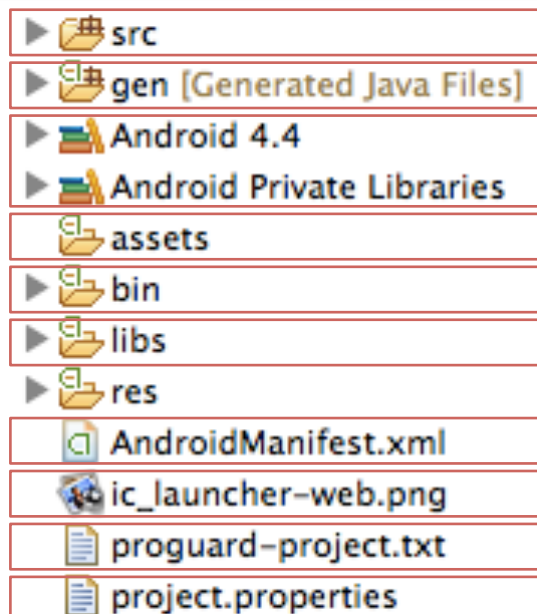


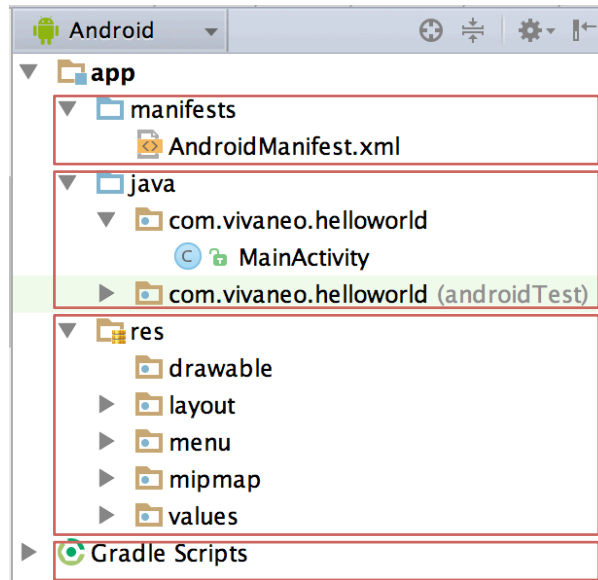
P2 : Développement Android

1. L'arborescence d'un projet
2. Le fichier AndroidManifest.xml
3. Les composants applicatifs
4. Cycle de vie d'une activité
5. Les ressources
6. Le débogueur
7. Mise à disposition d'une application

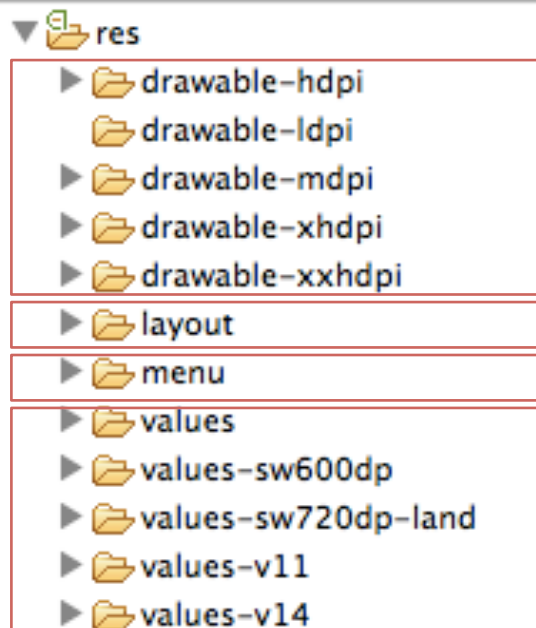
1 – L'arborescence d'un projet (plugin ADT)



1 – L'arborescence d'un projet (Studio)



1 – L'arborescence d'un projet (ressources)



1 – L'arborescence d'un projet (plugin ADT)

Un projet est décomposé de la manière suivante:

- src/ ou java/ : les sources Java du projet
- libs/: bibliothèques tierces
- res/drawable: ressources images
- res/layout: description des vues en XML
- res/values: chaînes de caractères et dimensions
- gen/: ressources autogénérées par ADT
- assets/: ressources brutes
- bin/classes: les classes compilées en .class
- bin/classes.dex: exécutable pour la JVM Dalvik
- bin/myapp.apk: application empaquetée avec ses ressources et prête pour le déploiement

2 – Le fichier AndroidManifest.xml

Contient tous les éléments de l'application :

- le nom du package de l'application
- l'icône, le nom de l'application et son thème
- le support du débogage (android:debuggable)
- la liste des composants (activité, service ...)
- les versions Android compatibles
- Les résolutions ou des tailles d'écran compatibles
- etc ...

2 – Le fichier AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.vivaneo.helloworld"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

2 – Le fichier AndroidManifest.xml

Versions compatibles :

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="18" />
```

Ecran supportés (optionnel) :

```
<supports-screens
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:xlargeScreens="true" />
```

2 – Le fichier AndroidManifest.xml

Application :

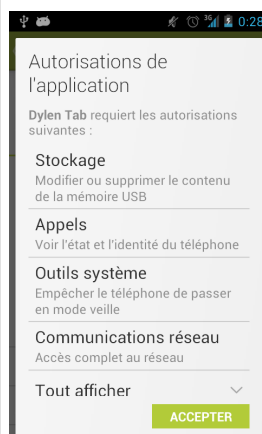
```
<application
    android:allowBackup="true"           // sauvegarde sur compte Google
    android:icon="@drawable/ic_launcher" // icône de l'application
    android:label="@string/app_name"     // nom de l'application
    android:theme="@style/SGTheme"       // theme de l'application
    android:hardwareAccelerated="true"   // accélération matérielle
                                         // (activé par défaut en version >= 14)
>
</application>
```

Activité :

```
<activity
    android:name=".ui.home.HomeActivity" // chemin de la source Java
    android:screenOrientation="portrait" // forcer le mode portrait
    android:noHistory="true"            // retirer l'écran de l'historique
    android:theme="@style/Theme.SGHome" // thème de l'activité
>
</activity>
```

2 – Le fichier AndroidManifest.xml

Exemple de permissions :



```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
```

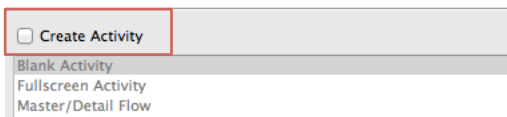
2 – Le fichier AndroidManifest.xml

TP « AndroidManifest » :

- Créer un nouveau projet sans créer d'activité (en décochant « Create Activity »).

Create Activity

Select whether to create an activity, and if so, what kind of activity.



☐ Create Activity

Blank Activity
Fullscreen Activity
Master/Detail Flow

- Ajouter une seconde activité « MainActivity.java »
- Ajouter les versions compatibles « 4.1 et supérieur »
- Ajouter les permissions « Internet », « état réseau »

3 – Les composants applicatifs

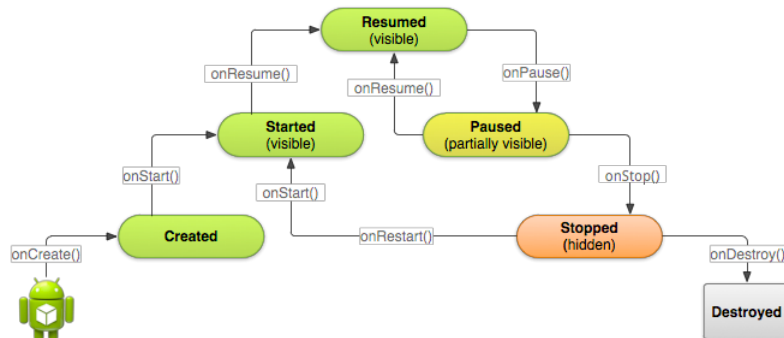
Une application peut être constituée de plusieurs composants :

- activity (bloc de base pour afficher des vues)
- service (exécuté en tâche de fond)
- content provider (permet de gérer et partager des informations au sein d'une application ou entre applications)
- widget (composant graphique affiché sur le bureau Android)

4 – Cycle de vie d'une activité

Les applications réagissent à différents états (démarrage, pause, reprise, arrêt ...).

Le système peut néanmoins décider de mettre en pause ou arrêter un processus (chaque application fonctionne dans son propre processus).



4 – Cycle de vie d'une activité

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    protected void onResume() {  
        // TODO Auto-generated method stub  
        super.onResume();  
    }  
  
    @Override  
    protected void onRestart() {  
        // TODO Auto-generated method stub  
        super.onRestart();  
    }  
  
    @Override  
    protected void onStart() {  
        // TODO Auto-generated method stub  
        super.onStart();  
    }  
  
    @Override  
    protected void onStop() {  
        // TODO Auto-generated method stub  
        super.onStop();  
    }  
  
    @Override  
    protected void onDestroy() {  
        // TODO Auto-generated method stub  
        super.onDestroy();  
    }  
}
```

4 – Cycle de vie d’une activité

TP « Etats » :

- Créer un nouveau projet avec une « Empty Activity » (ou utiliser le projet actuel).
- Ajouter des logs sur chaque état.

```
Log.i("state", "");
```
- Vérifier les logs avec le « LogCat » pour voir les interactions.

5 – Les ressources

Les ressources sont utilisées dans le code au travers de la classe statique « **R.** » (rechargé automatiquement après chaque changement dans le projet).

Les types de ressource (répertoire « res »)

- res/values: fichiers XML convertis en différents types de ressource (exemple: « strings.xml »).
- res/drawable: images .png ou .jpg qui sont converties en bitmap ou .9.png qui sont converties en « 9-patches » (images ajustables).
- res/layout: fichiers XML convertis en mise en page d’écran (gabarit).
- res/anim: fichiers XML convertis en animation.
- res/xml: fichiers XML qui peuvent être lus et convertis à l’exécution par la méthode « ressources.getXML ».
- res/raw: fichiers à ajouter à l’application (voir aussi assets).

5 – Les ressources : res/values/strings.xml

Il est possible de déclarer des chaînes de caractère, des dimensions, des couleurs [...] comme valeurs simples.

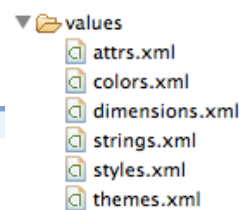
Ces valeurs sont déclarées dans des fichiers XML.

Exemple de fichier « strings.xml » :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">ATest</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>

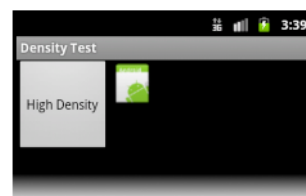
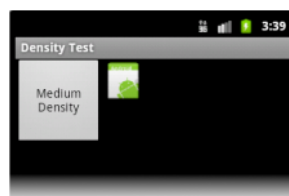
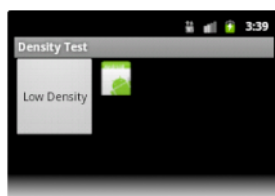
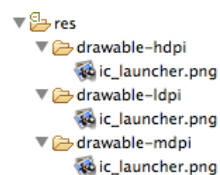
</resources>
```



5 – Les ressources : res/drawable-*/img.png

Il y a différents dossiers « drawable » pour chaque résolution (ldpi, mdpi, hdpi etc ...).

Exemple :



5 – Les ressources : res/drawable/img.xml

Les différents états d'une image sont gérés avec un fichier XML.

Exemple « res/drawable/bouton_fleche.xml » :

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/bouton_fleche_on" android:state_pressed="true"/>
  <item android:drawable="@drawable/bouton_fleche_on" android:state_focused="true"/>
  <item android:drawable="@drawable/bouton_fleche_off"/>
</selector>
```



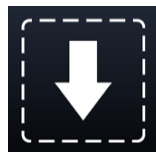
bouton_fleche_off.png



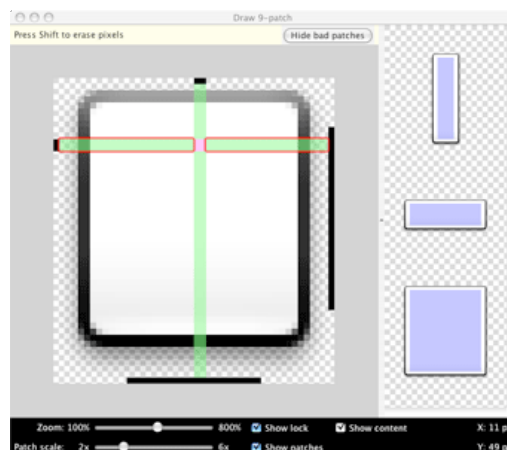
bouton_fleche_on.png

5 – Les ressources : res/drawable/img.9.png

Exemple « 9-patch » :



Outils « draw9patch.bat »
disponibles dans le répertoire
tools du sdk.



5 – Les ressources : res/layout/activity.xml

Exemple d'un layout « activity.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"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world" />

</RelativeLayout>
```

5 – Les ressources : res/anim/

Exemple d'une animation « fade.xml »:

```
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:fromAlpha="0.0" android:toAlpha="1.0"
    android:duration="@android:integer/config_longAnimTime" />
```

Exemple d'une animation « slide_left.xml »:

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <translate android:fromXDelta="100%p" android:toXDelta="0"
        android:duration="@android:integer/config_shortAnimTime" />
</set>
```

5 – Les ressources : res/xml/preference.xml

Exemple d'une page de préférence :

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android">

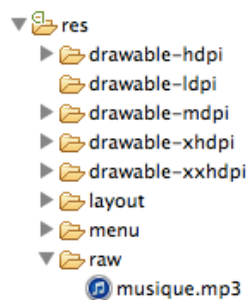
    <!-- My custom preference type. This just replaces the actual widget
         portion of the preference, if the whole preference wanted to be
         replaced we would use the layout attribute instead of the widgetLayout
         attribute. -->
    <com.example.android.apis.preference.MyPreference
        android:key="my_preference"
        android:title="@string/title_my_preference"
        android:summary="@string/summary_my_preference"
        android:defaultValue="100" />

    <CheckBoxPreference
        android:key="advanced_checkbox_preference"
        android:title="@string/title_advanced_toggle_preference"
        android:summaryOn="@string/summary_on_advanced_toggle_preference"
        android:summaryOff="@string/summary_off_advanced_toggle_preference" />

</PreferenceScreen>
```

5 – Les ressources : res/raw/musique.mp3

Exemple de lecture d'un fichier audio « .mp3 » :



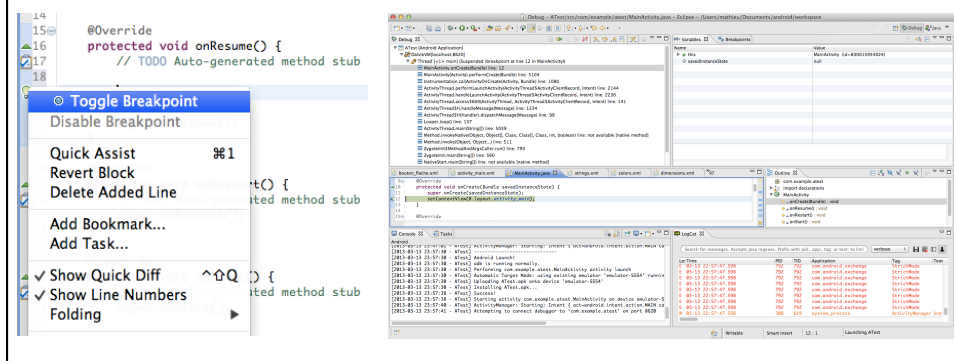
Code dans l'activité :

```
MediaPlayer mMediaPlayer = MediaPlayer.create(this, R.raw.musique);
mMediaPlayer.start();
```

6 – Le débogueur (Eclipse)

Il est nécessaire d'ajouter des points d'arrêt dans votre code (clique droit sur la barre latérale).

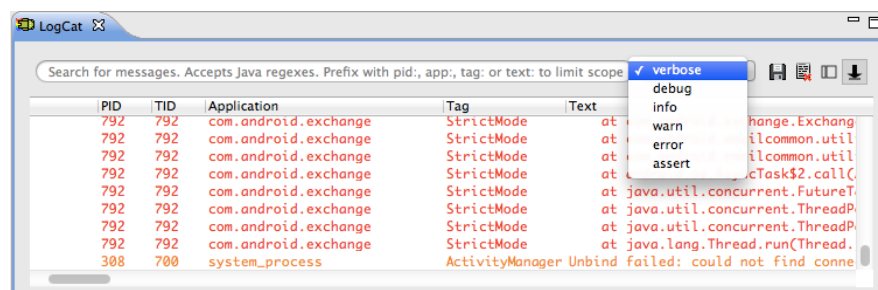
Au lieu de cliquer sur le bouton d'exécution, cliquez sur le bouton de débogage 



6 – Le débogueur (Eclipse)

La perspective DDMS fournit également le LogCat qui permet d'afficher tout le journal des logs.

Il est également possible de filtrer les informations avec le champs de recherche ou par type.



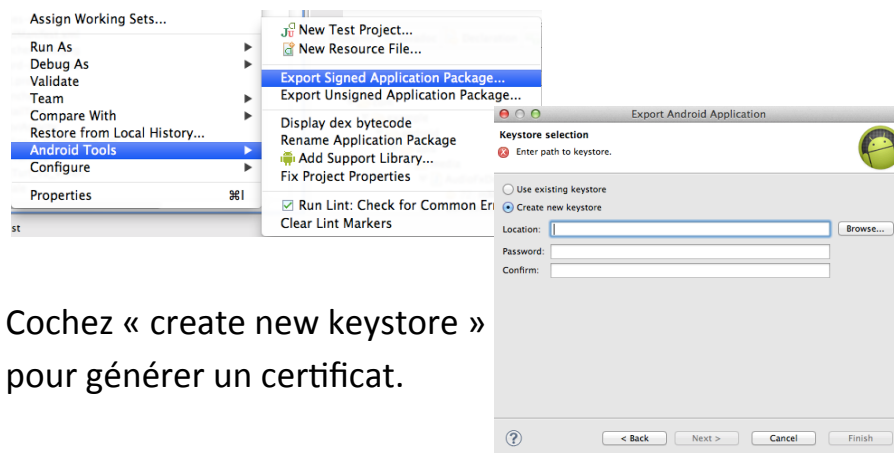
6 – Le débogueur

TP « Débogueur » :

- Déclarer un entier
- Incrémenter la valeur de l'entier sur les méthodes « onCreate » et « onStart »
- Lancer l'application en mode « Debug »
- Ajouter des « Logs » et filtrer les résultats dans le « Logcats »

7 – Mise à disposition d'une application

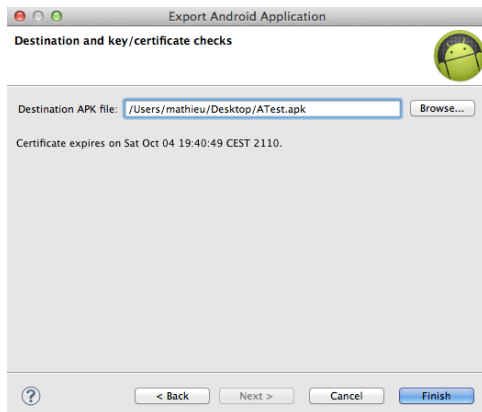
Pour déployer votre application il faut la signer avec un certificat.



Cochez « create new keystore »
pour générer un certificat.

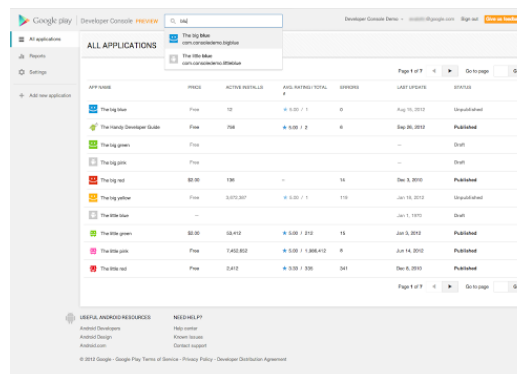
7 – Mise à disposition d'une application

Une fois terminé, vous pouvez transmettre le fichier APK sur le Play Store Android (optionnel).



7 – Mise à disposition d'une application

Pour publier l'application sur le Play Store, il faut vous enregistrer sur <https://play.google.com/apps/publish>. Après avoir validé les conditions générales, vous devrez vous acquitter de 25\$ pour ensuite accéder à l'interface de publication.



7 – Mise à disposition d'une application

TP « Export » :

- Exporter une application en créant un nouveau certificat