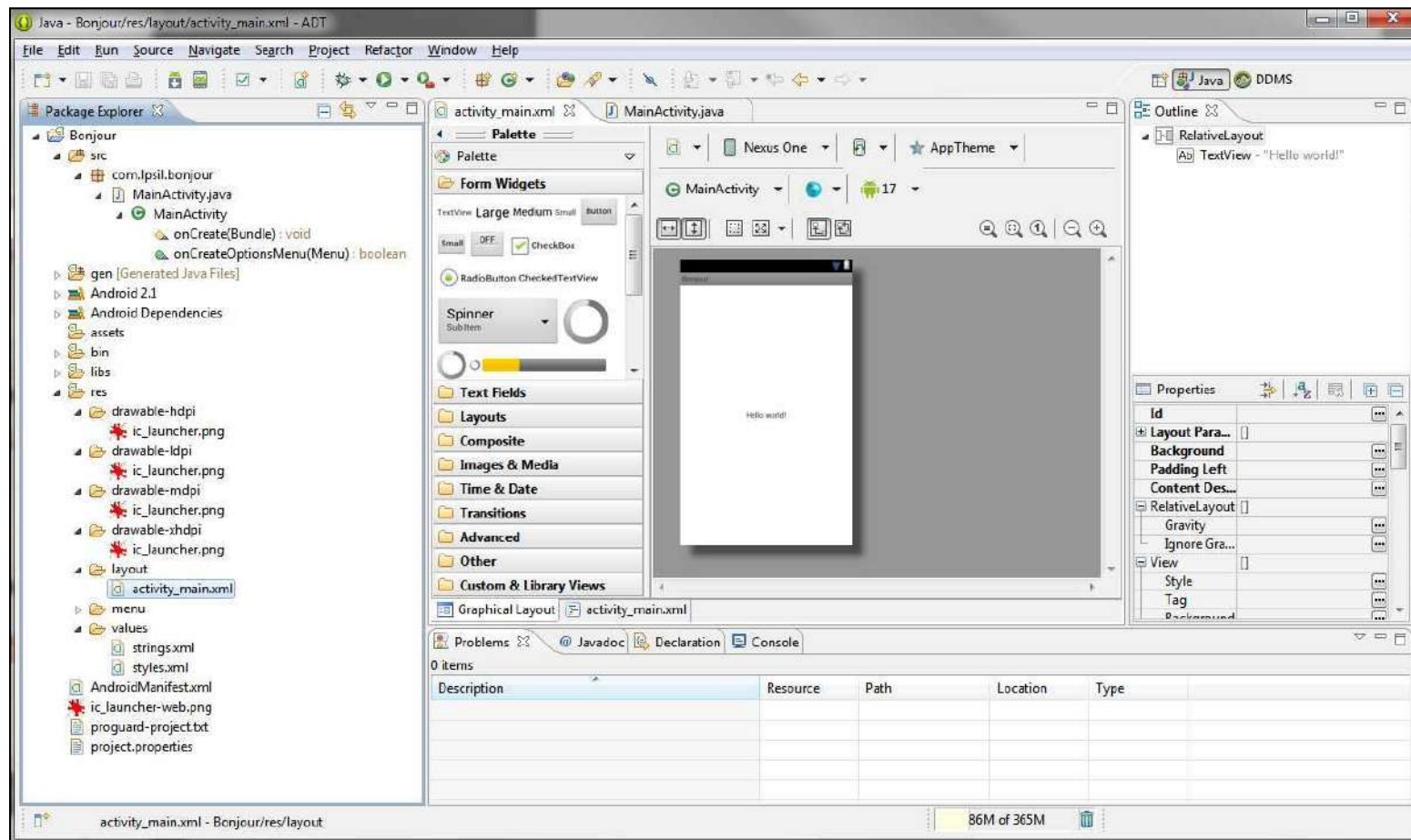
Aperçu du premier projet (2)

44



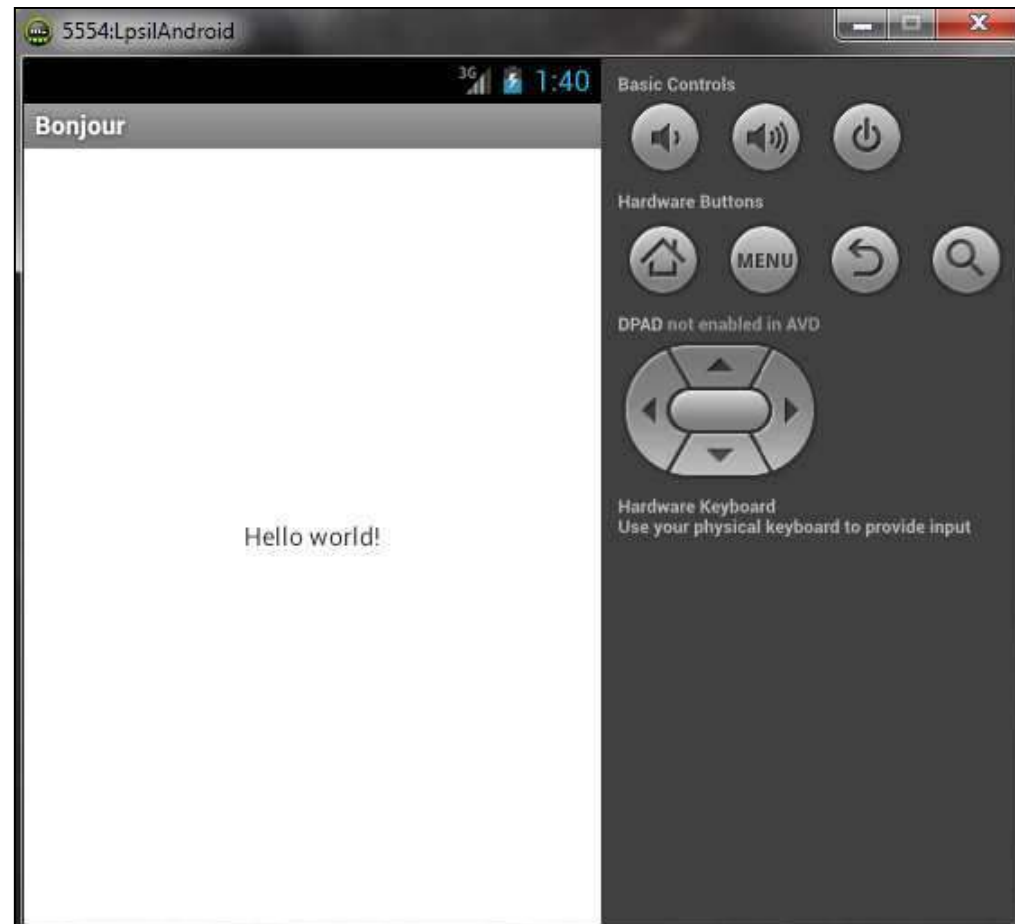
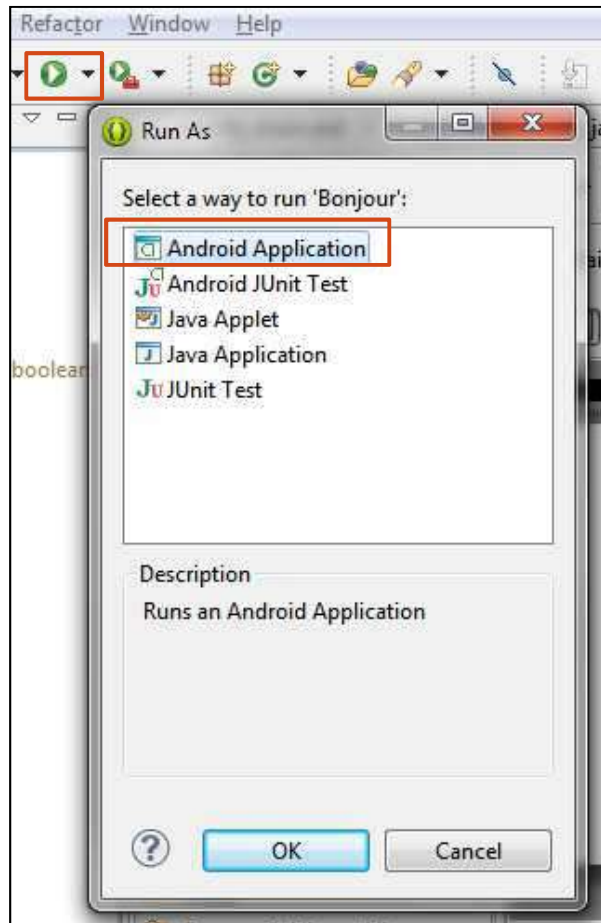
Aperçu du premier projet (3)

45



Lancement de l'application dans l'émulateur

46



Etude de la structure du projet (1)

47

Élément considéré	Description
Répertoire src	Le répertoire de l'ensemble des sources du projet. C'est dans ce répertoire que vous allez ajouter et modifier le code source de l'application.
Répertoire libs	Contient les bibliothèques tierces qui serviront à votre application.
Répertoire res	Contient toutes les ressources telles que les images, les dispositions de l'interface graphique, etc. nécessaires à l'application. Ces ressources seront accessibles grâce à la classe R décrite plus loin.
Répertoire gen	Contient l'ensemble des fichiers générés par ADT afin d'assister le développeur. Si vous supprimez un fichier dans ce répertoire, ADT s'empressera aussitôt de le régénérer. Attention , si le projet contient des erreurs ne permettant pas sa compilation, ADT ne générera aucun fichier. Par conséquent, si vous ne trouvez pas la bonne version de fichier ou si le fichier n'est pas généré automatiquement, vérifiez bien que l'ensemble du projet compile.
Répertoire assets	Contient toutes les ressources brutes (raw bytes) ne nécessitant aucun traitement par ADT ou Android. À la différence des ressources placées dans le répertoire res, les ressources brutes seront accessibles grâce à un flux de données et non grâce à la classe R décrite plus loin.
Fichier AndroidManifest.xml	Le fichier XML décrivant l'application et ses composants – activités, services, etc.
Fichier project.properties	Fichier de propriétés utilisé pour la compilation.

Etude de la structure du projet (2)

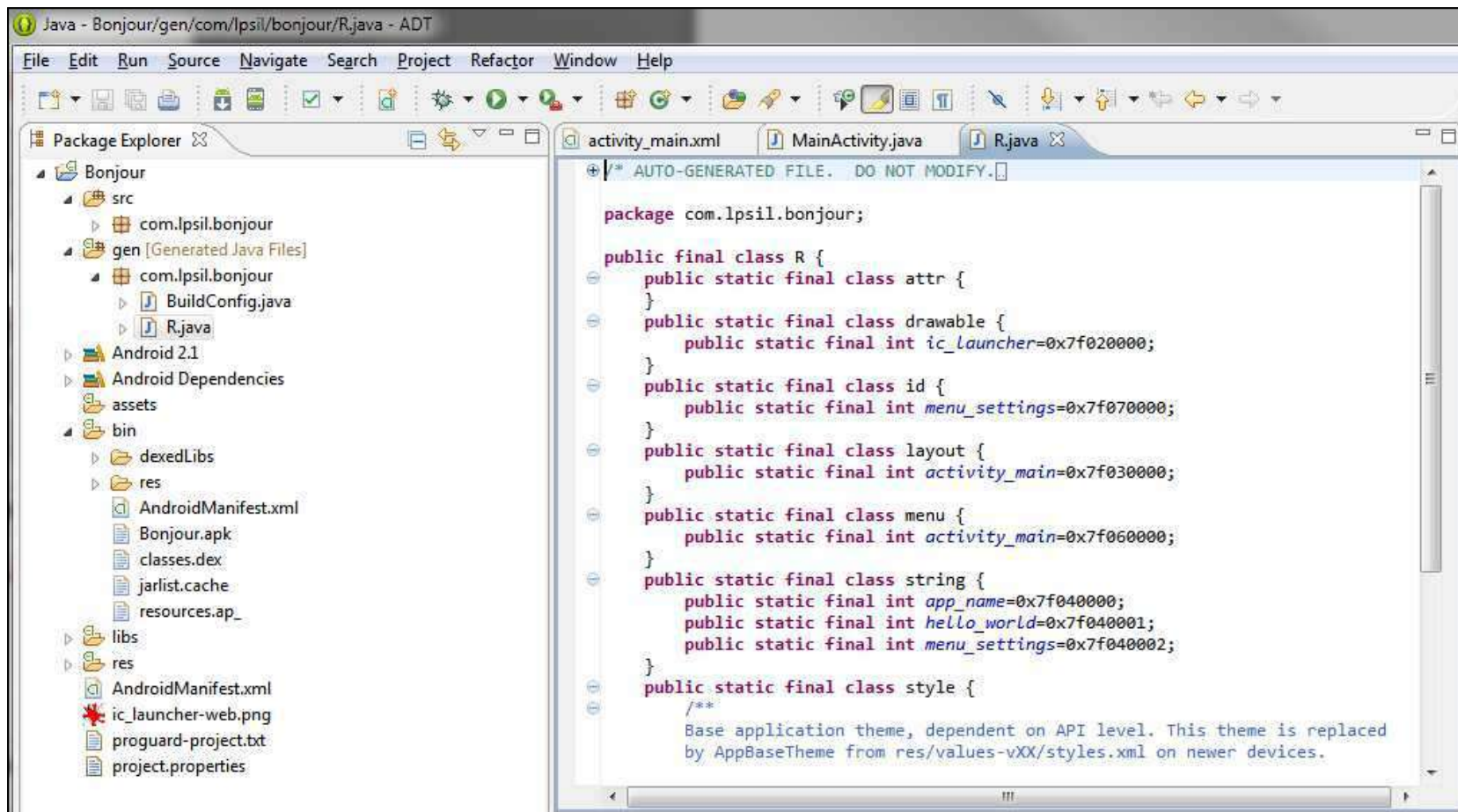
48

- Le répertoire `res` est composé de différents sous-répertoires dont les suivants :
 - `res/drawable` contient les ressources de type image (PNG, JPEG et GIF) ;
 - `res/layout` contient les descriptions des interfaces utilisateur ;
 - `res/menu` contient la description des menus ;
 - `res/values` contient les chaînes de caractères, les dimensions, etc. ;

- Dans les versions précédentes, il y avait aussi:
 - `res/xml` contenait des fichiers XML supplémentaires (préférences, etc.) ;
 - `res/raw` contenait des ressources autres que celles décrites ci-dessus qui seront empaquetées sans aucun traitement.

Etude de la structure du projet (3)

49



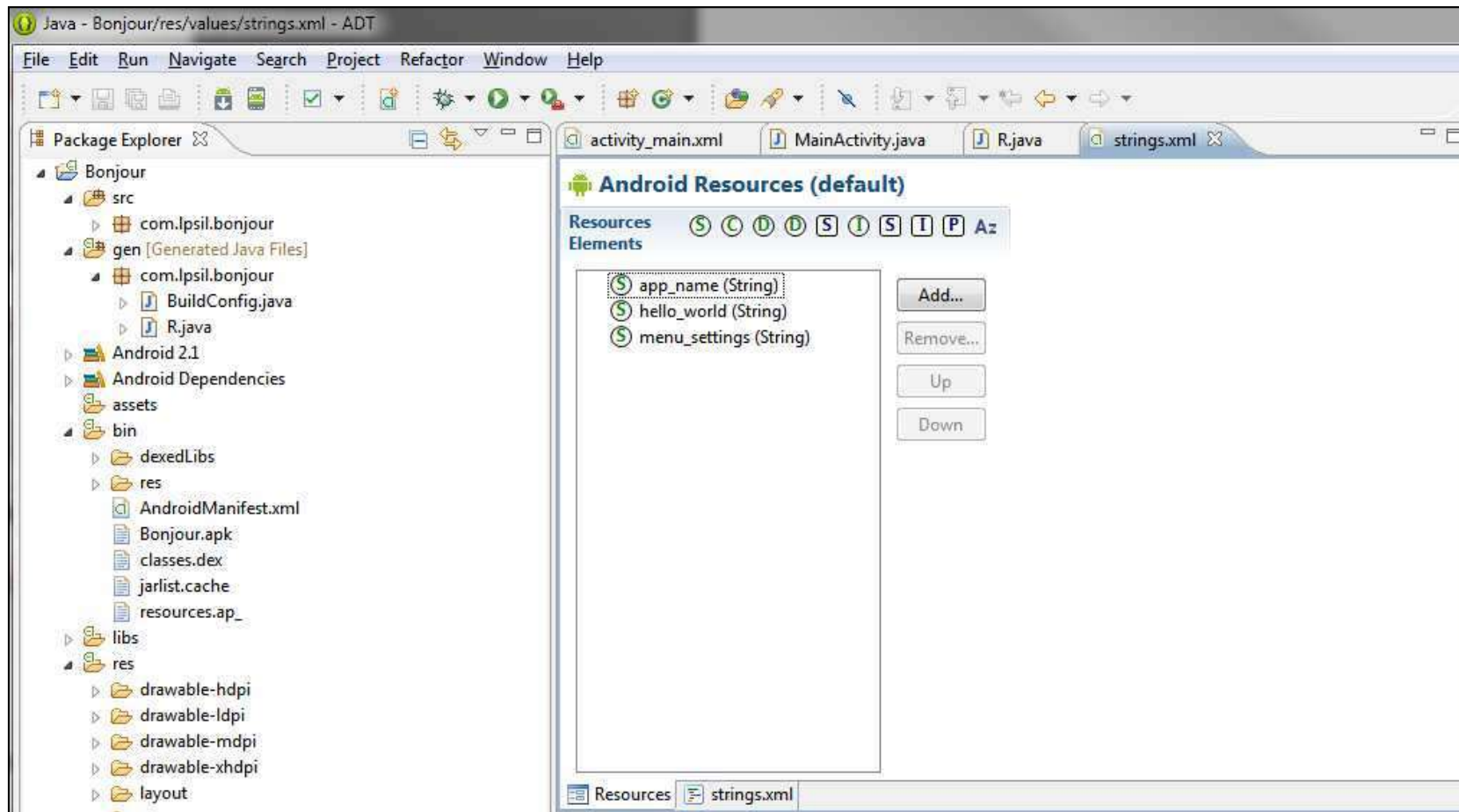
The screenshot shows an IDE window titled "Java - Bonjour/gen/com/ipsil/bonjour/R.java - ADT". The interface includes a menu bar (File, Edit, Run, Source, Navigate, Search, Project, Refactor, Window, Help), a toolbar, and a Package Explorer on the left. The Package Explorer shows the project structure for "Bonjour", including folders for "src", "gen", "Android 2.1", "Android Dependencies", "assets", "bin", "libs", and "res". The "gen" folder is expanded, showing "com.ipsil.bonjour" with files "BuildConfig.java" and "R.java". The main editor displays the content of "R.java", which is an auto-generated file. The code defines a public final class R containing several static final classes: attr, drawable, id, layout, menu, string, and style. Each class contains static final fields representing resource IDs, such as ic_launcher, menu_settings, activity_main, app_name, hello_world, and menu_settings. A comment at the bottom explains that the style class is a base application theme that is replaced by AppBaseTheme from res/values-vXX/styles.xml on newer devices.

```
package com.ipsil.bonjour;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int menu_settings=0x7f070000;
    }
    public static final class layout {
        public static final int activity_main=0x7f030000;
    }
    public static final class menu {
        public static final int activity_main=0x7f060000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int hello_world=0x7f040001;
        public static final int menu_settings=0x7f040002;
    }
    public static final class style {
        /**
         Base application theme, dependent on API level. This theme is replaced
         by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
```

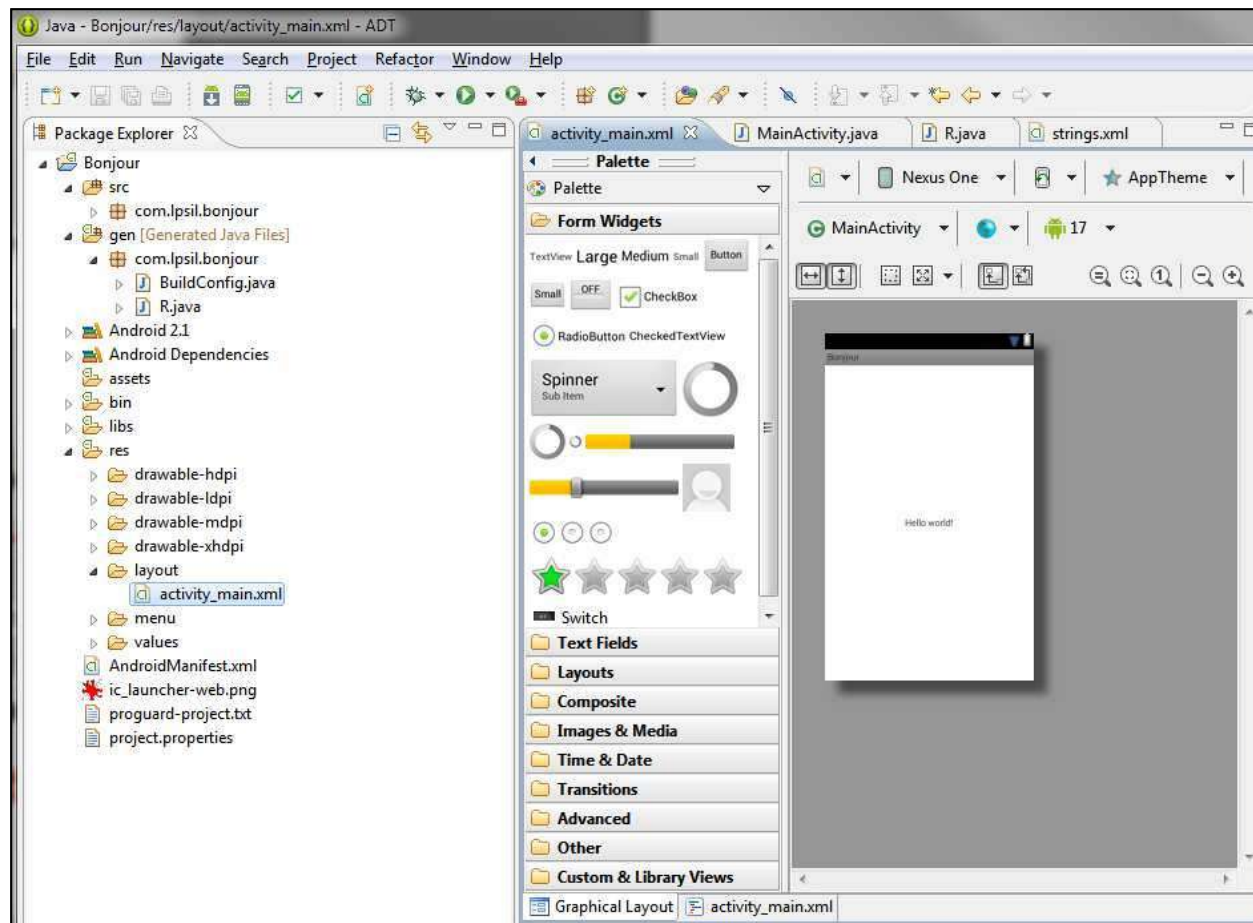
Etude de la structure du projet (4)

50



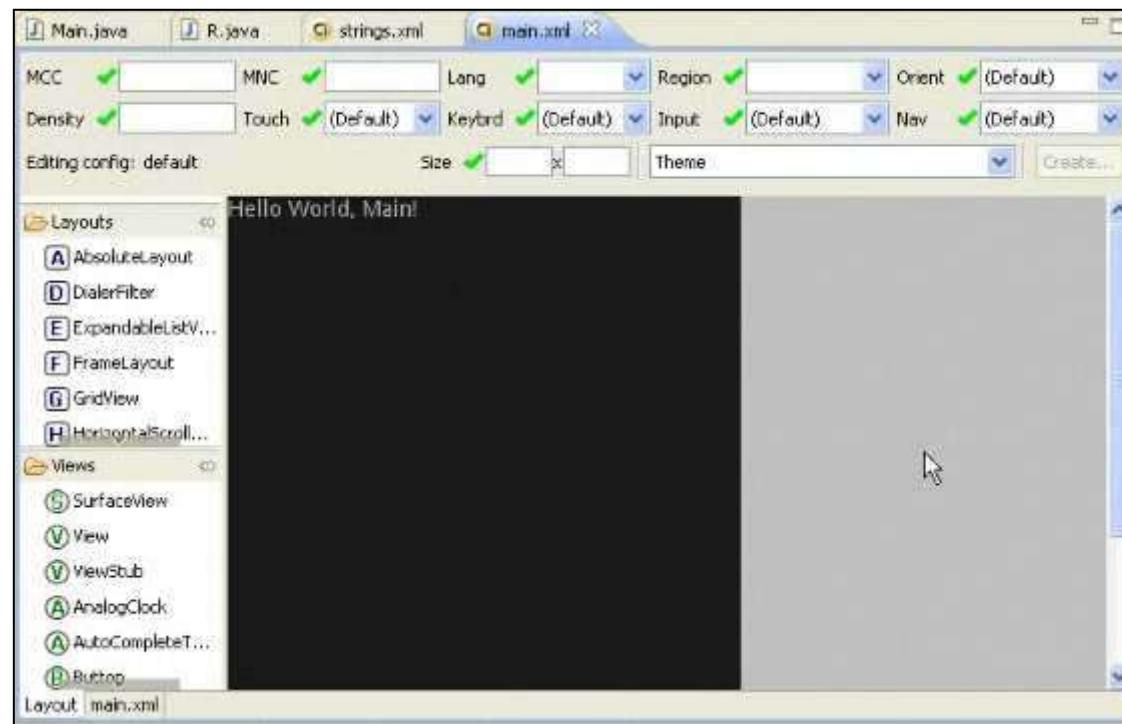
Etude de la structure du projet (5)

51



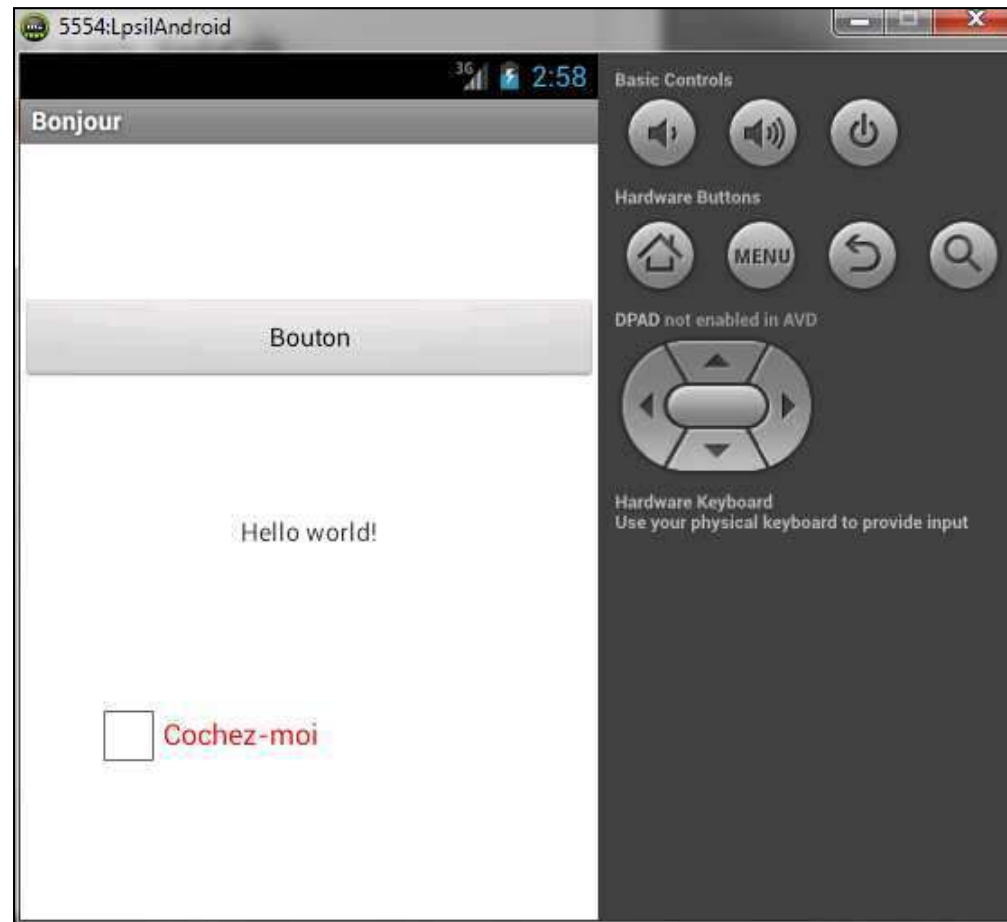
Anciennes versions...

52



L'application Bonjour modifiée

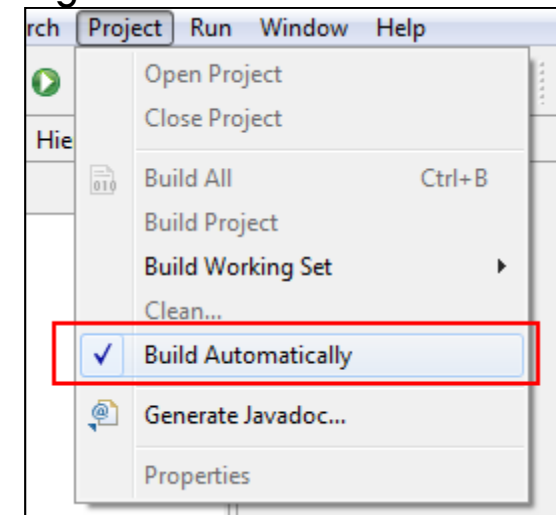
53



Remarques sur la compilation

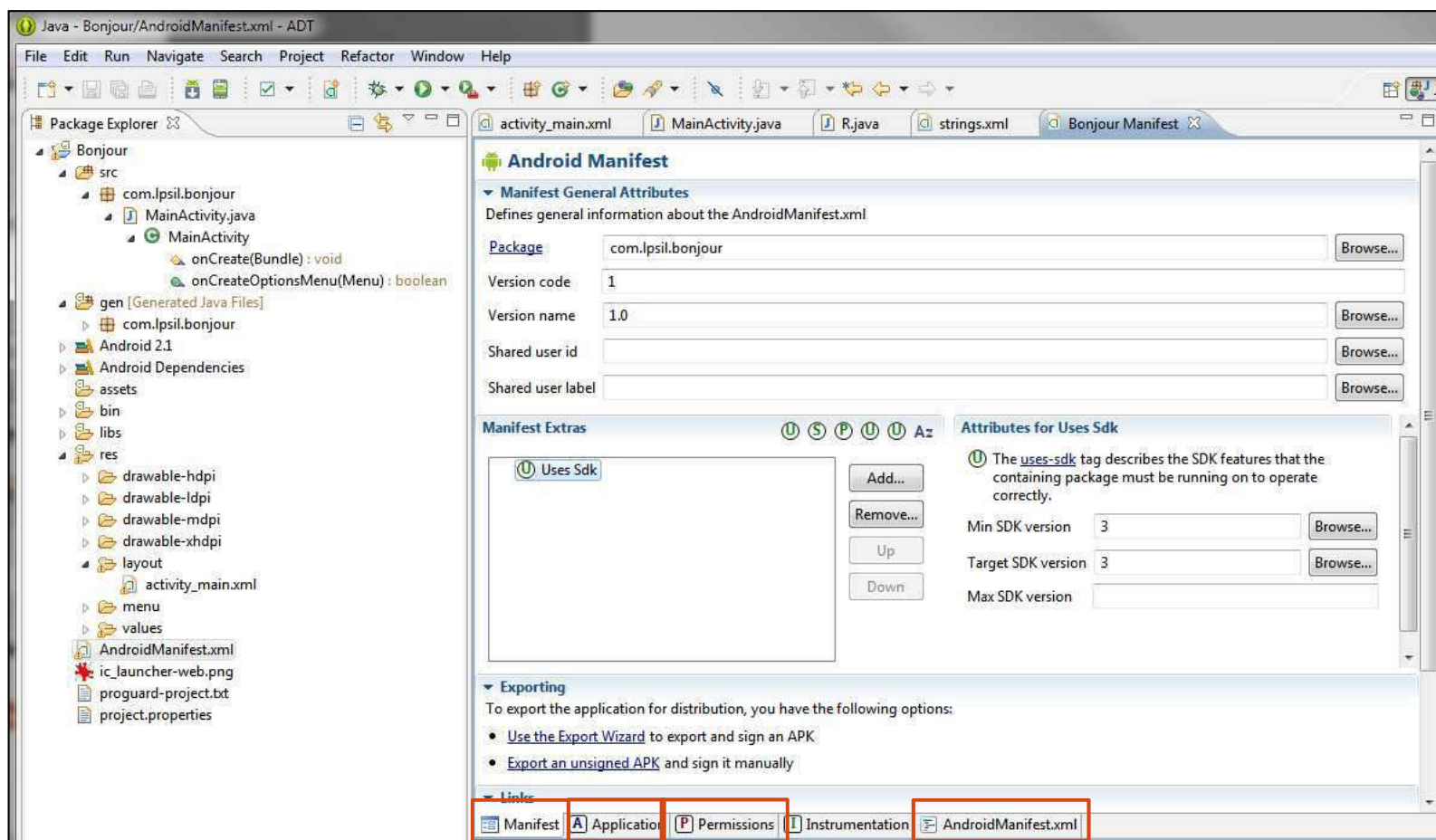
54

- Méthode de compilation – Menu Projet > case Build Automatically cochée
 - Les projets sont compilés automatiquement dès qu'on enregistre un fichier source.
 - Seulement si pas d'erreurs dans le code
 - L'application exécutée est la dernière version compilée
- Le résultat d'une compilation sera placé selon l'arborescence suivante :
 - **bin/classes.dex** : représente l'exécutable compilé pour la machine virtuelle Dalvik (résultat de la compilation du code source);
 - **bin/res** : contient les ressources de votre application;
 - **bin/Bonjour.apk** : il s'agit de votre application compilée et empaquetée pour être déployée dans le système Android.
Ce fichier contient le fichier **.dex**, les ressources compilées et non compilées (celles contenues dans le répertoire **/raw**) et enfin le fichier de configuration de l'application.



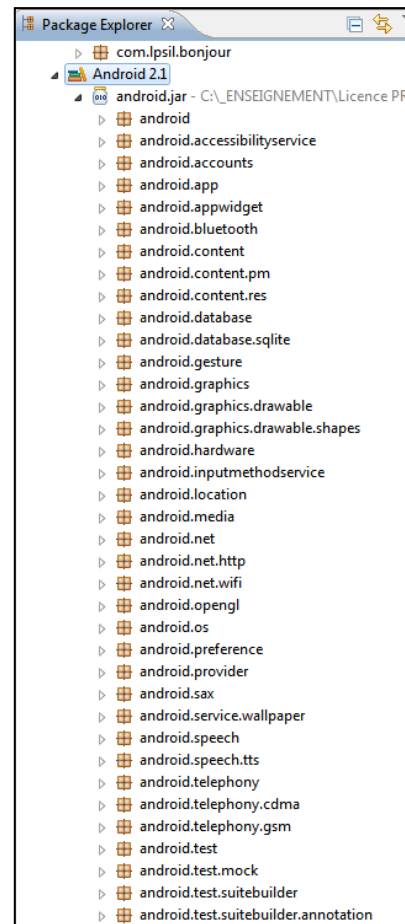
Fichier AndroidManifest.xml

55



Librairies

56



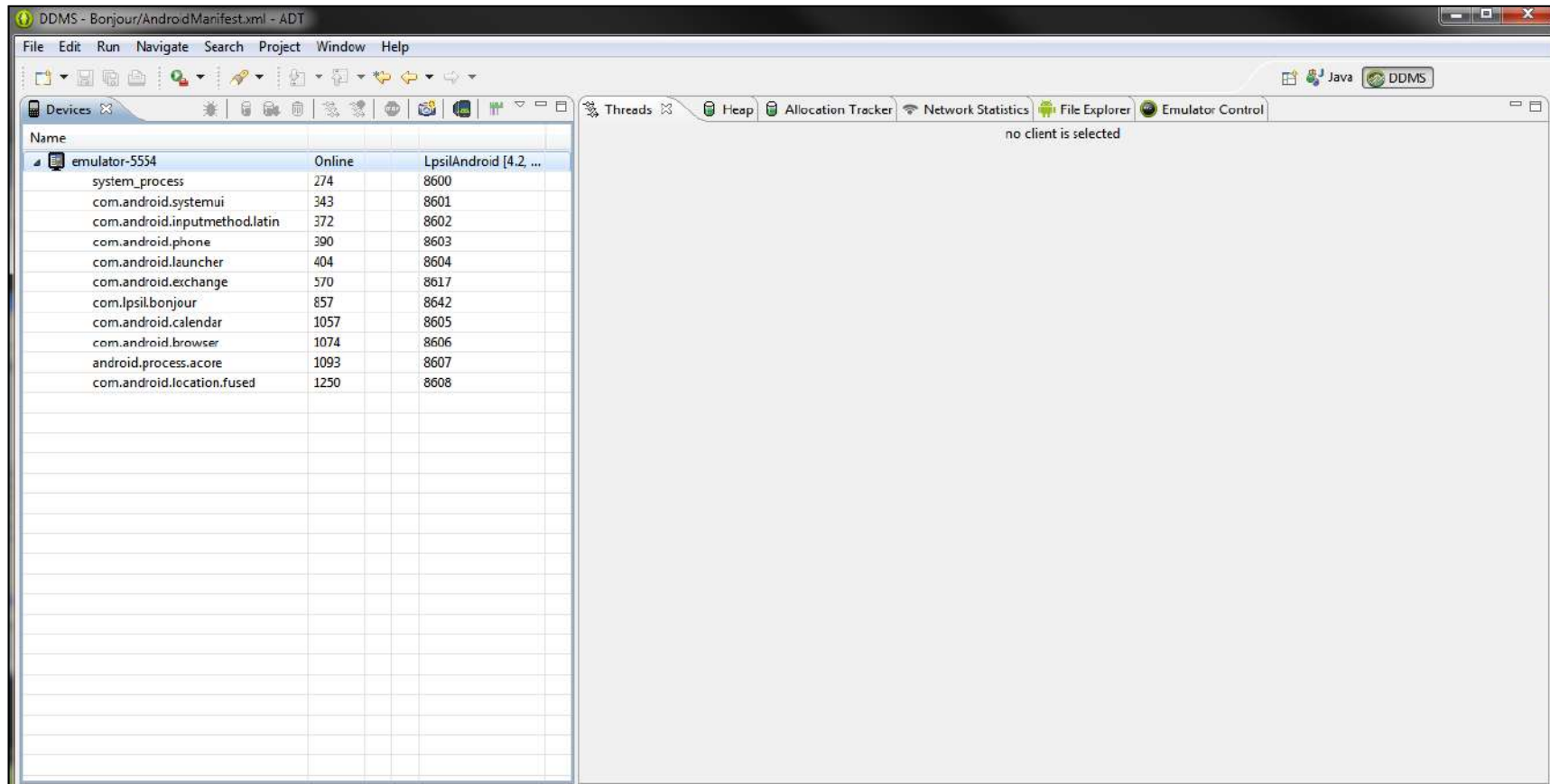
**5^e
partie**

Perspectives et DDMS

Dalvik Debug Monitor Service

Ecran d'accueil d'Eclipse DDMS

58



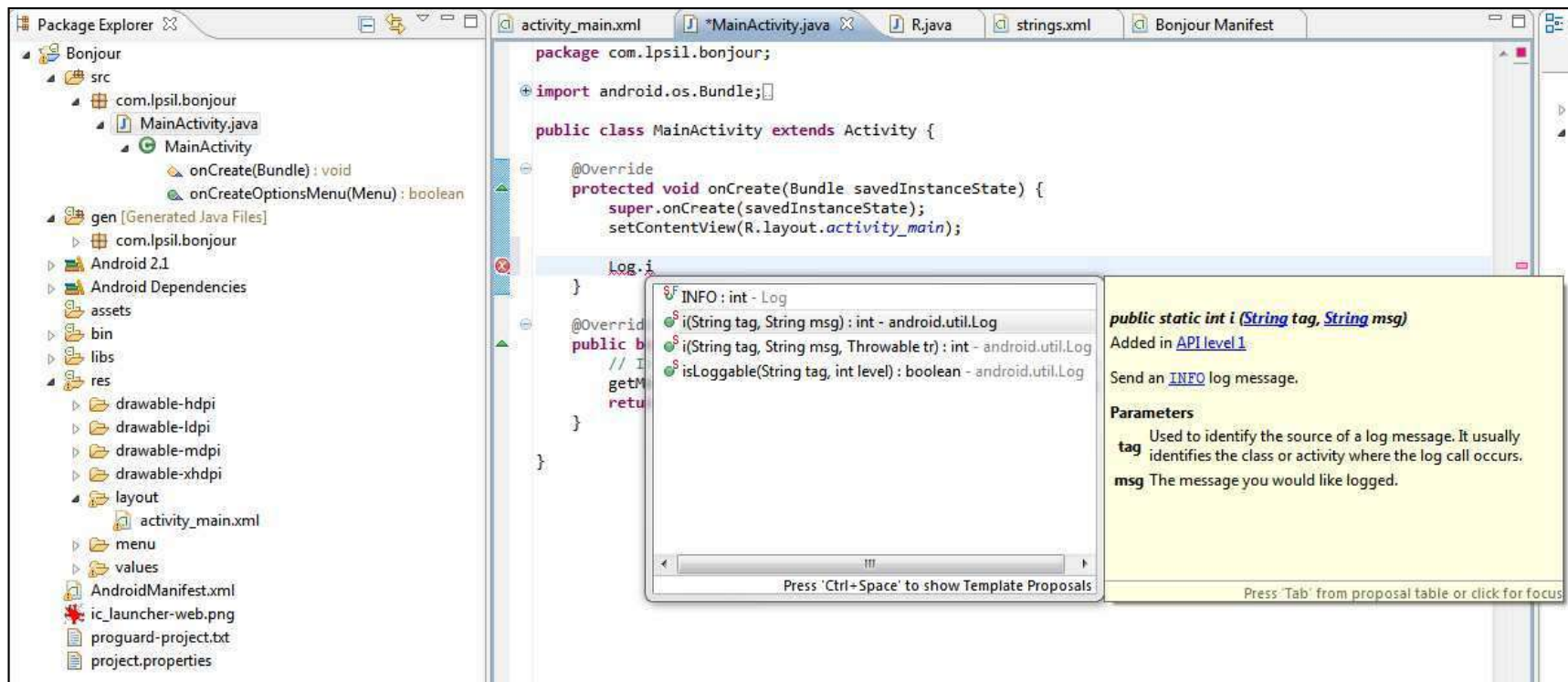
Le LogCat – vue d'ensemble

59

Time	pid	tag	Message
07-22 16:01	D 1344	Andro...	Shutting down VM
07-22 16:01	D 1344	dalvikv	DestroyJavaVM waiting for non-dae...
07-22 16:01	D 1344	dalvikv	DestroyJavaVM shutting VM down
07-22 16:01	D 1344	dalvikv	HeapWorker thread shutting down
07-22 16:01	D 1344	dalvikv	HeapWorker thread has shut down
07-22 16:01	D 1344	jdvp	JDWP shutting down net...
07-22 16:01	D 1344	jdvp	+++ peer disconnected
07-22 16:01	I 1344	dalvikv	Debugger has detached, object reg...
07-22 16:01	D 1344	dalvikv	VM cleaning up
07-22 16:01	D 1344	dalvikv	LinearAlloc 0x0 used 629804 of 41...
07-22 16:01	D 1357	Andro...	>>>>>>>>>>>>>>>> AndroidRuntime STA...
07-22 16:01	D 1357	Andro...	CheckJNI is ON
07-22 16:01	D 1357	Andro...	--- registering native functions ---
07-22 16:01	I 1357	jdvp	received file descriptor 20 from ADB
07-22 16:01	D 1357	Andro...	Shutting down VM
07-22 16:01	D 1357	dalvikv	DestroyJavaVM waiting for non-dae...
07-22 16:01	D 1357	dalvikv	DestroyJavaVM shutting VM down
07-22 16:01	D 1357	dalvikv	HeapWorker thread shutting down
07-22 16:01	D 1357	dalvikv	HeapWorker thread has shut down
07-22 16:01	D 1357	jdvp	JDWP shutting down net...
07-22 16:01	D 1357	jdvp	+++ peer disconnected

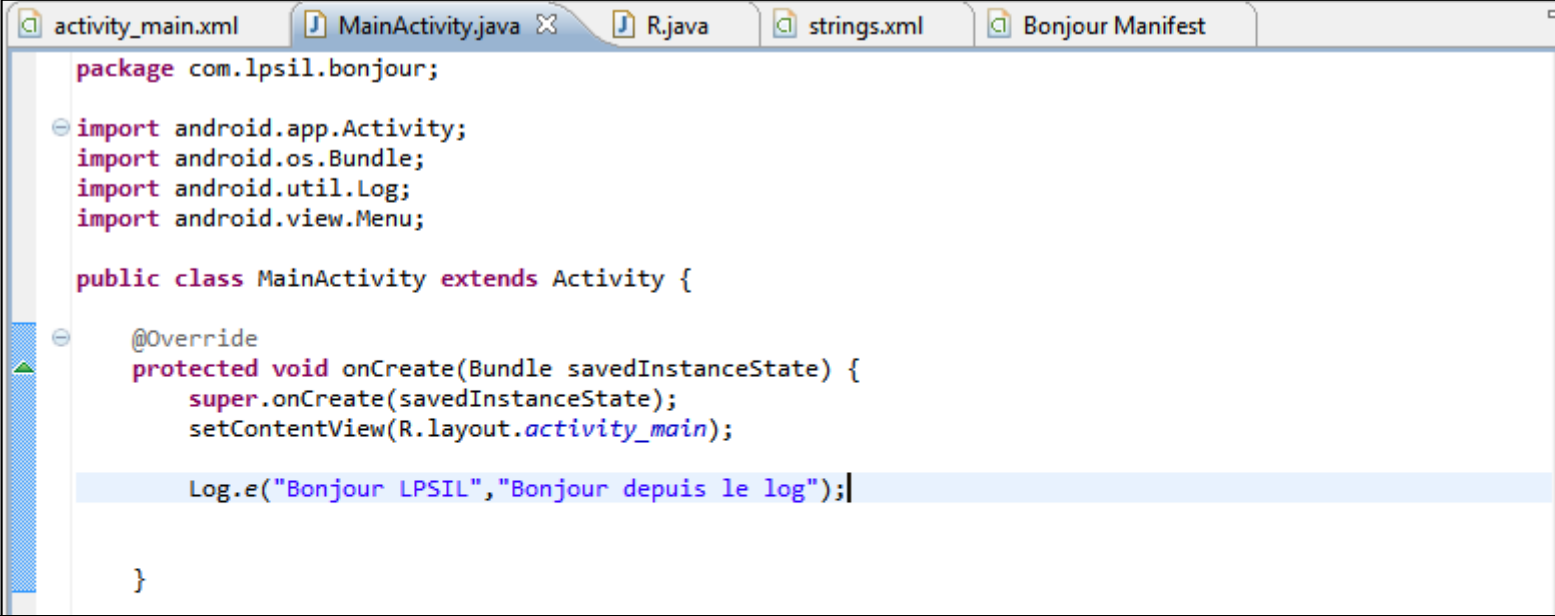
Le LogCat – utilisation dans le cadre d'un projet (1)

60



Le LogCat – utilisation dans le cadre d'un projet (2)

61



The screenshot shows an IDE window with the following code in MainActivity.java:

```
package com.lpsil.bonjour;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;

public class MainActivity extends Activity {

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

        Log.e("Bonjour LPSIL", "Bonjour depuis le log");
    }
}
```

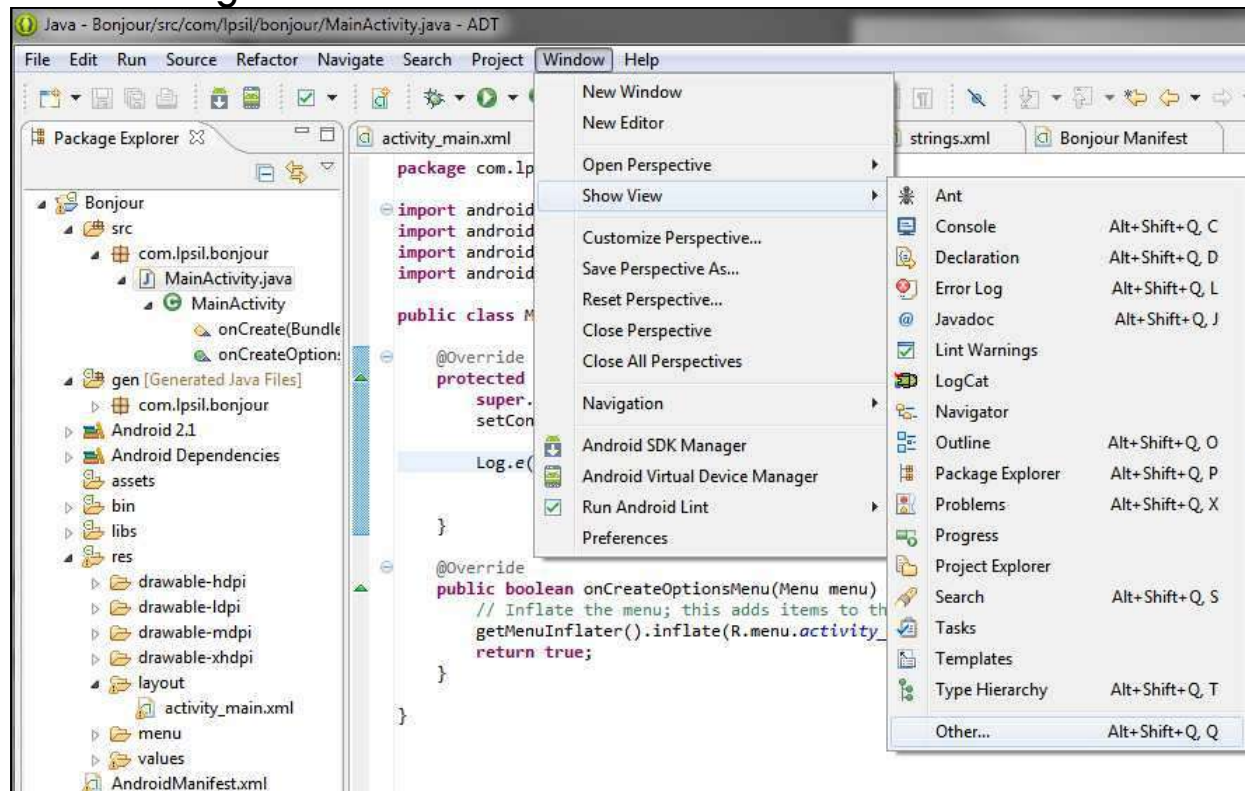
Below the code editor is the LogCat window. It shows a search bar and a table of log messages. The table has columns: L..., Time, PID, TID, Application, Tag, and Text. A single log message is visible:

L...	Time	PID	TID	Application	Tag	Text
E	01-08 16:41:16.177	7680	7680	com.lpsil.bonjour	Bonjour LPSIL	Bonjour depuis le log

Ajouter une vue à une perspective

62

- Menu Window > lien Show View > Other...
- Android > LogCat

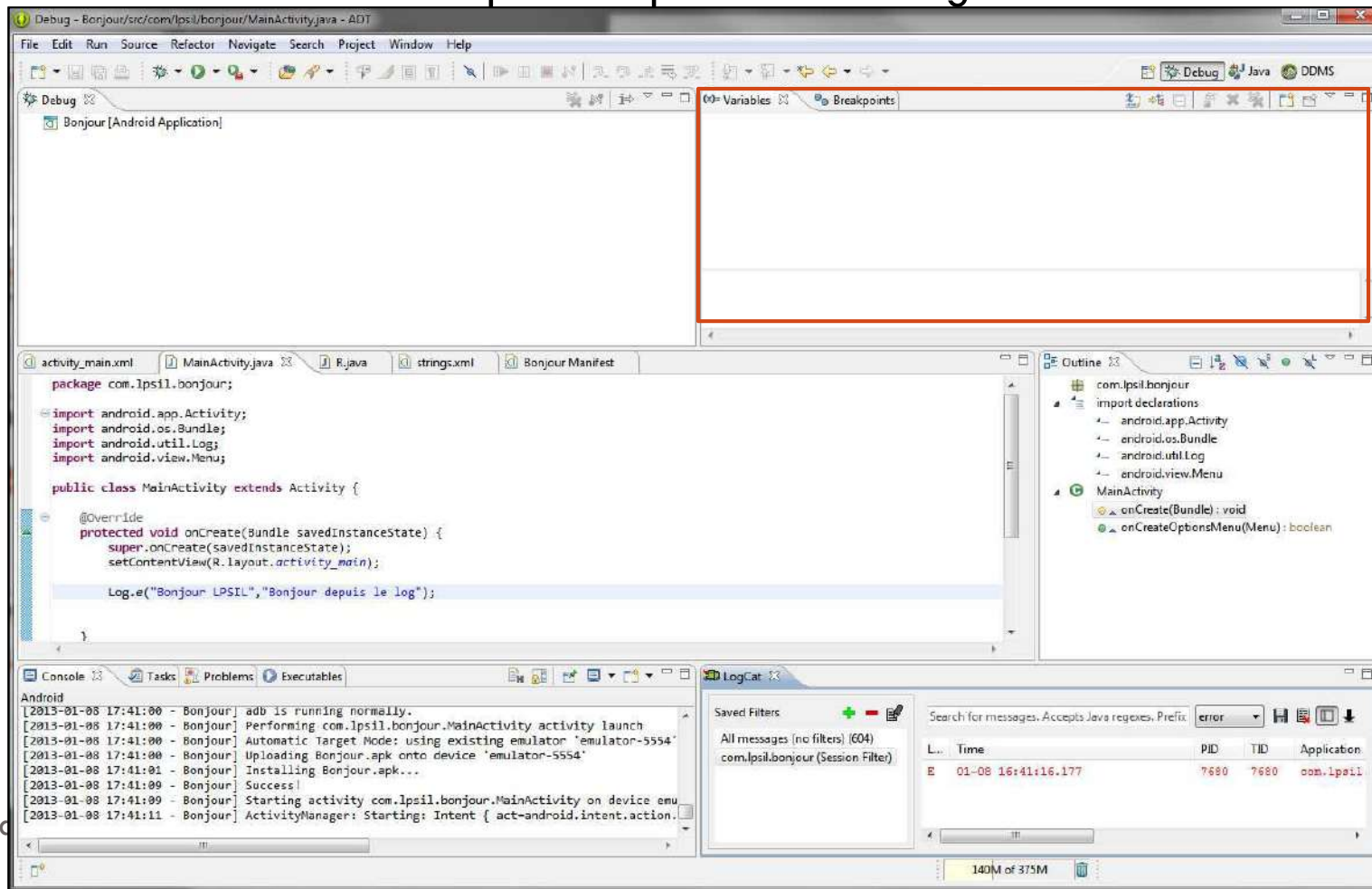


**6^e
partie**

Le Débuggeur

Passer en perspective debug

- Menu Window > Lien Open Perspective > Debug



Définir un point d'arrêt

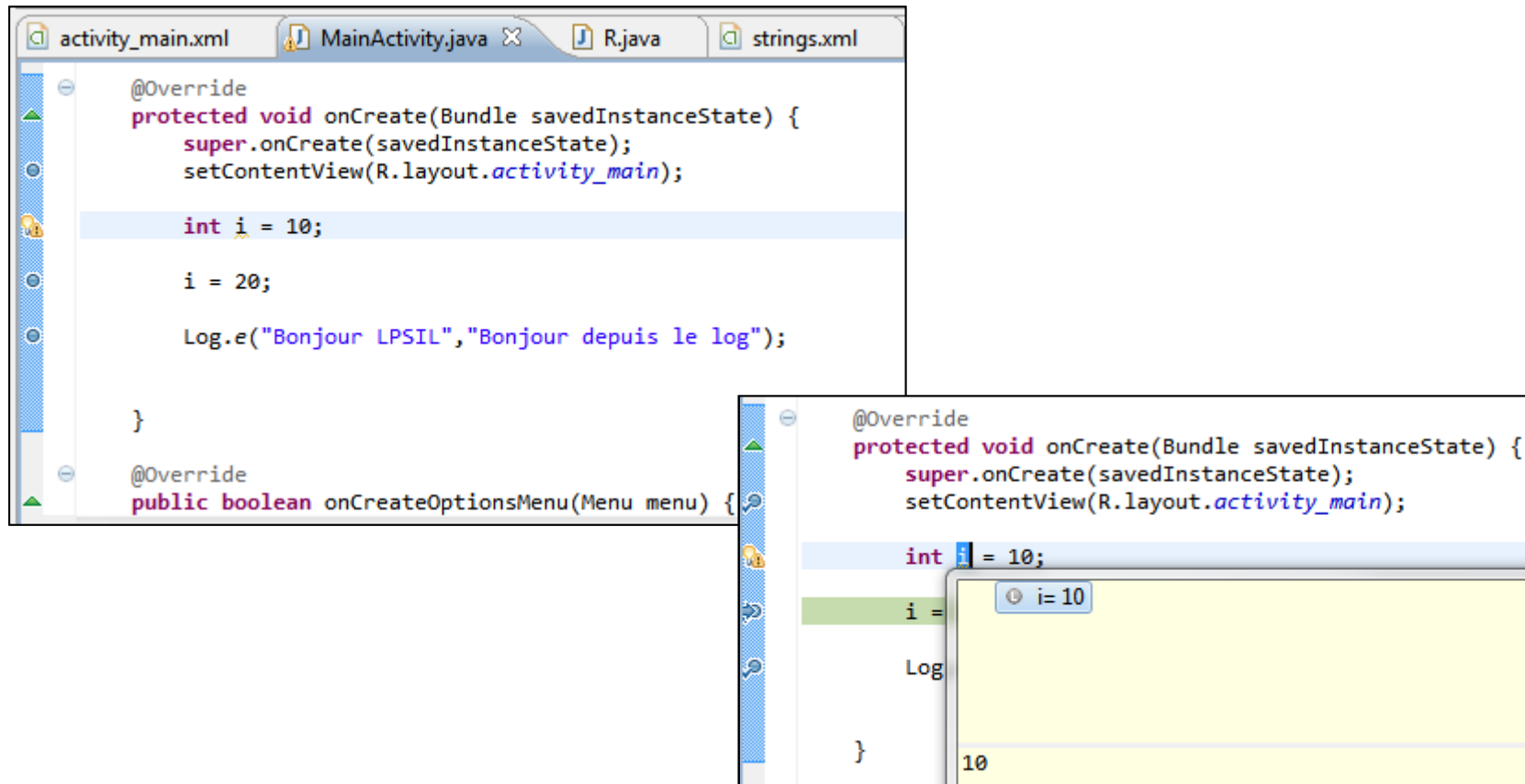
65

The screenshot displays the Android Studio IDE with the following components:

- Debugger:** Shows a suspended thread at line 13 of MainActivity.java. The stack trace includes MainActivity.onCreate(Bundle) and various system methods.
- Variables:** Shows the 'this' variable pointing to MainActivity (id=830015930808) and 'savedInstanceState' as null.
- Code Editor:** Shows the MainActivity.java source code with a breakpoint set on the `setContentView(R.layout.activity_main);` line. The code includes package declarations, imports, and the `onCreate` method.
- LogCat:** Shows system logs for the application launch, including messages like 'adb is running normally', 'Performing com.lpsil.bonjour.MainActivity activity launch', and 'Attempting to connect debugger to 'com.lpsil.bonjour' on port'.

Surveiller les valeurs des variables

66



The image shows two overlapping screenshots of an IDE window displaying the `MainActivity.java` file. The top screenshot shows the `onCreate` method with the following code:

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

    int i = 10;

    i = 20;

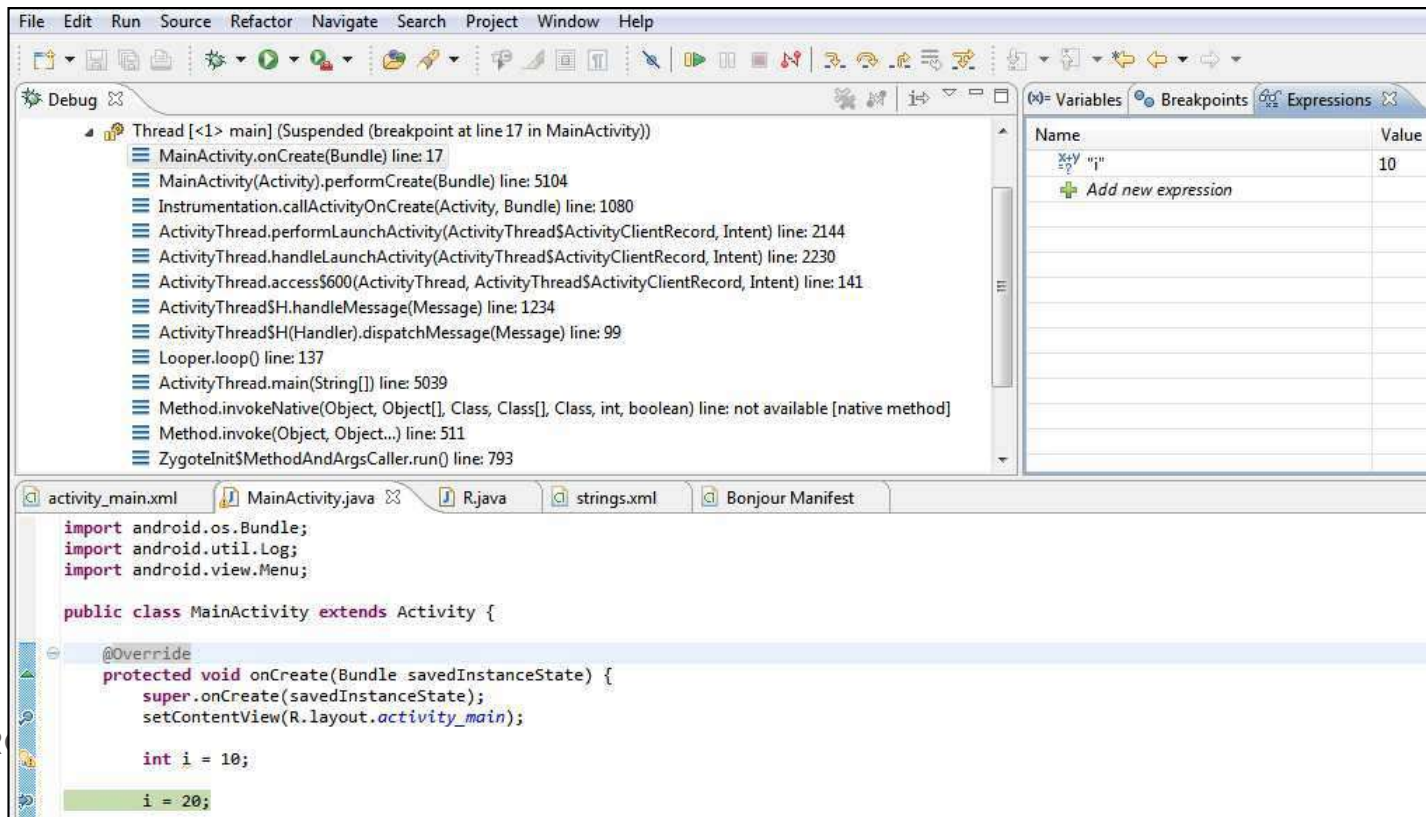
    Log.e("Bonjour LPSIL", "Bonjour depuis le log");
}
```

The bottom screenshot shows the same code, but with a tooltip over the variable `i` in the line `int i = 10;`. The tooltip displays the value `i = 10`. The `Log` statement is also visible in the bottom screenshot.

Ajouter des espions (expressions)

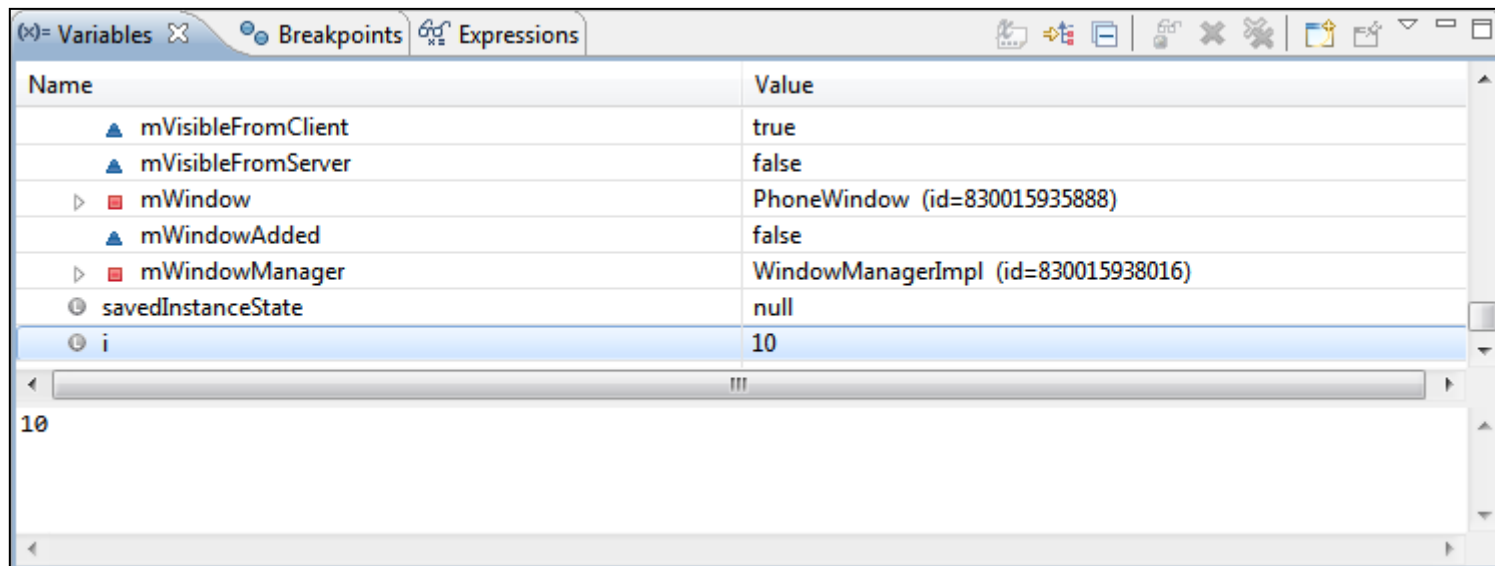
67

- Menu Window > Show View > Other...
 - ▣ Debug > Expressions
- Onglet « Expressions » > Lien Add new expression... > « i »



Etat des variables

68



The screenshot shows the 'Variables' window in an IDE. The window has tabs for 'Variables', 'Breakpoints', and 'Expressions'. The 'Variables' tab is active, displaying a table of variables and their values. The variable 'i' is selected and highlighted in blue. Below the table, the value '10' is displayed in a text area.

Name	Value
▲ mVisibleFromClient	true
▲ mVisibleFromServer	false
▷ ■ mWindow	PhoneWindow (id=830015935888)
▲ mWindowAdded	false
▷ ■ mWindowManager	WindowManagerImpl (id=830015938016)
⊙ savedInstanceState	null
⊙ i	10

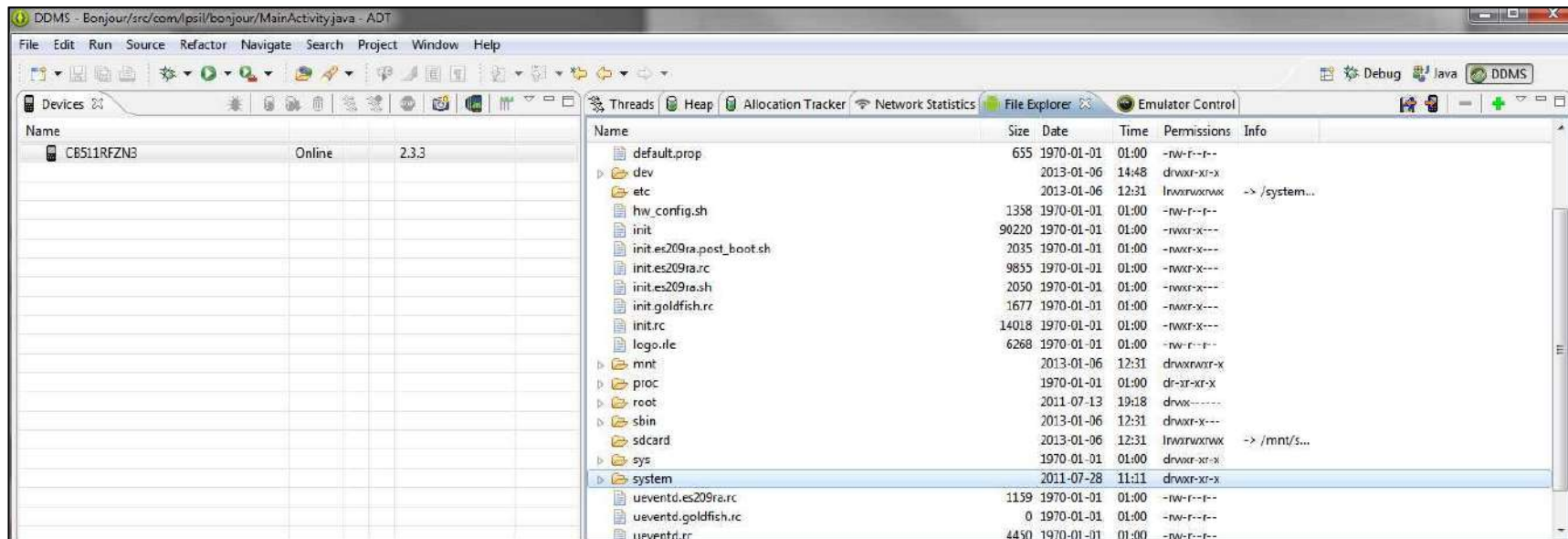
10

**7^e
partie**

Le pilote USB

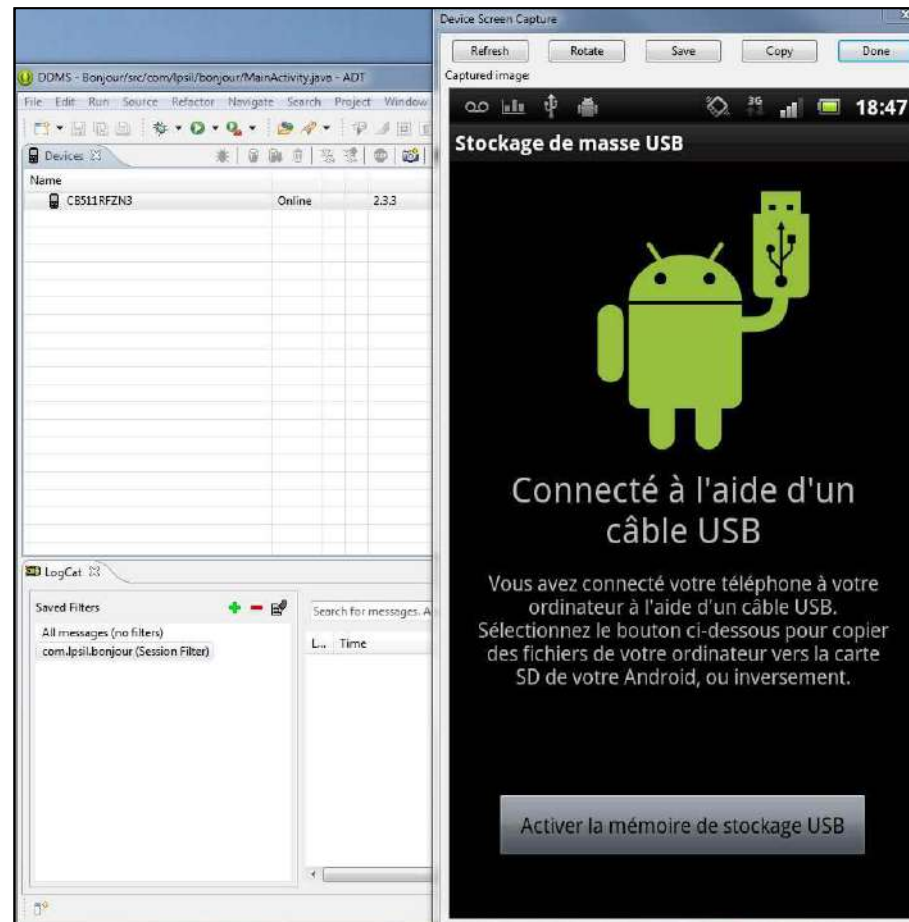
Accès au terminal mobile avec Eclipse

70



Faire une capture d'écran

71



Paramétrer le terminal mobile

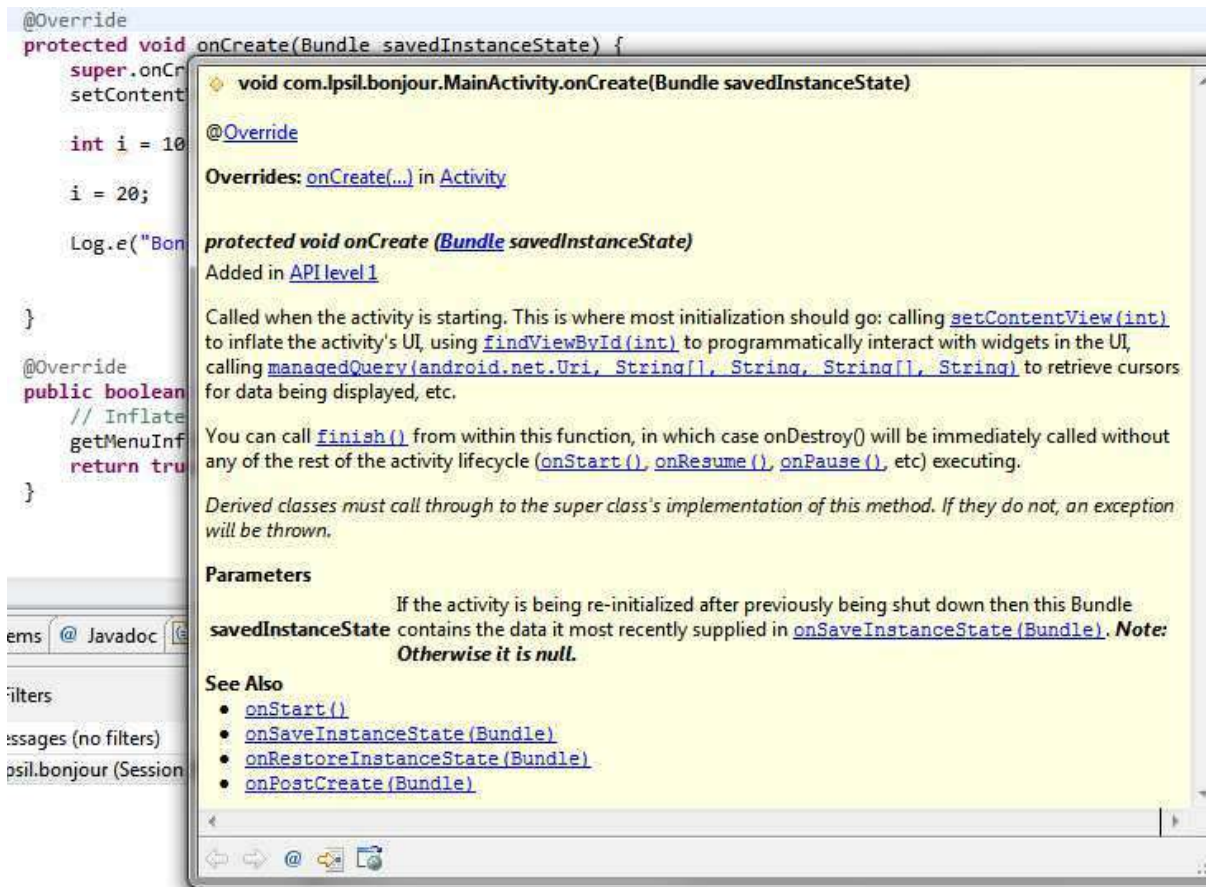
72

- Brancher la tablette avec le câble USB fourni
- Sur la tablette, aller dans **Paramètres > Applications**
 - cocher la case **Sources inconnues**
 - Menu **Développement** > cocher la case **Débogage USB**

**8^e
partie**

Les aides et astuces

Aide dynamique



The screenshot shows an IDE with a code editor on the left and a dynamic help popup on the right. The code editor contains the following Java code:

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

    int i = 10;
    i = 20;

    Log.e("Bonjour", "Bonjour");
}

@Override
public boolean onCreateOptionsMenu() {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, this);
    return true;
}
```

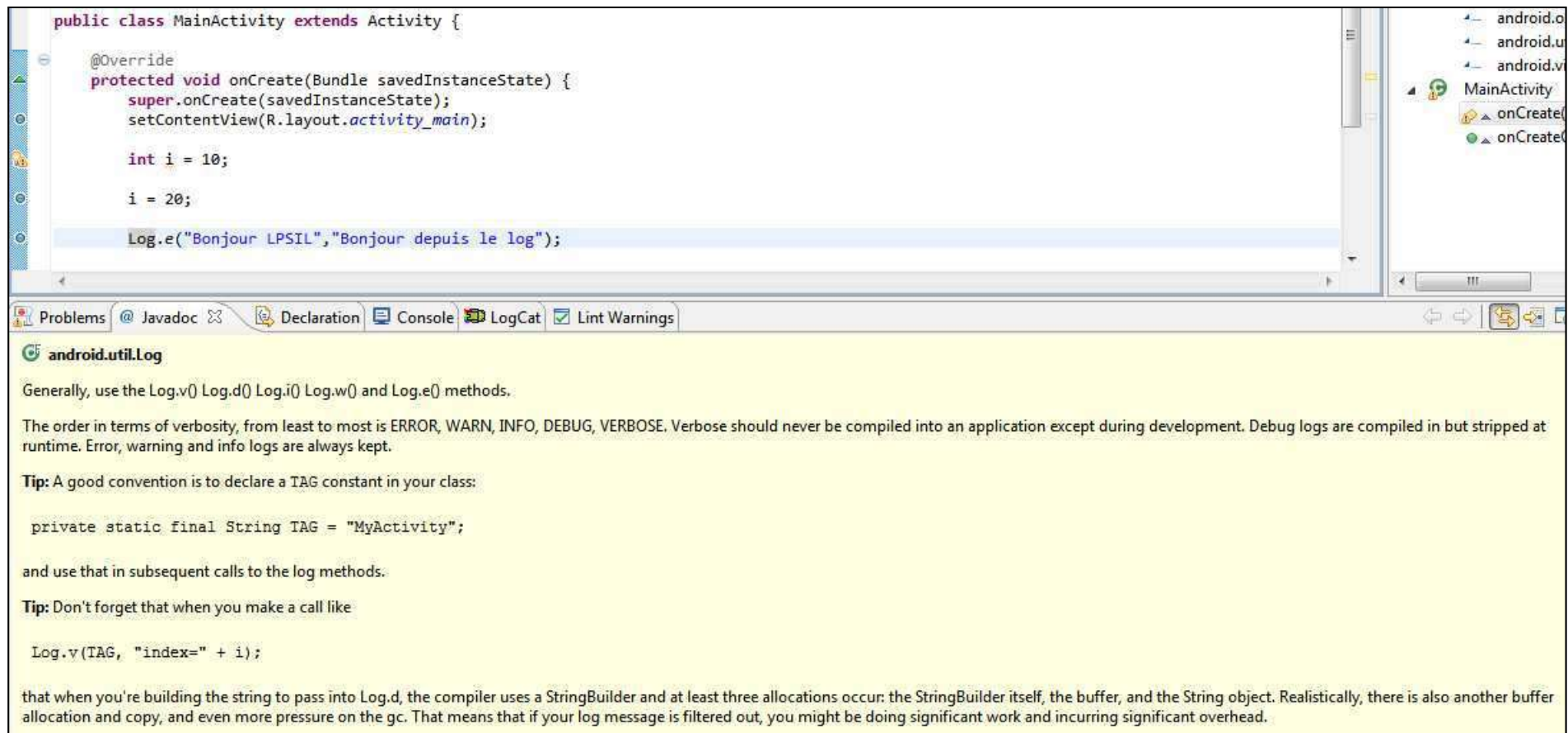
The dynamic help popup displays the following information for the `void com.ipsil.bonjour.MainActivity.onCreate(Bundle savedInstanceState)` method:

- Annotation:** `@Override`
- Overrides:** `onCreate(...)` in `Activity`
- Signature:** `protected void onCreate (Bundle savedInstanceState)`
- Added in:** `API level 1`
- Description:** Called when the activity is starting. This is where most initialization should go: calling `setContentView(int)` to inflate the activity's UI, using `findViewById(int)` to programmatically interact with widgets in the UI, calling `managedQuery(android.net.Uri, String[], String, String[], String)` to retrieve cursors for data being displayed, etc.
- Usage:** You can call `finish()` from within this function, in which case `onDestroy()` will be immediately called without any of the rest of the activity lifecycle (`onStart()`, `onResume()`, `onPause()`, etc) executing.
- Warning:** Derived classes must call through to the super class's implementation of this method. If they do not, an exception will be thrown.
- Parameters:**
 - `savedInstanceState`: If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in `onSaveInstanceState(Bundle)`. **Note: Otherwise it is null.**
- See Also:**
 - `onStart()`
 - `onSaveInstanceState(Bundle)`
 - `onRestoreInstanceState(Bundle)`
 - `onPostCreate(Bundle)`

Javadoc

75

- Menu Window > Show View > Javadoc



The screenshot shows an IDE window with a Java code editor and a Javadoc viewer. The code editor displays the following code:

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        int i = 10;  
  
        i = 20;  
  
        Log.e("Bonjour LPSIL", "Bonjour depuis le log");  
    }  
}
```

The Javadoc viewer shows the following content:

android.util.Log

Generally, use the `Log.v()` `Log.d()` `Log.i()` `Log.w()` and `Log.e()` methods.

The order in terms of verbosity, from least to most is ERROR, WARN, INFO, DEBUG, VERBOSE. Verbose should never be compiled into an application except during development. Debug logs are compiled in but stripped at runtime. Error, warning and info logs are always kept.

Tip: A good convention is to declare a TAG constant in your class:

```
private static final String TAG = "MyActivity";
```

and use that in subsequent calls to the log methods.

Tip: Don't forget that when you make a call like

```
Log.v(TAG, "index=" + i);
```

that when you're building the string to pass into `Log.d`, the compiler uses a `StringBuilder` and at least three allocations occur: the `StringBuilder` itself, the buffer, and the `String` object. Realistically, there is also another buffer allocation and copy, and even more pressure on the gc. That means that if your log message is filtered out, you might be doing significant work and incurring significant overhead.

Onglet Problems

76

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        int i = 10;  
  
        i = 20;  
  
        Log.e("Bonjour LPSIL", "Bonjour depuis le log");  
    }  
}
```

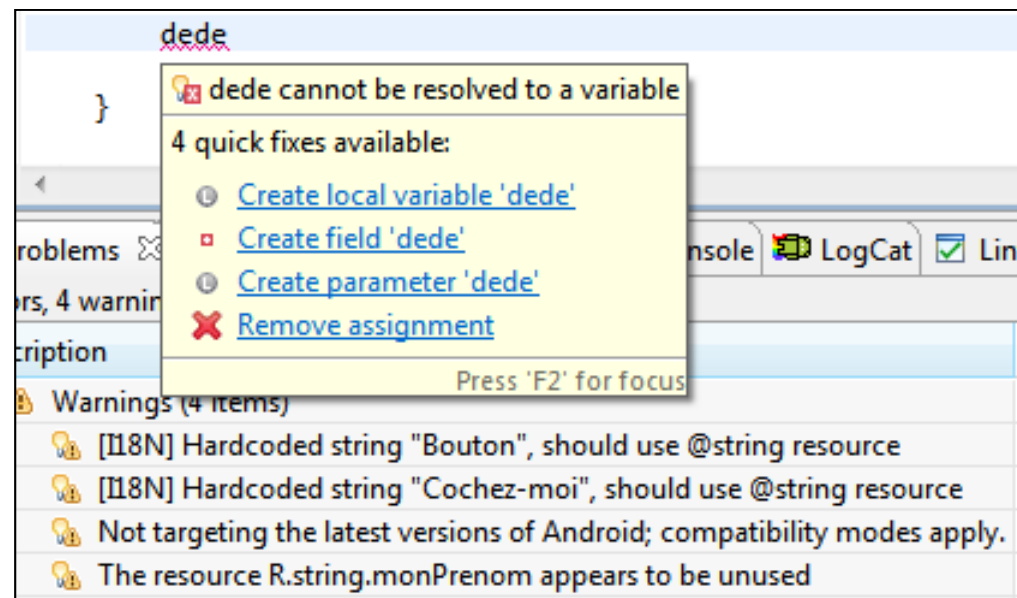
Problems | @ Javadoc | Declaration | Console | LogCat | Lint Warnings

0 errors, 5 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (5 items)				
[I18N] Hardcoded string "Bouton", should use @string resource	activity_main.xml	/Bonjour/res/layout	line 41	Android Lint Problem
[I18N] Hardcoded string "Cochez-moi", should use @string resource	activity_main.xml	/Bonjour/res/layout	line 31	Android Lint Problem
Not targeting the latest versions of Android; compatibility modes apply.	AndroidManifest.xml	/Bonjour	line 7	Android Lint Problem
The resource R.string.monPrenom appears to be unused	strings.xml	/Bonjour/res/values	line 7	Android Lint Problem
The value of the local variable i is not used	MainActivity.java	/Bonjour/src/com/lpsil/bonjour	line 15	Java Problem

En cas d'erreur...

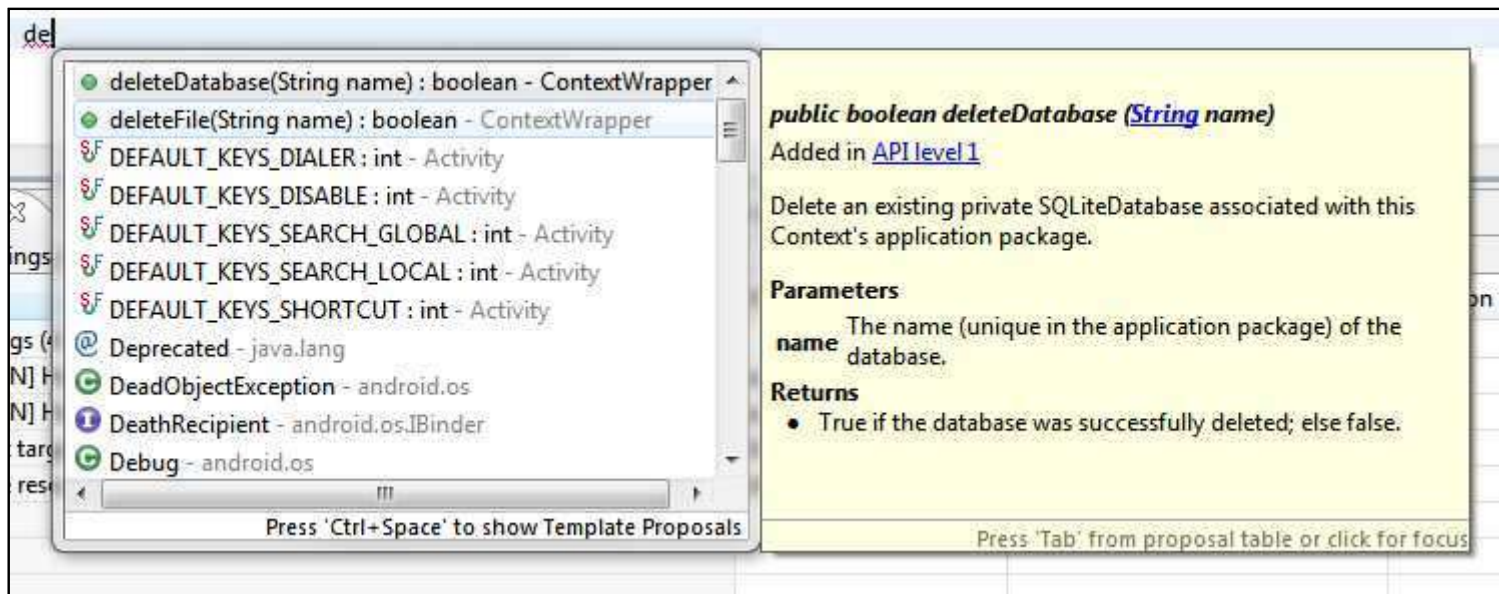
77



Auto-complétion

78

- **Ctrl + espace**



Auto-indenter

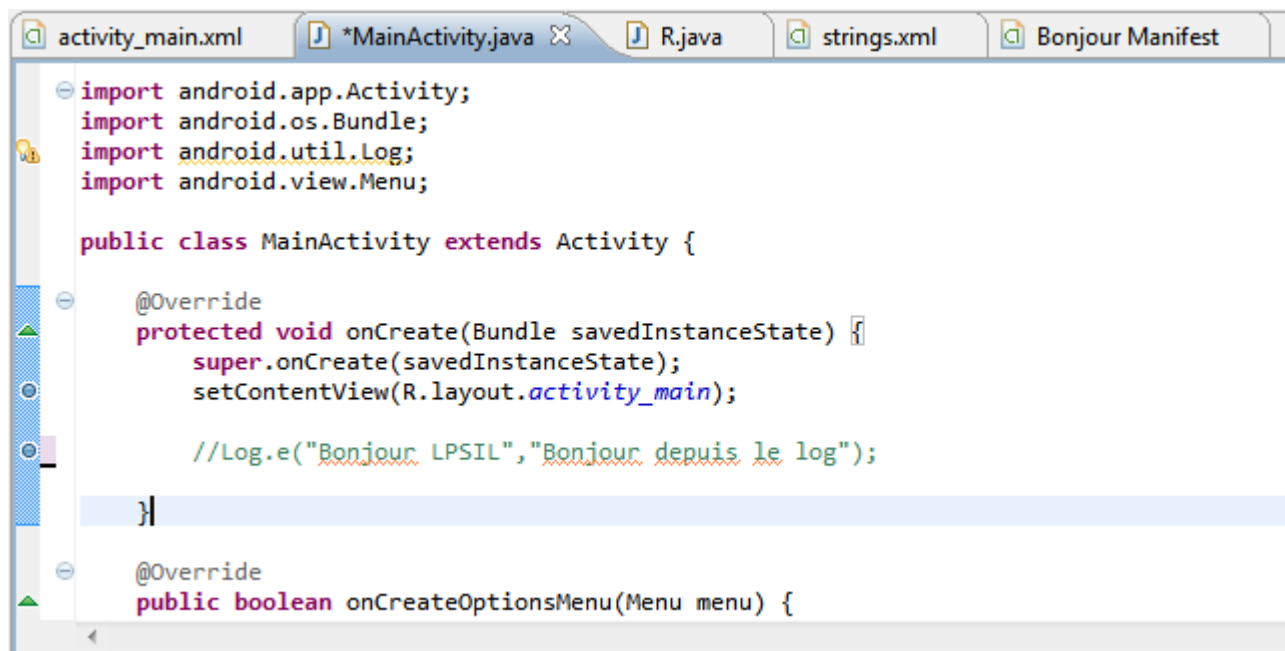
79

- `Ctrl + I`

Commentaires

80

- `//`
- `/* code commenté */`
- Raccourci clavier: sélectionner code puis `Ctrl + Shift + C`
 - Une 2e fois, cela supprime le commentaire



```
activity_main.xml | *MainActivity.java | R.java | strings.xml | Bonjour Manifest
import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;

public class MainActivity extends Activity {

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

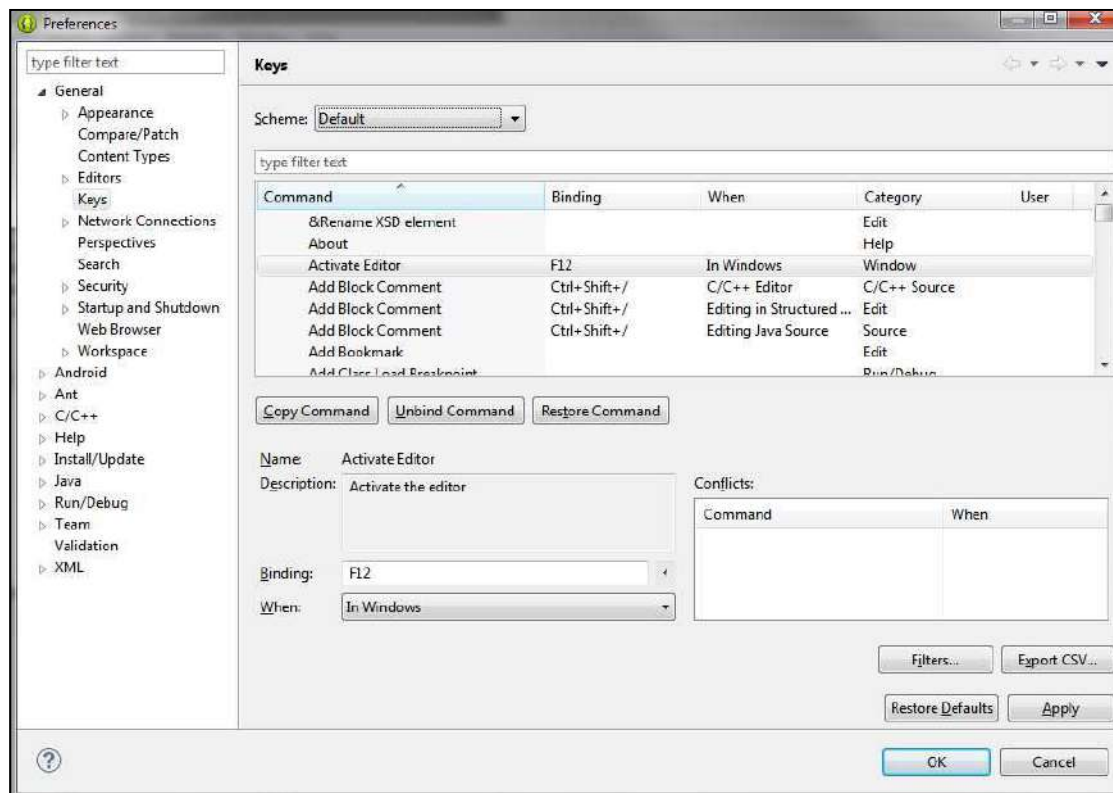
        //Log.e("Bonjour LPSIL", "Bonjour depuis le log");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```


Liste des raccourcis clavier

81

- **Ctrl + Shift + L**
 - 1^e fois: affichage de l'ensemble des raccourcis
 - 2^e fois: page des préférences



Activate Editor	F12
Add Block Comment	Ctrl+Shift+ /
Add Import	Ctrl+Shift+M
Add Javadoc Comment	Alt+Shift+J
All Instances	Ctrl+Shift+N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+C
Close	Ctrl+W
Close All	Ctrl+Shift+W
Collapse	Ctrl+Numpad_Subtract
Collapse All	Ctrl+Shift+Numpad_Divide
Content Assist	Ctrl+Space
Context Information	Ctrl+Shift+Space
Copy	Ctrl+C
Copy Lines	Ctrl+Alt+Down
Correct Indentation	Ctrl+I
Cut	Ctrl+X
Debug	F11
Debug Android Application	Alt+Shift+A, D
Debug Ant Build	Alt+Shift+D, Q
Debug JUnit Test	Alt+Shift+D, T
Debug Java Applet	Alt+Shift+D, A

Press "Ctrl+Shift+L" to open the preference page.

Semestre Printemps 2013