



ANDROID

19/04/2019

L.Freund, laurent.freund@wf3.fr

Plan des séances

Hello World (pour changer !!!)

Les ressources

L'IHM

Internet

Le stockage des données

La géolocalisation

...

Un peu de culture

3



- 2008 : G1 => Mobiles, tablettes, TV...
- Pile logicielle Open-Source supportée par l'Open Handset Alliance
http://www.openhandsetalliance.com/oha_members.html
- Système d'exploitation open source
- Applications natives : browser, mail, contacts...
- Environnement de développement : Android Studio
- Développement de surcouches par les constructeurs (ex HTC), opérateurs...
- Développement gratuit

Le marché il y a 10-6 ans

4

LE MOBILE EN FRANCE EN 2013

55% DES INTERNAUTES CONSULTENT L'INTERNET MOBILE QUOTIDIENNEMENT

70,5 MILLIONS DE FORAITS MOBILES
Pour une population totale de 65 millions, soit un taux de pénétration de 108%

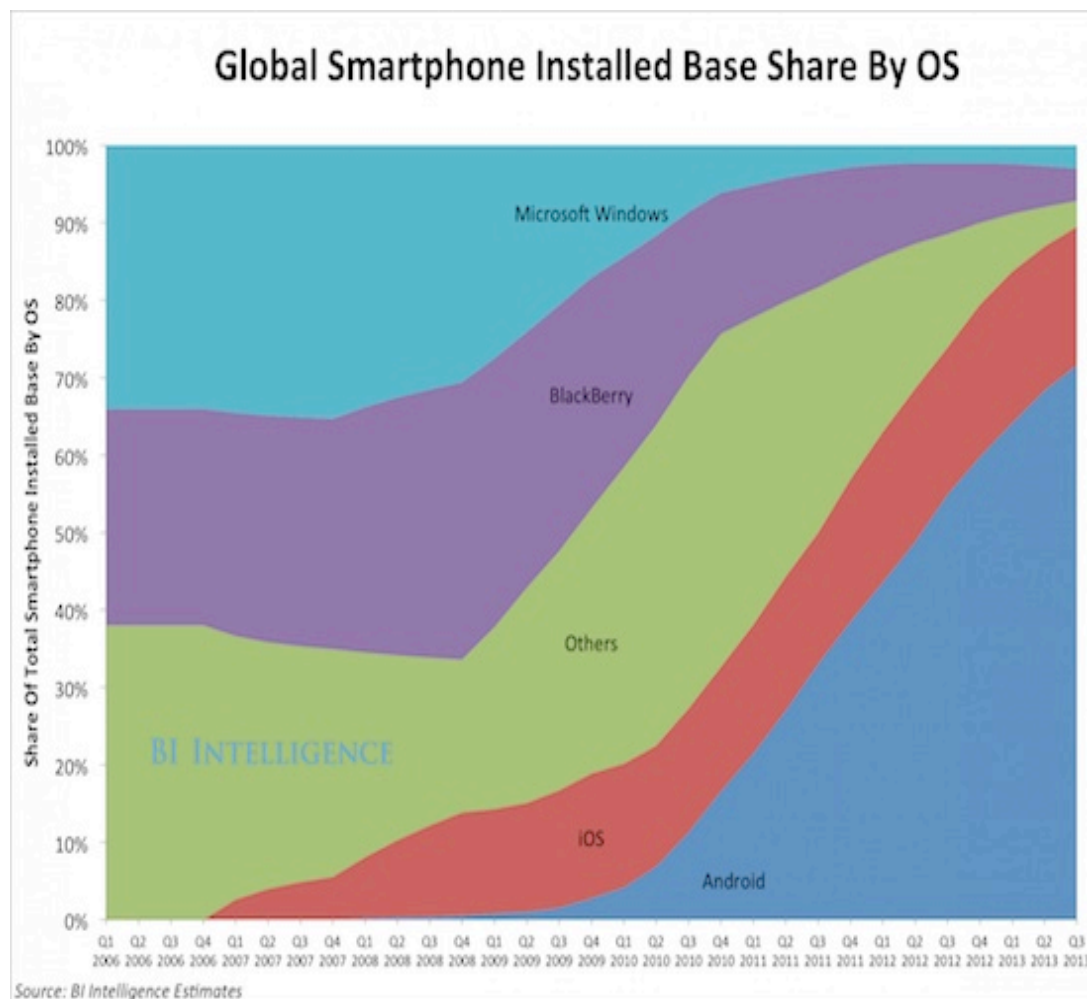
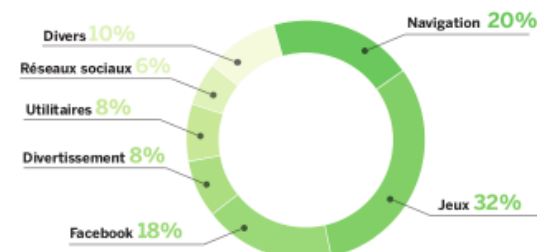
41% DES TÉLÉPHONES MOBILES SONT DES SMARTPHONES

15% DES FRANÇAIS ONT UNE TABLETTE

RÉPARTITION DES MODÈLES DE TÉLÉPHONE



TEMPS PASSÉ SUR UN APPAREIL CONNECTÉ



Le marché ces 6 dernières années

5

Ventes de smartphone en France (janvier 2015)

Android iOS Windows RIM Autres

Android; 65,30%



iOS; 20,20%

Microsoft
Windows; 13,00%



RIM; 1,10%

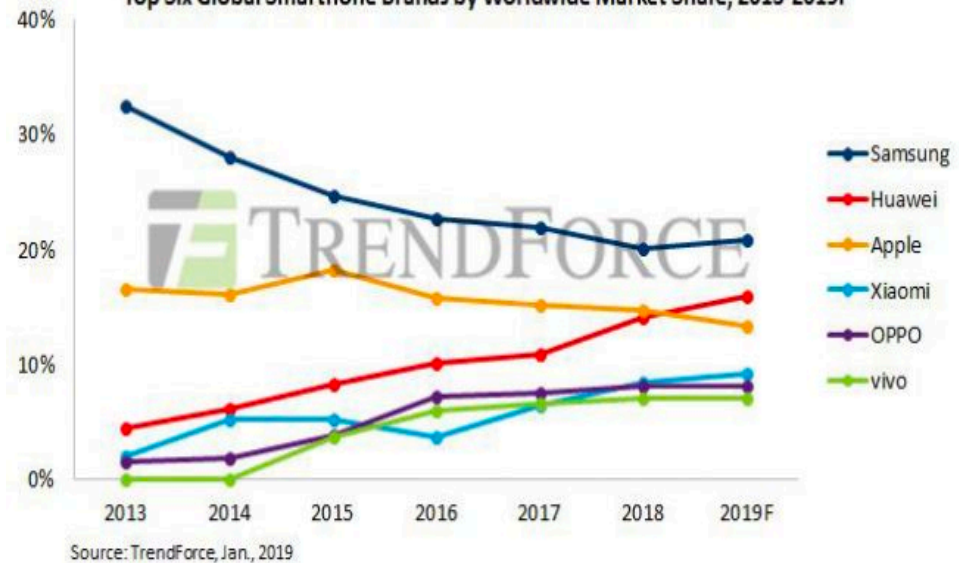
Autres; 0,30%

A propos des statistiques Kantar WorldPanel: Part de marché sur les ventes de smartphone en France. (Panel de consommateur)

janv-15

Infographie : EcoConscient

Top Six Global Smartphone Brands by Worldwide Market Share, 2013-2019F





APIs : accès complet au matériel

6

- ❑ Réseau GSM, EDGE, 3G, 4G, LTE...
- ❑ Géolocalisation : Google Map, géocodage
- ❑ Communication : Wifi, bluetooth, NFC
- ❑ Capteurs : alimentation, champ magnétique
- ❑ IHM : Multimédia, navigateur HTML5, Widgets, fonds d'écran
- ❑ Graphisme : OpenGL ES 2.0, 2D, 3D
- ❑ Services arrière plan : tâches de fond, notifications
- ❑ Data : Bases de données tte, fournisseurs de contenu

les autres n'ont pas...



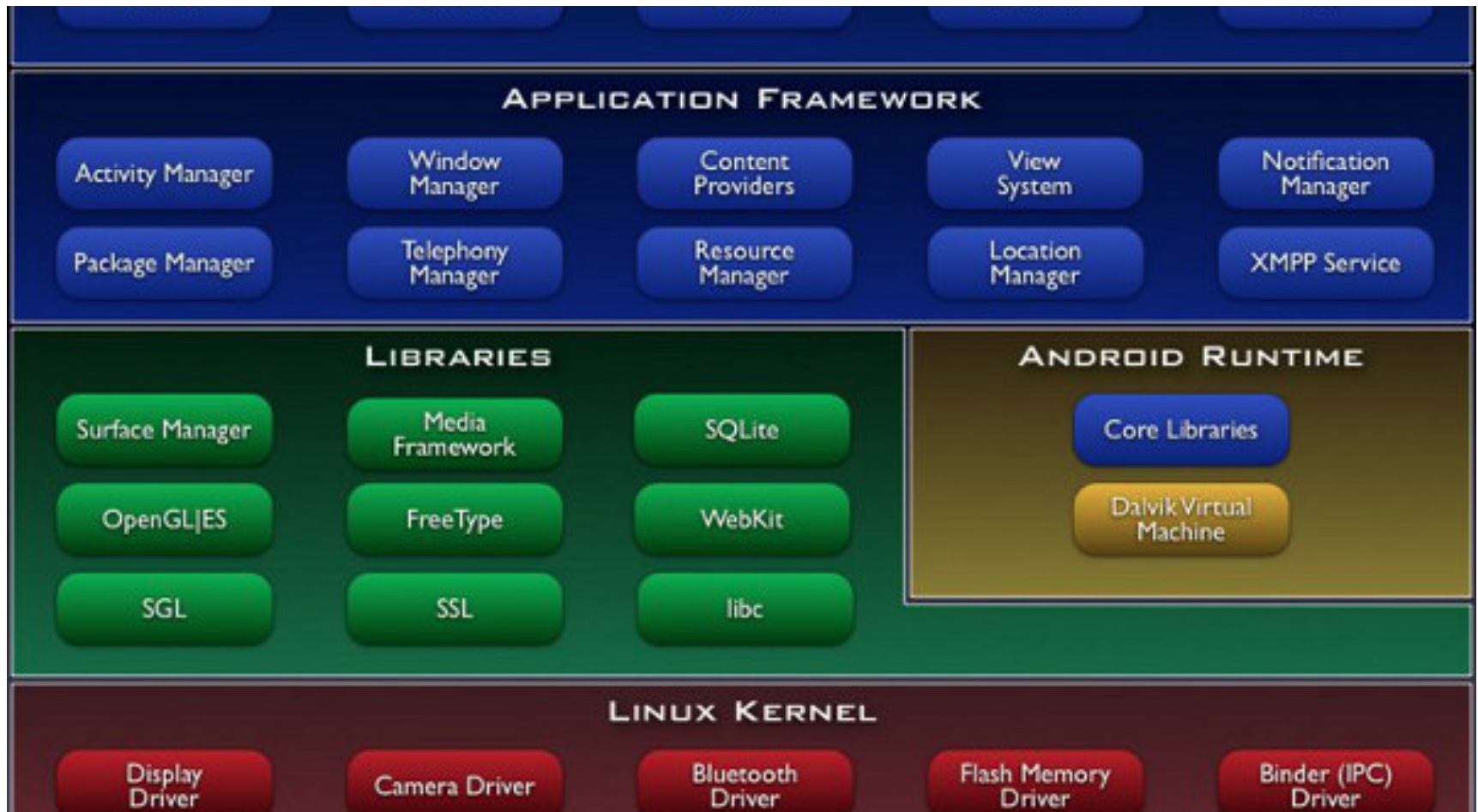
7

- ❑ Google Maps
- ❑ Services d'arrières plan
- ❑ Données partagées et communication interprocessus
- ❑ Applications natives = applications tierces
- ❑ Android Beam
- ❑ Widget, WallPaper, boite de recherche



Machines virtuelle

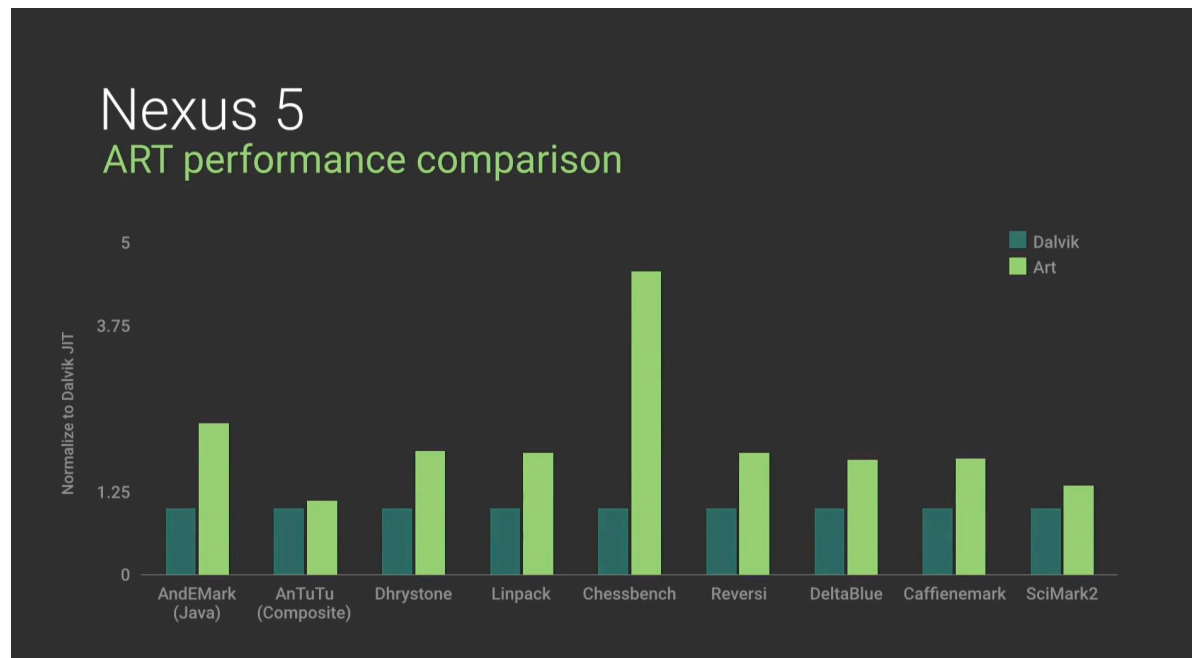
8



Machines virtuelle Dalvik -> ART

9

- **Dalvik compile à la volée les applications, lors de leur lancement, alors que ART compile les applications lors de leur installation.**
- La durée d'installation sera donc plus longue mais les avantages sur le long terme ne sont pas négligeables.



10

Hello World

Prise en main d'Android Studio

Hello World !!

```
>HELLO WORLD  
>_
```

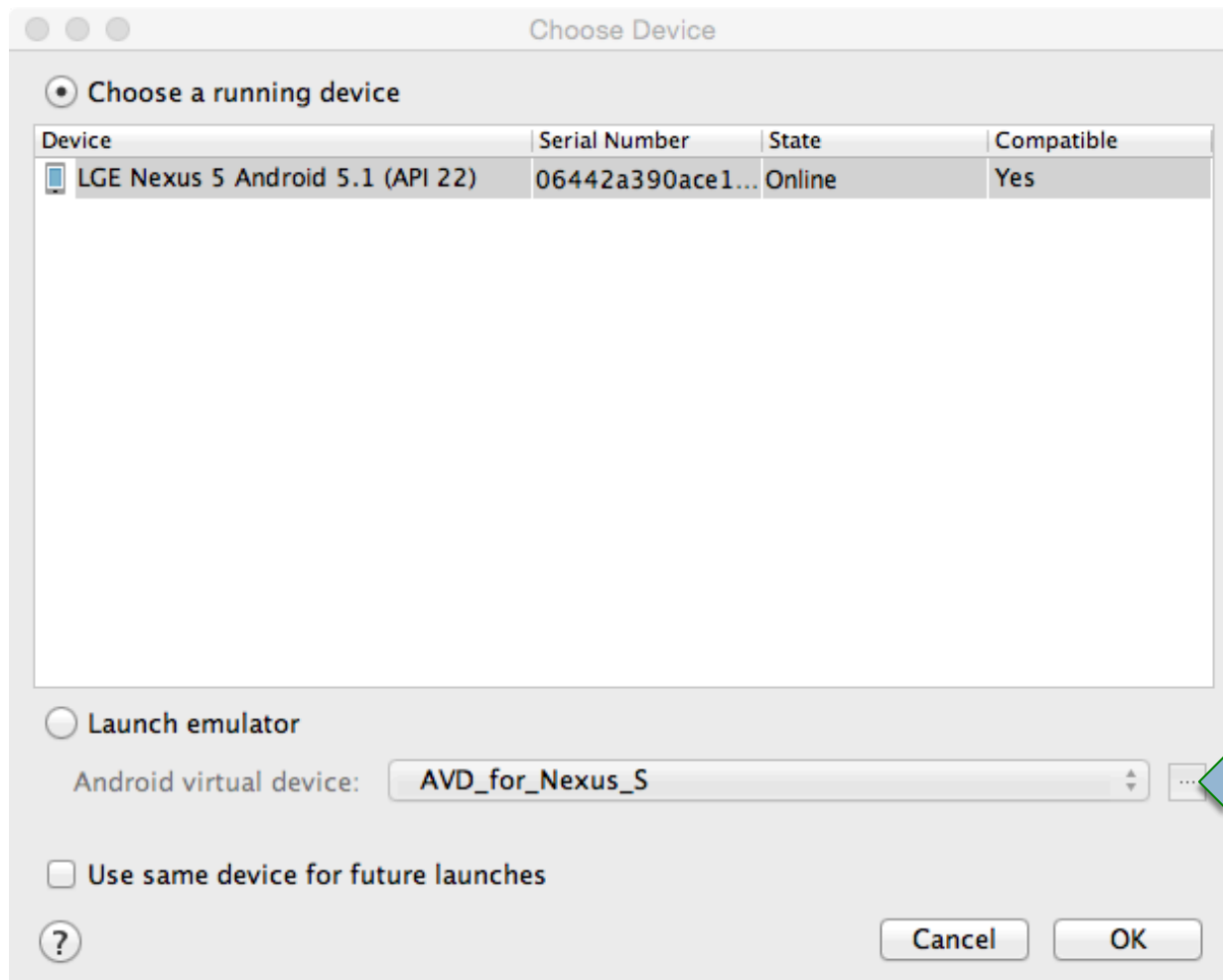
11

- Lancer Android Studio
- *New->New Project*
- *Choisir Empty Activity [Default]*
- Application Name : Hello
- Company Domain: formation.fr
- Phone & Tablet [x]
- Min SDK: API 19
 - ▣ <http://developer.android.com/about/dashboards/index.html>
- Empty Activity
- Activity Name : HelloDolly
- Title : Hello Dolly

Execution : AVD ou Mobile



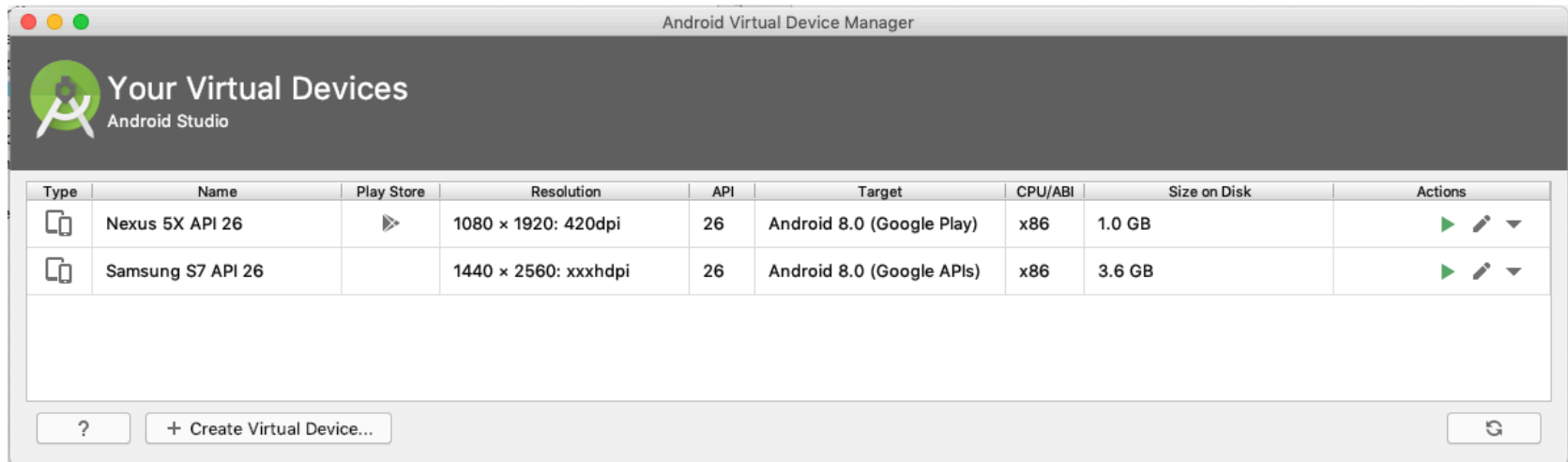
12



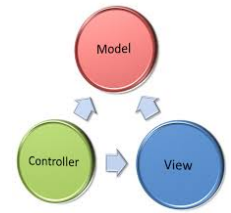
AVD : Création de l'émulateur



13

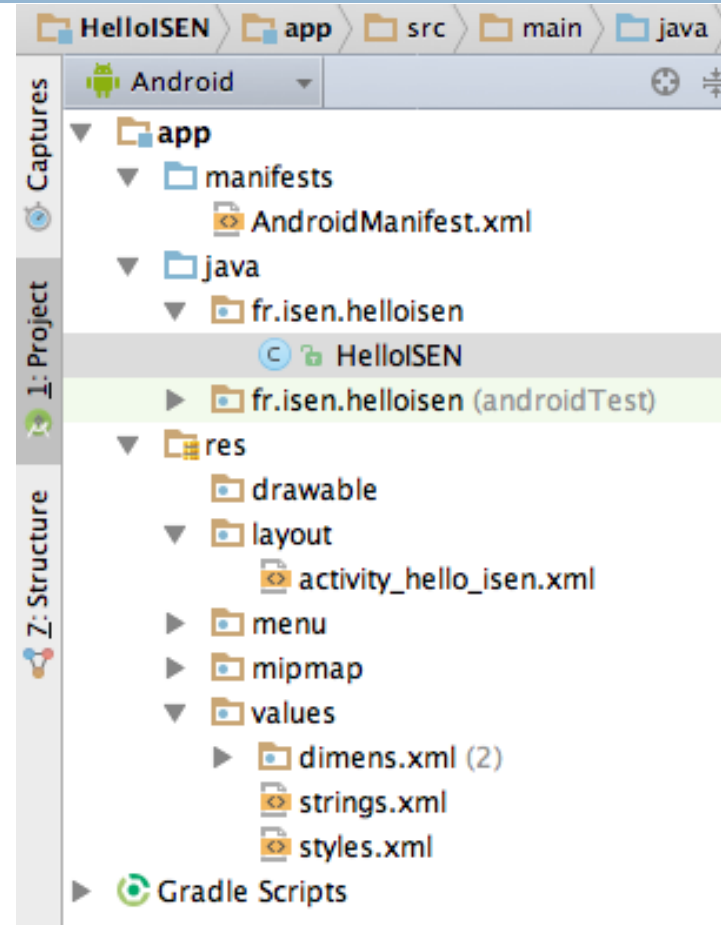


Décorticage du code (MVC)

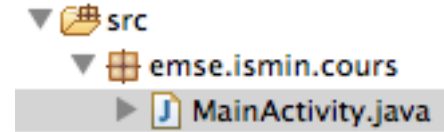


14

- Le code : Java
- L'IHM : xml
 - ▣ *Activity_hello_isen*
- Les définitions des valeurs : xml
 - ▣ *Strings.xml*



MainActivity.java



15

```
package emse.ismin.cours;
import android.os.Bundle; ..
/**
 * Hello World
 * @author freund
 */
public class MainActivity extends Activity {

    /**
     * Display the main view with the Hello world txt
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // affichage vue
    }
}
```

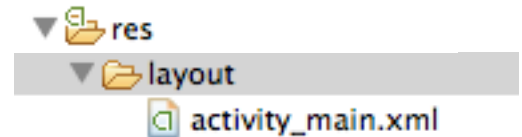
/** Commentaires */ : javadoc

@Override : vérif compilateur

R.layout.activity_main: ihm

activity_main.xml

16



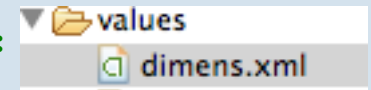
```
<?xml version="1.0" encoding="utf-8"?><android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/></android.support.constraint.ConstraintLayout>
```

<RelativeLayout ... > : layoutManager

"match_parent" / "wrap_content"

padding : espacement

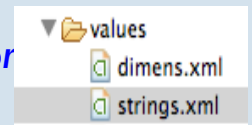
@dimen:



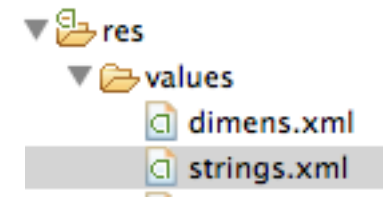
".MainActivity" ← IHM (Themes...)

<TextView ... > : display text

"@string/hello_world"



strings.xml



17

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

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

</resources>
```

"app_name": nom de l'appli

"hello_world": Le texte passionnant

Les tools

Android Virtual Device

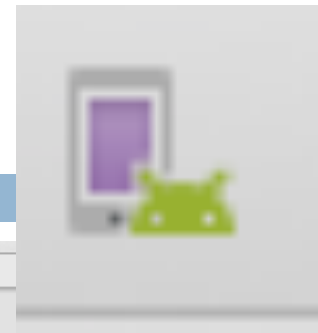
Android SDK Manager

Android Device Monitor DDMS

-> LogCat

Android Virtual Device

19



Android Virtual Devices Device Definitions

List of existing Android Virtual Devices located at /Users/freund/.android/avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ Andros1.6	Android 1.6	1.6	4	ARM (armeabi)
✓ AVDgooMap	Google APIs (Google Inc.)	2.1	7	ARM (armeabi)
✓ NexusS	Android 2.3.1	2.3.1	9	ARM (armeabi)
✓ Andros2.3.3	Android 2.3.3	2.3.3	10	ARM (armeabi)
✓ AndrosGoogAPI10	Google APIs (Google Inc.)	2.3.3	10	ARM (armeabi)

New...
Edit...
Delete...
Repair...
Details...
Start...
Refresh

AVD details

Name: NexusS
CPU/ABI: ARM (armeabi)
Path: /Users/freund/.android/avd/NexusS.avd
Target: Android 2.3.1 (API level 9)
Skin: 480x800

hw.dPad: no
hw.lcd.density: 240
avd.ini.encoding: ISO-8859-1
hw.device.hash: 499058361
hw.camera.back: none
disk.dataPartition.size: 200M
skin.dynamic: yes
hw.keyboard: yes
hw.ramSize: 343

✓ A valid Android Virtual Device. 🗑️ A repairable Android Virtual Device.
✗ An Android Virtual Device that failed to load. Click 'Details'.

Project Structure



20

The screenshot shows the Android Studio interface with the Project Structure dialog open. The 'Properties' tab is active, showing configuration for the 'app' module. The left sidebar lists various developer services and modules, with 'app' selected under the 'Modules' section. The main panel displays the following settings:

Property	Value
Compile Sdk Version	API 23: Android 6.0 (Marshmallow)
Build Tools Version	23.0.2
Library Repository	
Ignore Assets Pattern	
Incremental Dex	
Source Compatibility	
Target Compatibility	

Android SDK Manager

21



- Appearance
- Menus and Toolbars
- ▼ System Settings
 - Passwords
 - HTTP Proxy
 - Updates
 - Usage Statistics
- Android SDK**
- Notifications
- Quick Lists
- Path Variables
- Keymap
- ▶ Editor
- Plugins
- ▶ Build, Execution, Deployment
- ▶ Tools

Android SDK Location:

SDK Platforms SDK Tools SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
☐	Android SDK Build Tools		Update Available: 24.0.0 ...
☒	Android Auto API Simulators	1.0.0	Installed
☐	Android Auto Desktop Head Unit emulator	1.1	Not installed
☐	Android SDK Platform-Tools 23.1	23.1.0	Update Available: 24.0.0 ...
☒	Android SDK Tools 24.4.1	24.4.1	Update Available: 25.0.10
☒	Android Support Library, rev 23.2.1	23.2.1	Installed
☒	Documentation for Android SDK	1	Installed
☐	GPU Debugging tools	1.0.3	Not installed
☒	Google Play APK Expansion Library, rev 3	3.0.0	Installed
☒	Google Play Billing Library, rev 5	5.0.0	Installed
☒	Google Play Licensing Library, rev 2	2.0.0	Installed
☒	Google Play services for Froyo, rev 12	12.0.0	Installed
☒	Google Play services, rev 29	29.0.0	Installed
☒	Google Repository, rev 24	24.0.0	Installed
☒	Google Web Driver, rev 2	2.0.0	Installed
☒	Intel x86 Emulator Accelerator (HAXM installer), rev 6.0	6.0.1	Installed
☐	LLDB	2.0.2558144	Not installed
☒	Local Maven repository for Support Libraries, rev 28	28.0.0	Installed

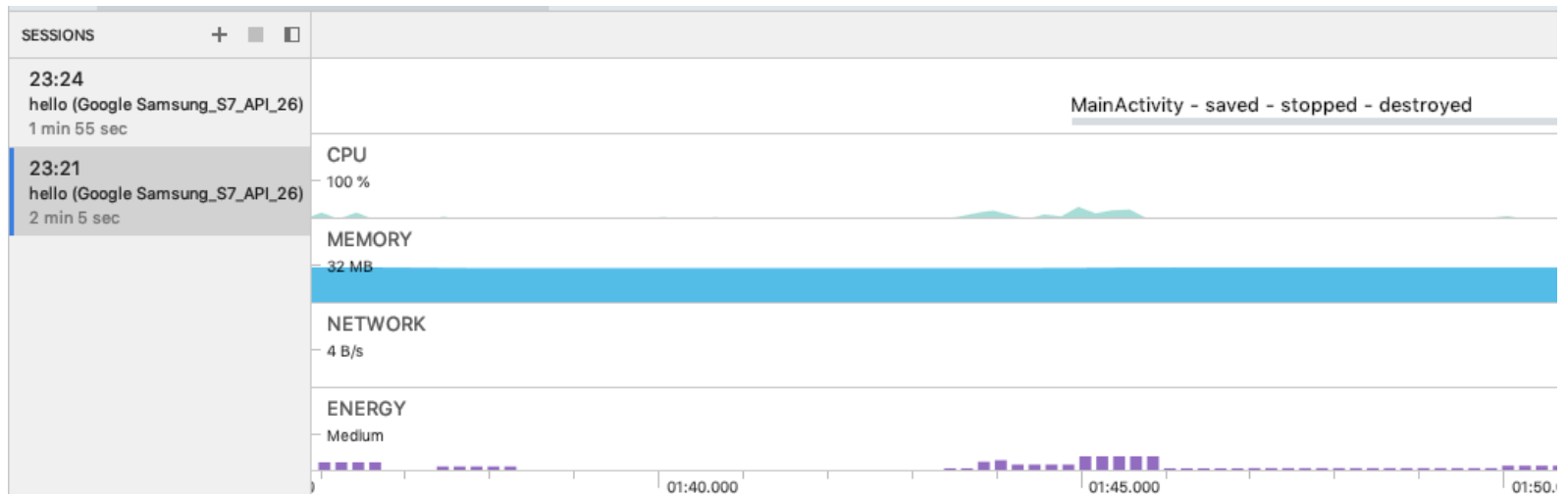
☐ Show Package Details

[Launch Standalone SDK Manager](#)

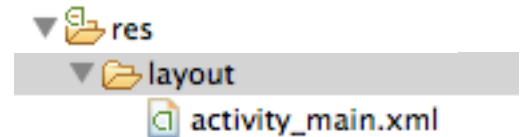


Profile App

22



Faisons nous la main



23

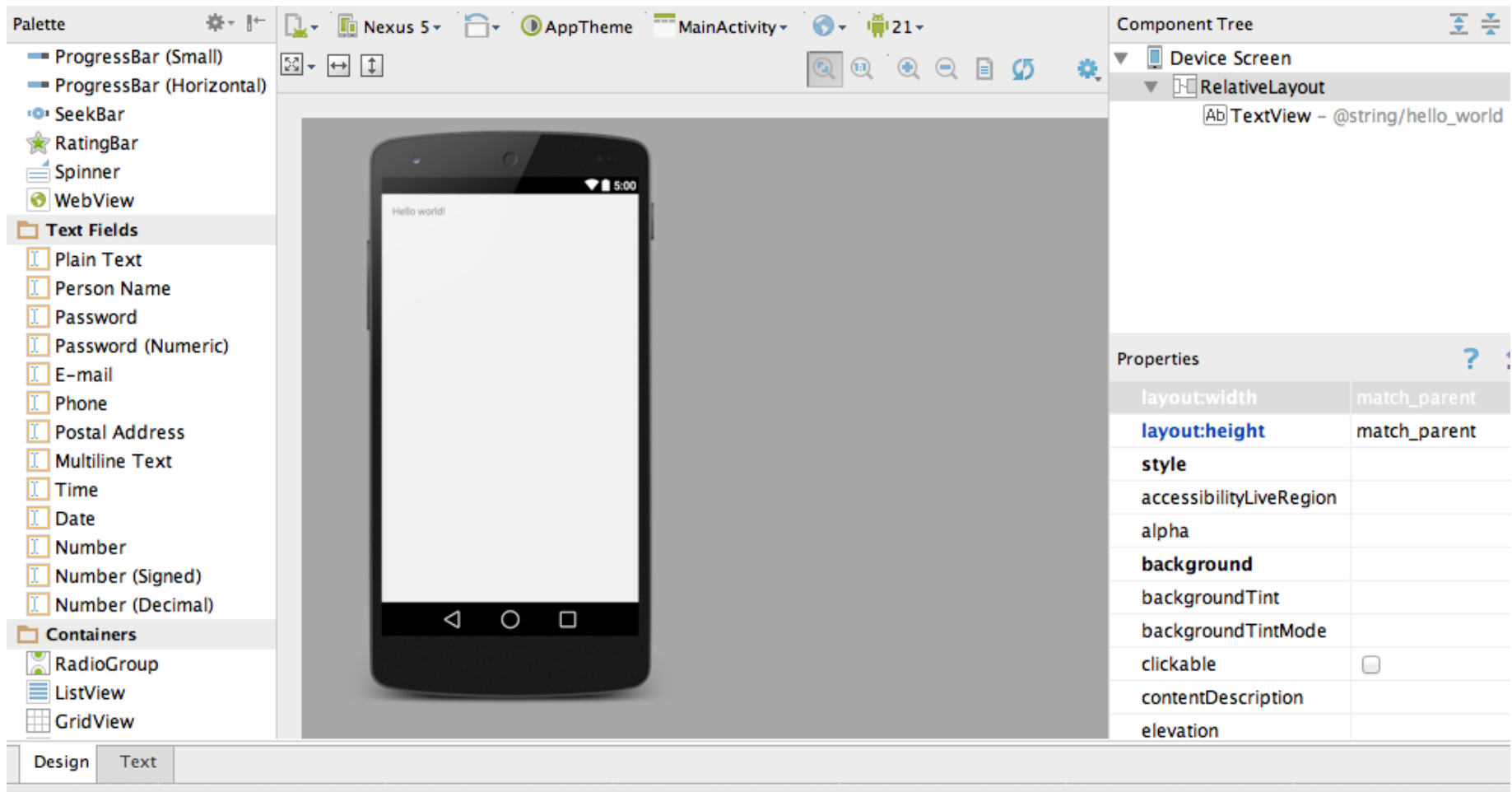
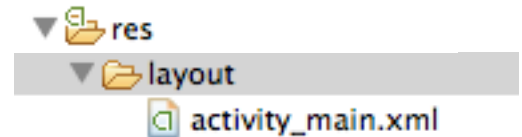
Ajout d'un champ de saisie et d'un bouton qui change le texte

-> nouvelles notions qui vont être apprises

- Modif de res/layout/*activity_main.xml*
- Ajout `<EditText .../>`, `<Button ...>`
- Les paramètres obligatoires
- Le placement dans un `ConstraintLayout`
- Les autres gestionnaires
- L'id d'un élément
- La gestion des events

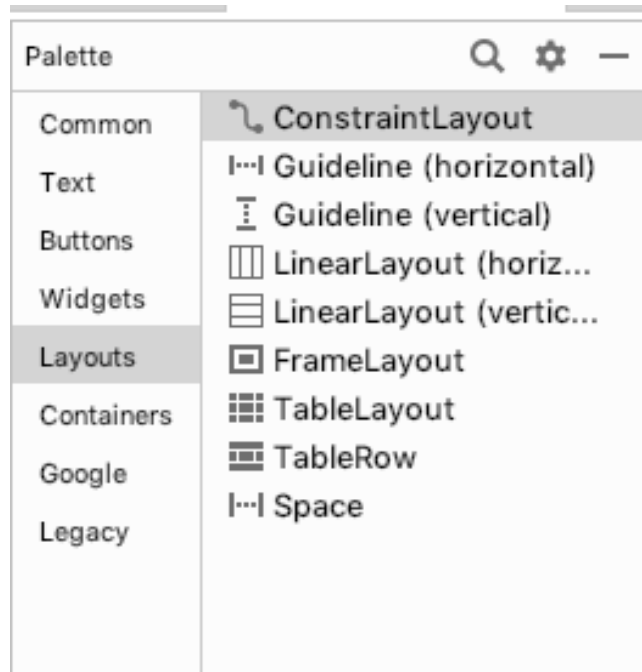
Design

24



Les Layout pour le placement

25



ConstraintLayout : référence depuis 2016

Notion de poids (LinearLayout), chargement rapide

<http://tutos-android-france.com/constraintlayout-partie-1/>

Placements des éléments

26

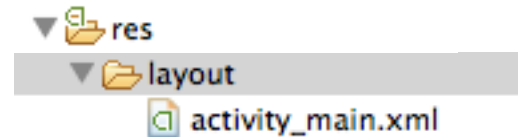
- Alignement avec le parent
- Alignement entre éléments (coté/bords)
- Alignement de la BaseLine : `text`
- Ajuster le Bias : `% (weight)`
- Les Chains : (selection puis click droit)
 - ▣ alignement sur une ligne
 - ▣ Sélection 2 élts, choix du type de la chaine (icone)
 - ▣ choix du weight (avec Odp comme width)

Placements des éléments

27

- Rajouter une guideline
 - ▣ Permet de faire des lignes de positionnement pour plusieurs éléments par exemple
- Rajouter une barrière

activity_main.xml



28

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />
```

<EditText

```
    android:id="@+id/editText1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/textView1"
    android:inputType="text" />
```

<Button

```
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_below="@id/editText1"
    android:text="@string/ok" />
```

- Obligatoires : placement
 - ▣ android:layout_width="*match_parent*"
 - ▣ android:layout_height="*wrap_content*"
 - ▣ android:inputType="*text*" (*voir graphical layout*)
- Identifiant
 - ▣ Définition : android:id="*@+id/editText1*"
 - ▣ *Ex. utilisation* : android:layout_below="*@id/editText1*"

1^{ère} solution : gestion inline de l'événement Button dans le .java

30

- Implémentation **inline** du `onClick`Listener

```
final static String TAG = "MainActivity" ;
```

```
...  
public boolean onCreate(Bundle savedInstanceState) {
```

```
...
```

```
// add button
```

```
Button but1 = (Button)findViewById(R.id.button1);
```

```
but1.setOnClickListener(new OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View V) {
```

```
        Log.d(TAG, "click...");
```

```
    }
```

```
});
```

```
return true ;
```

```
}
```

2nde solution : implémentation du listener du **Button** dans le .java

31

- Implémentation `onClickListener` par la classe

```
public class MainActivity extends Activity implements OnClickListener{
    ... onCreate {
        ...
        Button but1 = (Button)findViewById(R.id.button1);
        but1.setOnClickListener(this);
    }
    /** Manage buttons */
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.button1 :
                ((TextView)findViewById(R.id.textView1)).setText("You said: "
                    + ((EditText)findViewById(R.id.editText1)).getText());

                break ;
            default :
                Log.d(TAGS, "Event not managed");
        }
    }
}
```

3ième solution : Gestion de l'événement

Button dans le xml

32

- Le plus simple pour les boutons, ajouter dans `activity_main.xml`

```
android:onClick="sendMessage"
```

- Dans `MainActivity.java`

```
/** appelé quand on clique */  
public void sendMessage(View view) {  
    // blabla ...  
}
```

Exercice

33

- Créer une fenêtre de connexion avec passwd et date de naissance
- Le passwd est caché et c'est lui qui déclenche l'action d'affichage d'un Message : « Welcome » si le passwd est « xxx », sinon il affiche « invalid passwd »
 - ▣ `Toast.makeText(getApplicationContext(), "youhou ...", Toast.LENGTH_SHORT).show();`
 - ▣ *Attention*, il faut utiliser `...getText().toString()` pour transformer en String
- Un button OK est ajouté, il permet également de déclencher la connexion
- Un button Erase pour effacer le 2 champs
- Mettre des valeurs indicatives (Placeholder) dans les EditText : Hint

Les intentions

Mécanisme fondamental de transfert messages pour démarrer une **Activity** ou un **Service**

- *explicitement : en nommant l'**Activity***
- *implicitement : en laissant Android choisir*

*-L'**Intent** permet également de transférer des données*

Exemple : lancer une autre Activity

36

MainActivity -> Lancement de l'activité DetailActivity

```
Intent intent = new Intent(MainActivity.this, DetailActivity.class);
```

```
startActivity(intent); // démarrage activité
```

Exemple : lancer une autre Activity

37

Passage d'une donnée (« Hello ») à l'Activity

```
// définition d'un nom (identifiant) pour la donnée
public final static String EXTRA_MSG = "isfen.hello.MESSAGE";

Intent intent = new Intent(MainActivity.this, DetailActivity.class);

// ajout de la valeur du paramètre NOM
intent.putExtra(EXTRA_MSG, "Hello");

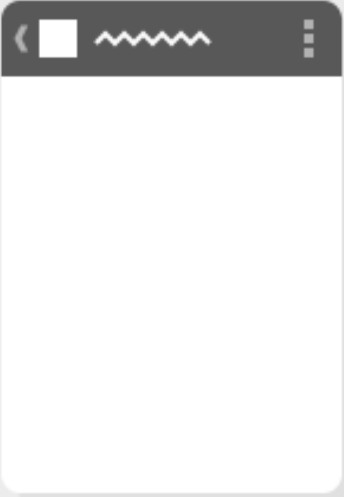
startActivity(intent); // démarrage activité
```

Création de **DetailActivity**

38

File->New->Activity->Empty Activity

Creates a new blank activity with an action bar.



Blank Activity

Activity Name:

Layout Name:

Title:

Menu Resource Name:

☐ Launcher Activity

Hierarchical Parent:

Package name:

AndroidManifest.xml

39

```
□ <?xml version="1.0" encoding="utf-8"?>
  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.isen.helloisen2intent" >

    <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>
      <activity
        android:name=".DetailActivity"
        android:label="@string/title_activity_detail"
        android:parentActivityName=".MainActivity" >
        <meta-data
          android:name="android.support.PARENT_ACTIVITY"
          android:value="fr.isen.helloisen2intent.MainActivity" />
        </activity>
      </application>
  </manifest>
```

Selection de l'application de lancement

40

```
<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>
```

Get an Intent : dans activité lancée

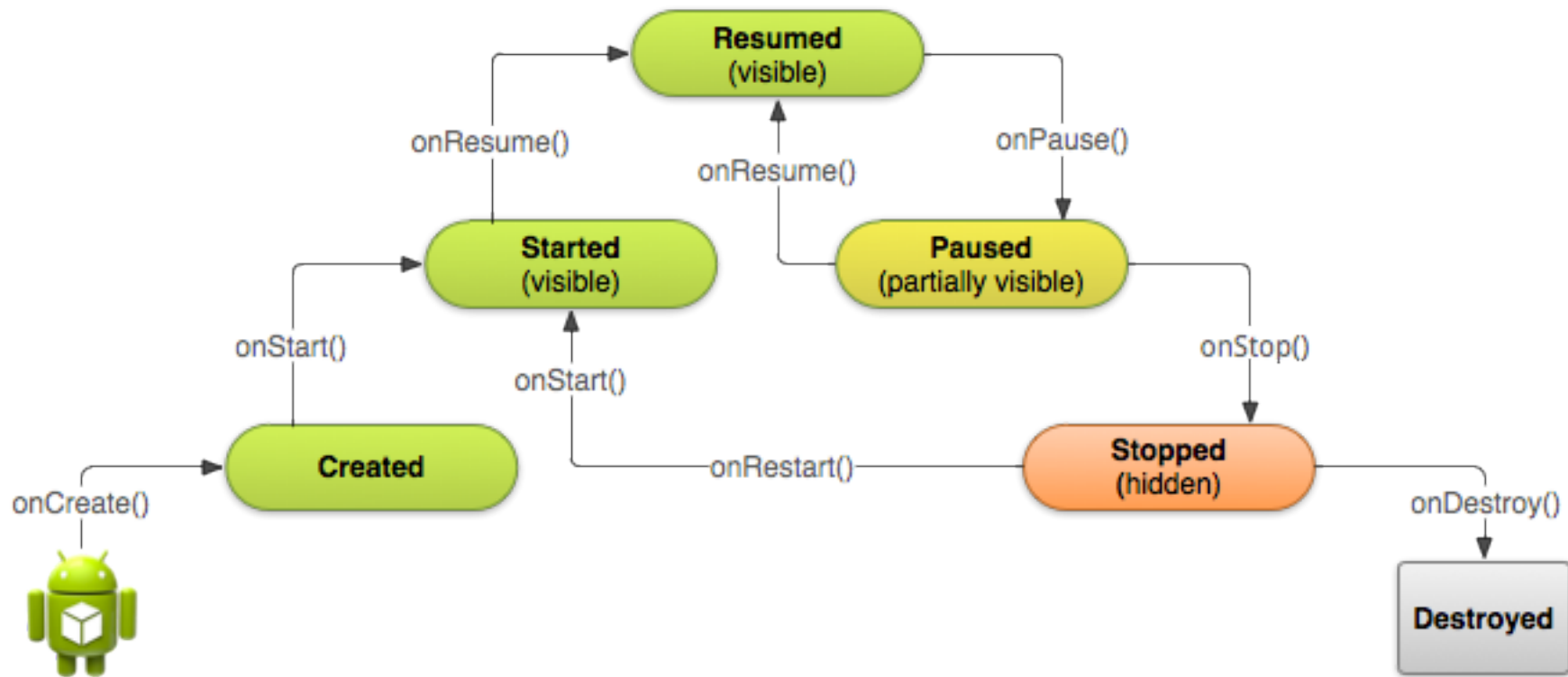
41

- **Récupération de l'intention qui a déclenché la création d'une application**

```
protected void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState);  
    Intent it = getIntent() ;  
    String str=  
it.getStringExtra(MainActivity.EXTRA_MSG);
```

Cycle de vie d'une activité

42



(Optionnel) Gestion du destroy de l'application

43

@Override

```
public void onDestroy() {  
    super.onDestroy(); // Always call the superclass  
  
    // Stop method tracing that activity started during onCreate()  
    android.os.Debug.stopMethodTracing();  
}
```


Que faire pendant le onPause()

44

- onPause : quand l'appli est encore visible par transparence, mais derrière une autre appli.
- Stopper les animations qui peuvent consommer.
- Sauvegarder les contenus. Ex : draft Mail
- Libérer les ressources system (Gps, écouteurs d'évènements..) pour limiter l'utilisation de la batterie

Stopping and Restarting an Activity

45

- Switch d'une application à une autre
- Lancement d'une autre activité à partir de l'activité (ex : lancement du browser, du mail...)
- Réception d'un appel téléphonique

- **Faire une fenêtre avec une demande de mot de passe et son nom**
- *Rajouter un bouton qui permet d'envoyer un mail à l'administrateur quand on a perdu son mot de passe*
- *Afficher bonjour 'le nom' dans la seconde fenêtre*
- *Faire une belle interface avec une image de fond et responsive*
- *Aide technique*
 - ▣ **Créer une classe contenant les constantes, EXTRA_MSG..., faire son import static (java 1.5) dans les autres classes**

```
import static fr.isen.hello_intent.Constantes.* ;
```
 - ▣ **Attention**, il faut utiliser `text.getText().toString()` pour transformer en String pour l'Intent

Début du projet

47

- ❑ Les promenades sonores, radio Grenouille
- ❑ <http://www.promenades-sonores.com/>
- ❑ Julie de Muer

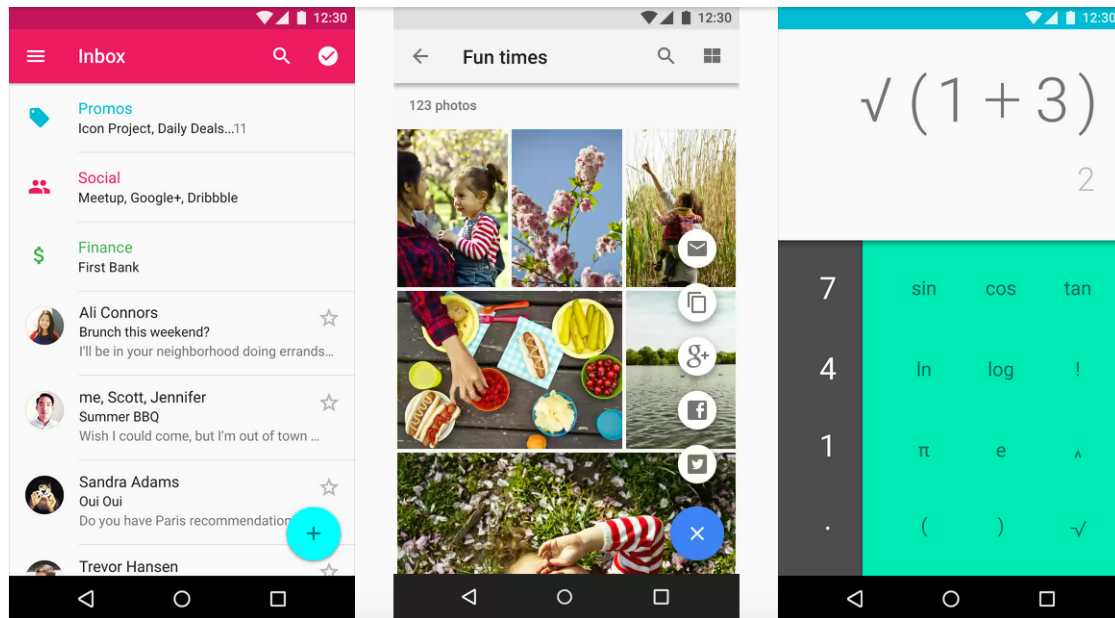


UX : Navigation Intuitive

48

▣ Les 3 leviers

- Regroupement logiques d'informations
- Animations pour l'information et le guidage
- Focus sur l'important



UX : Analyse de la navigation

49

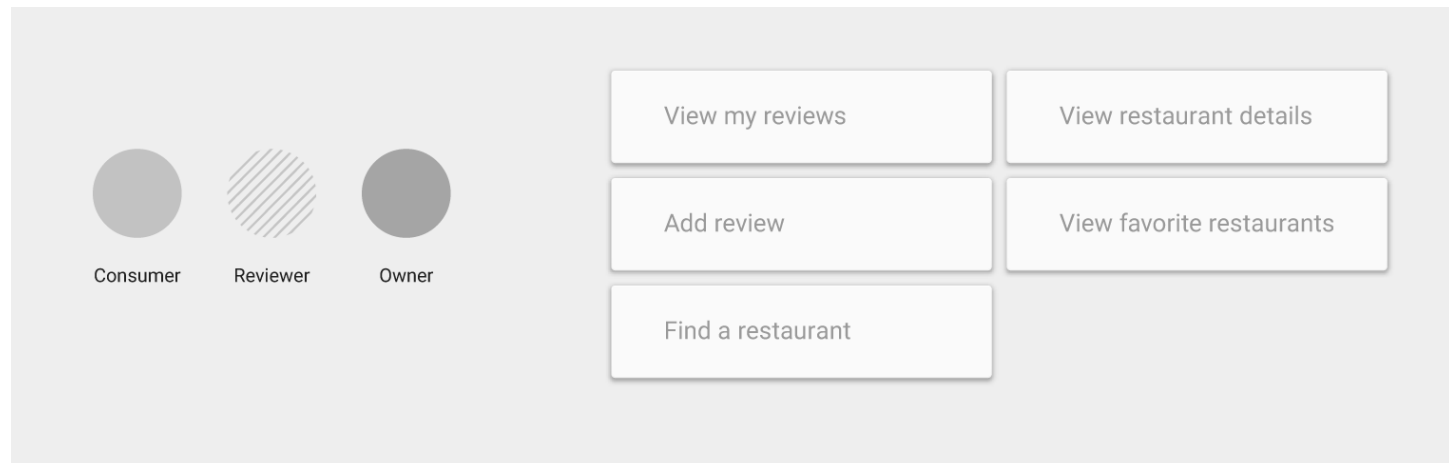
- ▣ Définir la structure de navigation
- ▣ Définir la hierarchie

[structure de navigation]

Définition de la structure

50

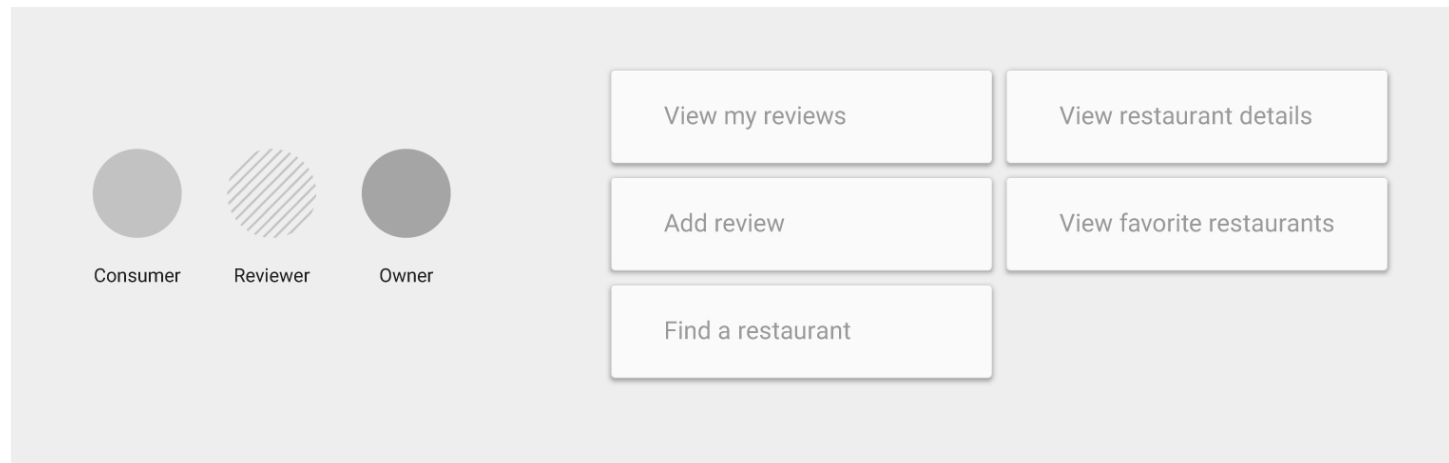
1. Faire l'inventaire des utilisateurs et des tâches
2. Pour chaque utilisateur, définir la priorité des tâches
3. Définir le séquençement des tâches



1) Inventaire

51

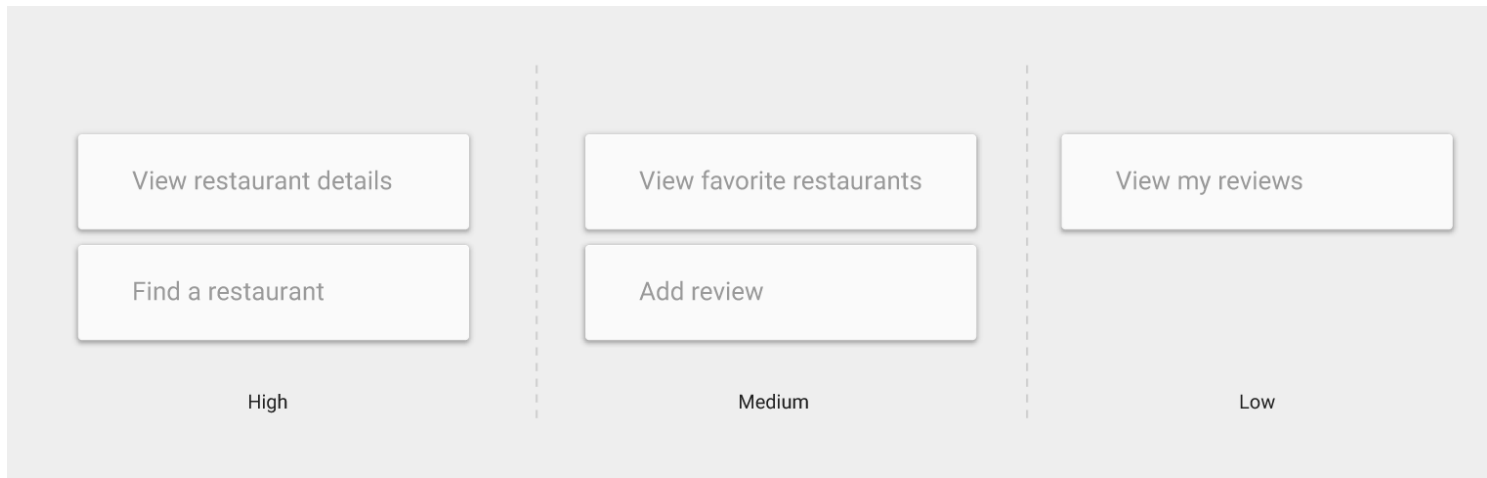
- ▣ Des utilisateurs
- ▣ Des tâches
- ▣ Exemple : App restaurant



2) Priorité

52

- En plusieurs catégories, ou priorité décroissante
- A faire pour chaque utilisateur si nécessaire



3) Définition des séquencements

53

- ▣ A faire pour chaque utilisateur si nécessaire
- ▣ Séquencement se recoupent parfois



[structure de navigation]

Définition de Hierarchie

54



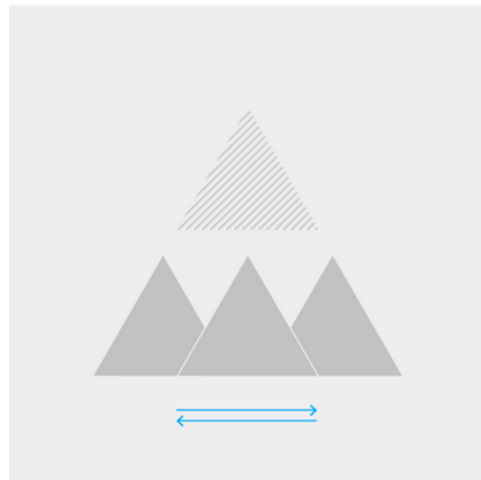
Home



Parent and child



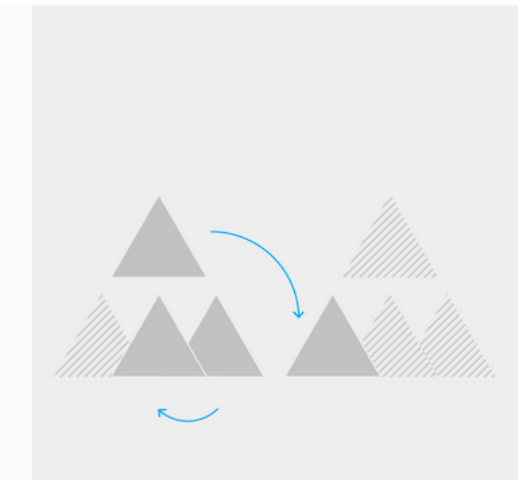
Navigation



Siblings



Collections



Links

Exercice UX

55

- ▣ Définir les utilisateurs
- ▣ Définir les actions
- ▣ Proposer des exemples de
 - Regroupement logiques d'informations
 - Informer, guider l'utilisateur par des Animations
 - Mise en valeur du contenu important

UX :

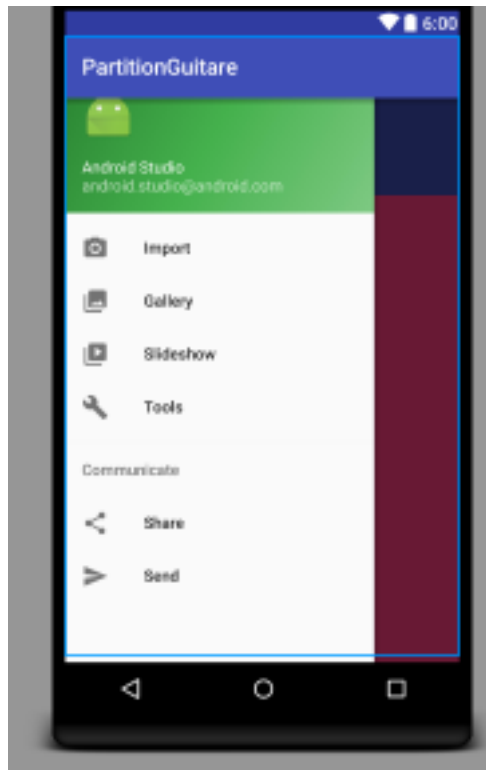
56

- ▣ Proposer des exemples de
 - Regroupement logiques d'informations
 - Informer, guider l'utilisateur par des Animations
 - Mise en valeur du contenu important

Navigation Drawer

57

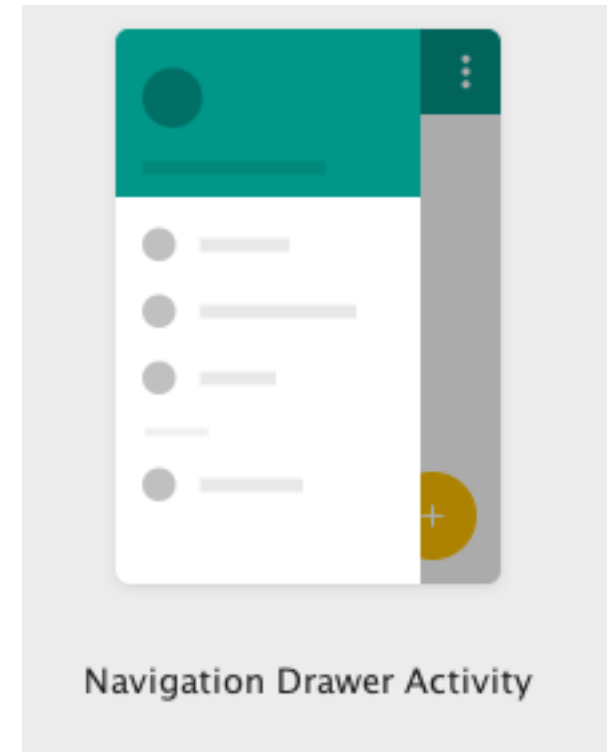
- Un Menu à gauche de Navigation



Début projet avec NavigationDrawer

58

- Créez un nouveau projet
 - ▣ Prendre à partir de la version Kit-Kat
 - ▣ Créer **Navigation Activity**
 - ▣ Title : Ballades Sonores



Fichiers générés

59

- **DrawerLayout** : activity_main.xml
 - **CoordinatorLayout** : app_bar_main.xml
 - **AppBarLayout**
 - **RelativeLayout** : content_main.xml
 - **FloatingActionButton**
 - **NavigationView**
 - **@layout/nav_header_main**
 - **@menu/activity_main_drawer**

DrawerLayout : dans l'ordre
contenu + navigation



app_bar_main.xml

60

Contenu de la frame principale

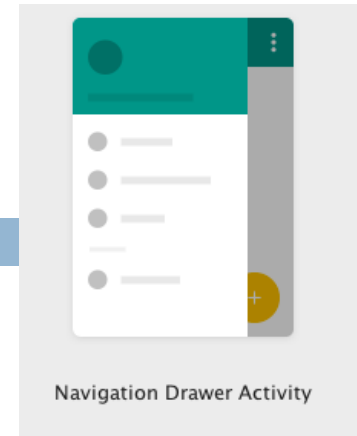
□ **CoordinatorLayout** (extends **FrameLayout**)

Création d'interactions entre les Childs : drags, swipes, flings,

□ **AppBarLayout** : barre du haut

□ + contenu principal : `content_main.xml` (Hello World)

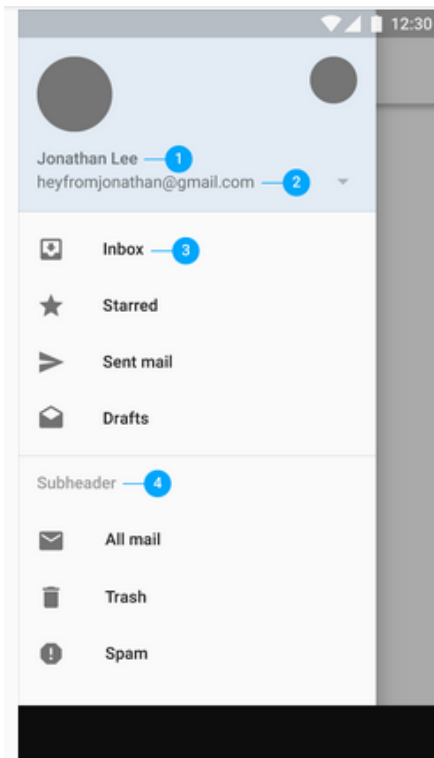
□ **FloatingActionButton**: **button** à supprimer si non nécessaire



NavigationView (app_bar_main.xml)

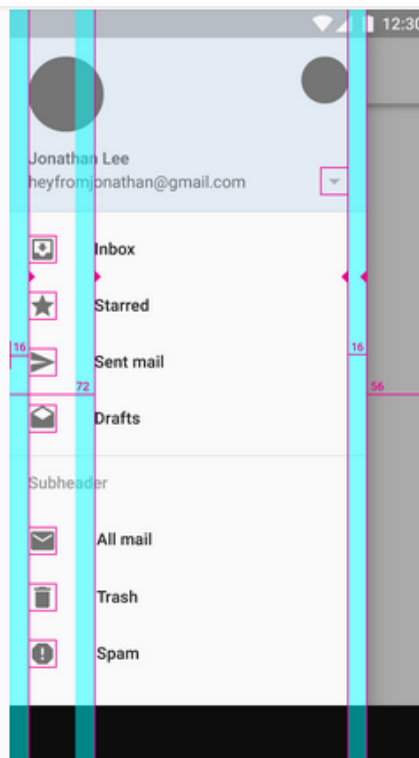
header + menu

61



Typography

1. Name: Roboto Medium, 14sp, #FFFFFF
2. Email address: Roboto Regular, 14sp, #FFFFFF
3. List item: Roboto Medium, 14sp, 87% #000000
4. Subheader: Roboto Medium, 14sp, 54% #000000. Aligns to the 16dp keyline.



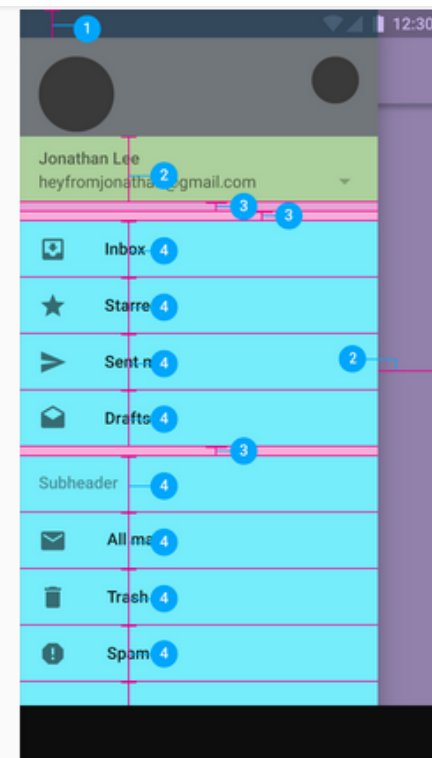
Keylines and margins

Icons align at screen left and right margins: 16dp

Icon values: 54% #000000

Content associated with an icon or avatar left margin: 72dp

Side nav width: Equal to the screen width minus the height of the action bar. In the example shown this is 56dp from the right edge of the screen. The maximum width of the nav



Vertical spacing

1. Status bar: 24dp
2. Subtitle: 56dp
3. Space between content areas: 8dp
4. Subtitles and list items: 48dp

Add 8dp padding at the top and bottom of every list grouping. One exception is at the top of a list with a subheader, because subheaders contain their own padding.

Customisation (1 / 2)

=> changement couleur (...img)

62

□ Changement des couleurs

- De l'icône, du texte

- De l'image de fond des items // img on verra plus tard

<android.support.design.widget.NavigationView

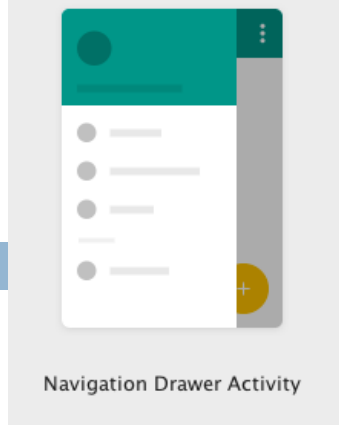
...

```
app:itemIconTint="#2196f3"
```

```
app:itemTextColor="#00DD00"
```

```
app:itemBackground="@drawable/my_img"
```

...



Customisation (2/2)

=> Chgt couleur selected/not

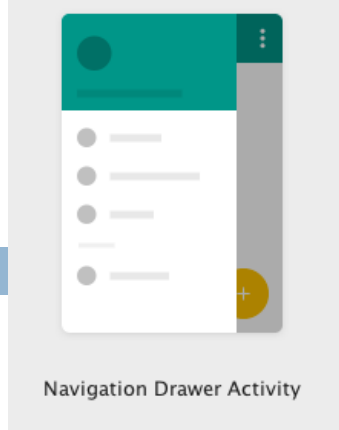
63

- *Changer de couleur en fonction de l'état*
- [Studio] Créer un répertoire color dans res (new/android resource dir)
 - Créer un fichier: state_list.xml (new/android resource file)

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- This is used when the Navigation Item is checked -->
  <item android:color="#ee6911" android:state_checked="true"/>
  <!-- This is the default text color -->
  <item android:color="#1199EE"/>
</selector>
```

- Remplacer l'affectation de la couleur par:
`app:itemTextColor="@color/state_list"`

On peut faire pareil pour l'icone et le background



Ajoutons du contenu dans l'Activity

64

- Il faut ajouter des Containers pour ajouter plusieurs page internes (Fragment)
- Dans les containers, il faut ajouter des Fragments
- Exemple de fragment : Page Vide, liste, Map...
- Ajoutons du swipe pour changer de fragment

Layout: une page qui défile avec indicateur : ViewPager+PagerStrip

65

- ▣ du swipe entre les pages : [ViewPager](#)
- ▣ Un indicateur des tabs : [PagerStrip](#)
- ▣ Des Fragments pour chaque fenêtre interne



1 / 5 Modification *res/layout/content_main.xml*

66

- ViewPager+ PagerTabStrip : remplacer entièrement le code de content_main.xml par

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.view.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/pager"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"

    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <android.support.v4.view.PagerTabStrip
        android:id="@+id/pager_header"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:backgroundTint="#002299"
        android:textColor="#f00"/>
    </android.support.v4.view.ViewPager>
```

2/5 Création des fragments internes

67

Exemple de création d'un fragment avec écrit
« Home »

- File->New->Fragment ->Fragment Blank
- Exemple HomeFragment
- Décocher “include interface callback »

—

...

3/5 Modification *fragment_home.xml*

68

Modif du .xml du fragment

- Changer la couleur
- changer le texte

4/5 MainActivity incluant PagerAdapter (1/2)

69

```
public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener,
        HomeFragment.OnFragmentInteractionListener {

    Fragment[] allFrgs = new Fragment [1] ; // création d'un tableau contenant les fragmets
    ViewPager mViewPager; // gestionnaire de pages (Pager)
    FragsPagerAdapter fragsPagerAdapter; // adapteur des fragments au Pager : classe interne

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        allFrgs[0] = HomeFragment.newInstance(); // création de la fenetre interne : fragment

        setContentView(R.layout.activity_main);

        // Création de l'Adapter des fragments et on l'affecte au viewPager
        // ViewPager use support library fragments, so use getSupportFragmentManager.
        fragsPagerAdapter = new FragsPagerAdapter(getSupportFragmentManager());
        mViewPager = (ViewPager) findViewById(R.id.pager);
        mViewPager.setAdapter(fragsPagerAdapter);

        ...
    }
}
```

Attention, pour les **import** il faut utiliser **PARTOUT** : **import android.support.v4.app.Fragment** ;

ATTENTION Modifier **HomeFragment.java** pour qu'il n'y ait plus de passage de paramètre

4/5 MainActivity incluant PagerAdapter (2/2)

70

```
...
@Override
public void onFragmentInteraction(Uri uri) {
}

/**
 * adapteur minimaliste pour les fragments
 */
public class FragsPagerAdapter extends FragmentStatePagerAdapter {
    public FragsPagerAdapter(FragmentManager fm) {
        super(fm);
    }
    @Override
    public Fragment getItem(int i) {
        return allFrgs[i];
    }
    @Override
    public int getCount() {
        return allFrgs.length ;
    }
    @Override
    public CharSequence getPageTitle(int position) {
        return "Title " + (position+1);
    }
}
}
```

4/5 [UX] Gestion de la touche Back

71

La touche back doit permettre de revenir aux fragments précédents, puis à quitter l'application si on est dans le fragment tout à gauche

Modifier la méthode `onBackPressed` pour qu'elle corresponde au code suivant

```
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        if (mViewPager.getCurrentItem() == 0) {
            // If the user is currently looking at the first step, allow the system to handle the
            // Back button. This calls finish() on this activity and pops the back stack.
            super.onBackPressed();
        } else {
            // Otherwise, select the previous step.
            mViewPager.setCurrentItem(mViewPager.getCurrentItem() - 1);
        }
    }
}
```

▼

Remarque : 2 types d'Adapter existent

72

- ▣ FragmentPagerAdapter

Nombre fixe et faible de pages

- ▣ FragmentStatePagerAdapter

Nombre de pages indéterminé, construction, destruction à chaque mouvement de doigt, minimize l'utilisation mémoire

Création d'autres fragments

73

- Créer 2 fragments en utilisant les facilités Visual Studio

ListBalladesFragment.java

New->fragment->fragment(List)

MapFragment.java

New->fragment->fragment (Blank)

MapFragment :

supprimer la méthode `onButtonPressed`

Modification des interfaces

74

- Les deux classes de Fragment générées incluent automatiquement les interfaces :

MapFragment.[OnFragmentInteractionListener](#)

ListBalladesFragment.[OnListFragmentInteractionListener](#)

Ces interfaces servent à définir la communication entre les Fragment et la [MainActivity](#)

- Il faut implémenter ces interfaces dans [MainActivity](#)

Class MainActivity ... implements MapFragment.[OnFragmentInteractionListener](#),
ListBalladesFragment.[OnListFragmentInteractionListener](#)

- Les modifier si nécessaire pour faire passer les paramètres souhaités. Par exemple faire passer une [String](#) au lieu de l'[URI](#) par défaut dans MapFragment.[OnFragmentInteractionListener](#)

Explication sur le Wizard Fragment (list)

75

- Le Wizard Fragment (list) crée
 - ▣ fragment_ballade_list.xml : layout affichant une list
 - ▣ fragment_ballade.xml : layout d'un elt de la list

- ▣ BalladeFragment : le Fragment qui fait correspondre une vue à une List de données
- ▣ MyBalladeRecyclerViewAdapter : l'Adapter qui crée une vue adaptée aux données
- ▣ DummyContent: une classe créant des Données fictives (Dummy)
 - DummyItem: une donnée fictive

Pour son propre projet il faut remplacer Dummy et Adapter

- Créer l'appli avec les fragments : Home, List, Apropas
- Modifier la liste pour y ajouter du contenu récupéré à partir du site web des promenades sonores

"12", "Le Fantôme du Sémaphore"

"4", "Gardanne 3 couleurs"

"19", "Frioul"

[UX] ViewPager

animation des transitions

77

- Gestion affichage des pages et adjacentes
- Faire une classe (ex `ZoomOutPageTransformer`) qui implémente `ViewPager.PageTransformer`
 - ▣ `public void transformPage(View view, float position) ;`
 - ▣ position `[-1, 1]` => 0 quand centrée
 - ▣ utiliser `setAlpha()`, `setTranslationX()`, `setScaleY()`, `setRotation()`....
- Affectation du `Transformer` au `ViewPager`

...

```
mViewPager.setPageTransformer(true, new ZoomOutPageTransformer());
```

[UX] Viewpager

Exemple 1 : Zoom

78

```
public class ZoomOutPageTransformer implements ViewPager.PageTransformer {
    private static final float MIN_SCALE = 0.85f;
    private static final float MIN_ALPHA = 0.5f;

    public void transformPage(View view, float position) {
        int pageWidth = view.getWidth();
        int pageHeight = view.getHeight();

        if (position < -1) { // [-Infinity,-1) This page is way off-screen to the left.
            view.setAlpha(0);

        } else if (position <= 1) { // [-1,1]
            // Modify the default slide transition to shrink the page as well
            float scaleFactor = Math.max(MIN_SCALE, 1 - Math.abs(position));
            float vertMargin = pageHeight * (1 - scaleFactor) / 2;
            float horzMargin = pageWidth * (1 - scaleFactor) / 2;
            if (position < 0) {
                view.setTranslationX(horzMargin - vertMargin / 2);
            } else {
                view.setTranslationX(-horzMargin + vertMargin / 2);
            }

            // Scale the page down (between MIN_SCALE and 1)
            view.setScaleX(scaleFactor);
            view.setScaleY(scaleFactor);

            // Fade the page relative to its size.
            view.setAlpha(MIN_ALPHA +
                           (scaleFactor - MIN_SCALE) /
                           (1 - MIN_SCALE) * (1 - MIN_ALPHA));

        } else { // (1,+Infinity]
            // This page is way off-screen to the right.
            view.setAlpha(0);
        }
    }
}
```

[UX] Viewpager

Exemple 2 : Venir du fond

79

```
□ public class DepthPageTransformer implements ViewPager.PageTransformer {  
    private static final float MIN_SCALE = 0.5f;  
  
    public void transformPage(View view, float position) {  
        int pageWidth = view.getWidth();  
  
        if (position < -1) { // [-Infinity,-1)  
            // This page is way off-screen to the left.  
            view.setAlpha(0);  
  
        } else if (position <= 0) { // [-1,0]  
            // Use the default slide transition when moving to the left page  
            view.setAlpha(1);  
            view.setTranslationX(0);  
            view.setScaleX(1);  
            view.setScaleY(1);  
  
        } else if (position <= 1) { // (0,1]  
            // Fade the page out.  
            view.setAlpha(1 - position);  
  
            // Counteract the default slide transition  
            view.setTranslationX(pageWidth * -position);  
  
            // Scale the page down (between MIN_SCALE and 1)  
            float scaleFactor = MIN_SCALE  
                + (1 - MIN_SCALE) * (1 - Math.abs(position));  
            view.setScaleX(scaleFactor);  
            view.setScaleY(scaleFactor);  
  
        } else { // (1,+Infinity]  
            // This page is way off-screen to the right.  
            view.setAlpha(0);  
        }  
    }  
}
```

80

L'action Bar

Modification de la barre de Menu

Ajout d'items dans la barre

/menu/main.xml

81

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity">
```

```
    <item android:id="@+id/action_search"
          android:icon="@android:drawable/ic_search"
          android:title="@string/action_search"
          app:showAsAction="ifRoom"
        />
```

```
    <item android:id="@+id/action_settings"
          android:title="@string/action_settings"
          android:orderInCategory="100"
          app:showAsAction="never" />
</menu>
```

Création d'une nouvelle activité : promenade

Faire une jolie activité pour chaque promenade
Cette activité sera appelée quand on cliquera
sur la liste

TP préliminaire

83

On veut déclencher une activité **Ballade** quand on clique sur un élément de la liste des ballades

- Créer l'activité **BalladeDesc** contenant 2 tabs
 - ▣ File/New/Activity/Tabbed Activity
 - ▣ Dans [Navigation Style], sélectionner **Action Bar Tabs (with ViewPager)**
- **Info** : le déclenchement de l'**Intent** pour créer la nouvelle activité se fait dans *onFragmentInteraction()*

Les classiques

A partir de données et d'un Adapter

Les classiques



85

- *RelativeLayout* : les uns par rapports aux autres et aux bords de l'écran
- *LinearLayout* : horizontal ou vertical avec des notions de weight (proportion)
- *FrameLayout* : supposition en haut à gauche
- *GridLayout* (API 14) : grille
- <http://developer.android.com/guide/topics/ui/declaring-layout.html>

Création d'un linear Layout

86

<RelativeLayout

... composants

<LinearLayout

android:layout_width="match_parent"

android:layout_height="match_parent"

android:orientation="horizontal" >

... rajouter les composants en leur donnant des poids

</LinearLayout>

</RelativeLayout>

LinearLayout dimensionnement

87

- *Définir proportions et les autres complètent la place*
- *Exemple*

Premier composant

- ▣ `android:layout_weight= "1 "`
- ▣ `android:layout_width="0dp »`

Second composant

- ▣ `android:layout_weight= "2"`
- ▣ `android:layout_width="0dp »`

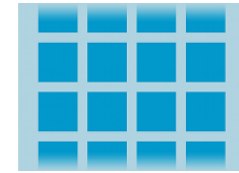
Le dernier composant meublera

A partir de données et d'un Adapter

88

- *ListView, GridView*

- Exemple simple pour une *ListView*



```
String [] myTab = {"un", "deux", "quatre"};
```

```
ArrayAdapter <String> adapter=
```

```
    new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, myTab);
```

```
ListView listView = (ListView) findViewById(R.id.listview); // id : layout.xml
```

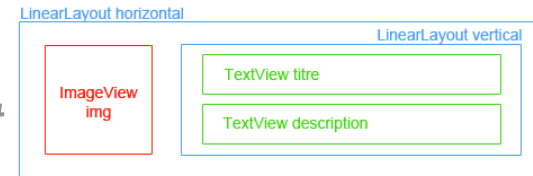
```
listView.setAdapter(adapter);
```

Création *Adapter Custom* pour une *ListView* (1 / 3)

89

1)Création de la *ListView*

Pour un *Fragment* : File->New->Fragment->Fragment List



2) Création du layout de l'item (*item_listballade.xml*) : (2 / 3)

3) Création de l'*Adapter*

- ▣ utiliser un *SimpleAdapter* (3/3)
- ▣ **OU** Création d'une classe *BalladeItemAdapter* qui hérite de *BaseAdapter*
- ▣ **OU** modifier l'adapter automatiquement généré par Android Studio (*MyItemReViewAdapter*)

4) Modification Java (*ListBalladeFragment*) généré : tp. 3/3

- ▣ Création d'une *ArrayList* de *HashMap* pour les *Data*
- ▣ Affection de l'*Adapter*

Modification du .xml généré pour la liste

fragment_item_list.xml

90

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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="fr.grenouille.balladessonores_db.ListBalladesFragment">

    <ListView android:id="@android:id/list"
    android:layout_width="match_parent"
        android:layout_height="match_parent" />

</FrameLayout>
```

Création du .xml (Layout) pour l'item *item_listballade.xml* (2/3)

91

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    >
    <ImageView
        android:id="@+id/img"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:padding="10px"
    />
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:paddingLeft="10px"
        android:layout_weight="1"
    >
        <TextView android:id="@+id/titre"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:textSize="16px"
            android:textStyle="bold"
        />
        <TextView android:id="@+id/description"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
        />
    </LinearLayout>
</LinearLayout>
```



Modification Java généré: BalladeFragment.java

(2/3)

92

```
// création de données pipo pour l'image, le nom et l'auteur

List listItem ;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mColumnCount = getArguments().getInt(ARG_COLUMN_COUNT);
    }
    // creation des données
    listItem = new ArrayList<HashMap<String, String>>();
    // recup d'un cursor avec le descriptif de la ballade
    for(int i=0 ; i < 3 ; i++) {
        HashMap<String, String> map = new HashMap<String, String>(); // img id
        String imgId = String.valueOf(getResources().getIdentifier("promenade_icon_"+i, "drawable",
getActivity().getPackageName()));
        Log.d(TAGS, "ImgId : "+imgId) ;
        map.put("imgId", imgId);
        map.put("Nom", "Nom_"+i);
        map.put("Auteur", "Auteur_"+i);
        listItem.add(map);
    }
}
```

Modification Java généré: BalladeFragment.java

(2/3)

93

```
// Création et affectation de l'adapter
// ajout du listener de click

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_item_list, container, false);
    AbsListView mListView = (AbsListView) view.findViewById(android.R.id.list);
    // Set the adapter
    SimpleAdapter mAdapter = new SimpleAdapter(getActivity(), listItem,
        R.layout.item_listballade,
        new String[] {"imgId", "Nom", "Auteur"},
        new int[] {R.id.img, R.id.titre, R.id.description});
    ((AdapterView<ListAdapter>)mListView).setAdapter(mAdapter);

    mListView.setOnItemClickListener(new AbsListView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            if (null != mListener)
                mListener.onListFragmentInteraction();
        }
    });

    return view;
}
```

Création des images

94

Utiliser l'outil Android file resizer pour générer les images des icones avec comme nom :

Ou le plugins :

<https://www.javahelps.com/2015/02/android-drawable-importer.html>

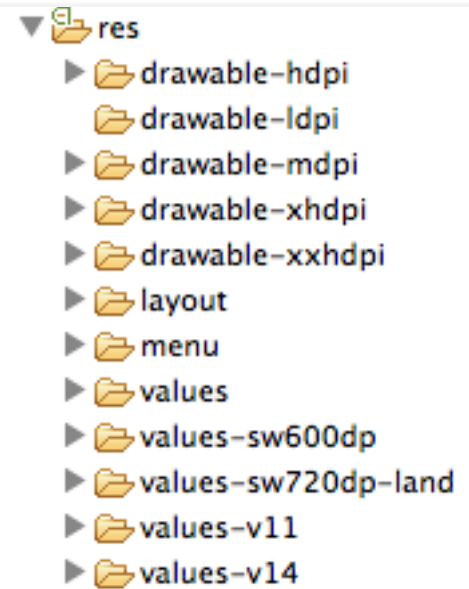
Rajouter le listener sur une Liste

95

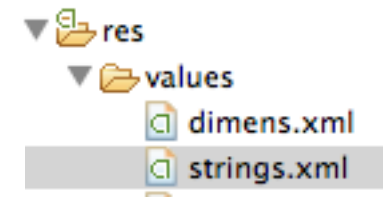
- 1) Rajouter à `ListBalladeFragment`
implements `AbsListView.OnItemClickListener`
- 2) Rajouter la méthode qui appelle le listener
public void onItemClick(`AdapterView<?>` parent,
`View` view, **int** position, **long** id) {
 ...
 mListener.onFragmentInteraction(position);
 ...
}

Les ressources

Il existe tout un système de rangement **res/** avec des **.xml** pour : les images, les layout, les menus, les string, les tableaux d'entiers ou de chaînes, les couleurs, les styles, les dimensions, les animations



Les ressources String strings.xml ou autre fichier

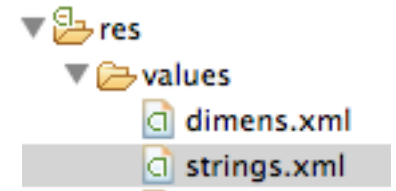


97

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="prenom">"Bob"</string>
    <string name="nom">"<b>Lafleche</b>"</string>
    <string name="directeur">Directeur : %s</string>

    <!-- Promo number (year) -->
        <integer-array name="numPromo">
            <item>2013</item>
            <item>2014</item>
            <item>2015</item>
        </integer-array>

    <!-- goldfather name -->
        <string-array name="parrainsPromo">
            <item>Onet</item>
            <item>Al Pacino</item>
            <item>Fantomax</item>
        </string-array>
</resources>
```



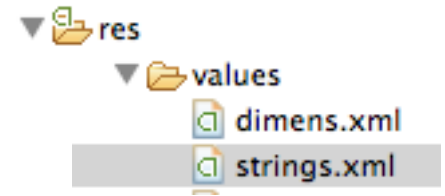
Lire les valeurs des ressources

98

- Depuis un autre **.xml** : **@type/nom_var**
 - @string/directeur**
 - @color/transp_blue**
 - @drawable/ic_launcher**

- Depuis un code **.java**, les méthodes nécessitent un identifiant ou une instance
 - ▣ Pour récupérer l' **identifiant** d'une ressource : **R.type.nom_var**
 - R.string.directeur**
 - R.color.transp_blue**
 - R.drawable.ic_launcher**

 - ▣ Pour récupérer la **valeur** de la ressource :
 - getRessources().getType(identifiants)**
 - String name = getResources().getString(R.string.directeur);**
 - int coul = getResources().getColor(R.color.transp_blue);**
 - ...**



Exemples de ressources

99

// method which requires identifiant

```
Toast.makeText(getApplicationContext(), R.string.nom, Toast.LENGTH_SHORT).show();
```

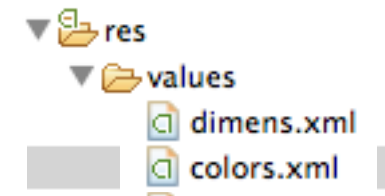
// get the name of the boss in the ressources : instance

```
String directeur = String.format(getString(R.string.directeur),  
    getString(R.string.nom));
```

// get all the name of the boss and concat

```
String[] dir=getResources().getStringArray(R.array.parrainsPromo) ;  
String allDir = "" ;  
for(String name : dir)  
    allDir = allDir.concat(name+"\t");
```

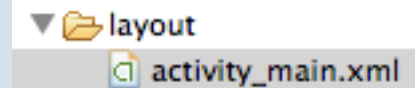

Les ressources colors.xml (ou autre .xml)



100

```
<resources>
    <color name="deep_blue">#00F</color>
    <color name="transp_blue">#AA0000FF</color>
</resources>
```

Formats : #RGB, #TRGB, #RRGGBB, #TTRRGGBB



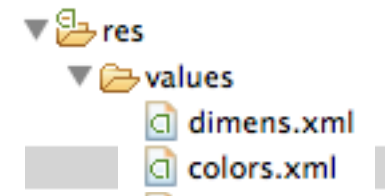
Utilisation dans le layout (activity_main.xml)

...

```
android:textColor="@color/transp_blue"
```

...

Les ressources dimens.xml (ou autre .xml)



101

```
<resources>
```

```
    <!-- Default screen margins, per the Android Design guidelines. -->
```

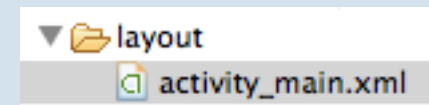
```
    <dimen name="activity_horizontal_margin">16dp</dimen>
```

```
    <dimen name="activity_vertical_margin">16dp</dimen>
```

```
</resources>
```

*Formats : px (pixel), in, pt, mm, **dp** (pixel density), **sp** (dp pour Font)*

Utilisation dans le layout (activity_main.xml)



...

```
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

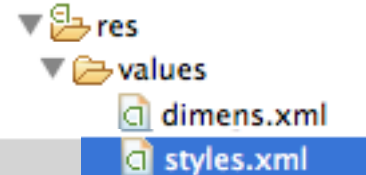
...

Les ressources, définition dans **styles.xml** et utilisation dans les **layout/a_b.xml**

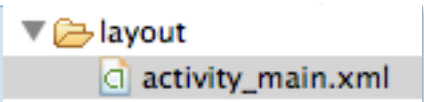
102

style: ensemble de propriétés pour une View (Font, size, color, padding, background, height...)

```
<!-- style.xml. ->
<style name="smallTxt">
    <item name='android:textSize'>14sp</item>
    <item name='android:textColor'>@color/transp_blue</item>
</style>
<style name="bigTxt">
    <item name='android:textSize'>18sp</item>
    <item name='android:textColor'>@color/transp_orange</item>
</style>
```



```
<!-- activity_main.xml. -->
<TextView
...
    android:textAppearance="@style/bigTxt"
/>
```



L'utilisation des styles d'android

103

Mieux que de redéfinir des styles, utiliser le thème courant avec
?android:

```
<!-- activity_main.xml. -->
<TextView
...
    android:textColor="?android:textColorPrimaryInverse"
/>
```

L'utilisation des thèmes

Theme.Light, Theme.Translucent, Theme.NoTitleBar.Fullscreen, Theme.Black...

104

Ex. création "**CustomTheme**" à partir Theme existant :Theme.Light

```
<!-- style.xml →  
<color name="custom_theme_color">#b0b0ff</color>  
<style name="CustomTheme" parent="Theme.AppCompat.Light">  
    <item name="android:windowBackground">@color/custom_theme_color</item>  
    <item name="android:colorBackground">@color/custom_theme_color</item>  
</style>
```

Définition thème pour l'application ou chaque activité séparément

```
<!-- AndroidManifest.xml →  
<application android:theme="@android:style/Theme.Translucent">  
...  
<activity android:theme="@style/CustomTheme">
```

For more about creating and using themes, read the [Styles and Themes guide](#).

Les autres ressources

105

- Les images : `mipmap/` pour les icones et `drawable/` pour les autres
- les layout : positionnement des éltls dans les fenêtres
- les menus
- les animations...

- Utiliser les ressources système : rajouter le mot `android`
 - `(.xml)` `android:textColor="@android:color/dark_gray"`
 - `(.java)` `..getResources().getColor(android.R.color/dark_gray);`

Selection des ressources

106

- Créer une arborescence pour les ressources en séparant chaque critère par '-'
- Mobile Country-Code : `mcc234-mnc20/mcc310/...`
- Langue et région : `en/fr/en-rUS/en-rGB/...`
- Largeur minimale de l'écran : `sw320dp/sw600dp/...`
- Largeur de l'écran disponible : `w320dp/w600dp...`
- Hauteur d'écran : `h480dp/...`
- Taille de l'écran : `small/medium/karge/xlarge, ...`
- Largeur/Longueur
- Orientation : `port/land/square`
- Mode dock, `car` ou `desk`
- Mode nuit ou jour : `night/notnight`
- Densité de l'écran en pixel : `ldpi, mdpi, hdpi, xhdpi,...`
- Type d'écran tactile : `notouch/stylus/finger`
- Visibilité du clavier : `keysexposed/keyshidden/keyssoft`
- Type du clavier : `nokeys/qwerty/12key`
- Visibilité touche de navigation : `navexposed/navhidden`
- Type de navigation : `nonav/dpad/tracball/whele`
- Version de l'api : `v7/v8....`

Exemple : `layout-xlarge-port-keyshidden, layout-large-land, ...`

Exemple de la langue

107

□ Créer une arborescence pour les ressources

```
MyProject/  
  res/  
    values/  
      strings.xml  
    values-es/  
      strings.xml  
    values-fr/  
      strings.xml.
```

- Le fichier sera automatiquement chargé en fonction de la valeur du paramètre 'langue' du mobile

Plusieurs taille d'écrans : gestion des layouts

108

- 4 tailles : *small, normal, large, xlarge...*
- 4 densités de pixels : *low (ldpi), medium (mdpi), high (hdpi), extra high (xhdpi)...*

-> Créer une arborescence

```
MyProject/  
  res/  
    layout/  
      main.xml  
    layout-large/  
      main.xml
```

Plusieurs layout en fonction de l'orientations et taille d'écrans

109

MyProject/

res/

layout/

default (portrait)

main.xml

layout-land/

landscape

main.xml

layout-large/

large (portrait)

main.xml

layout-large-land/

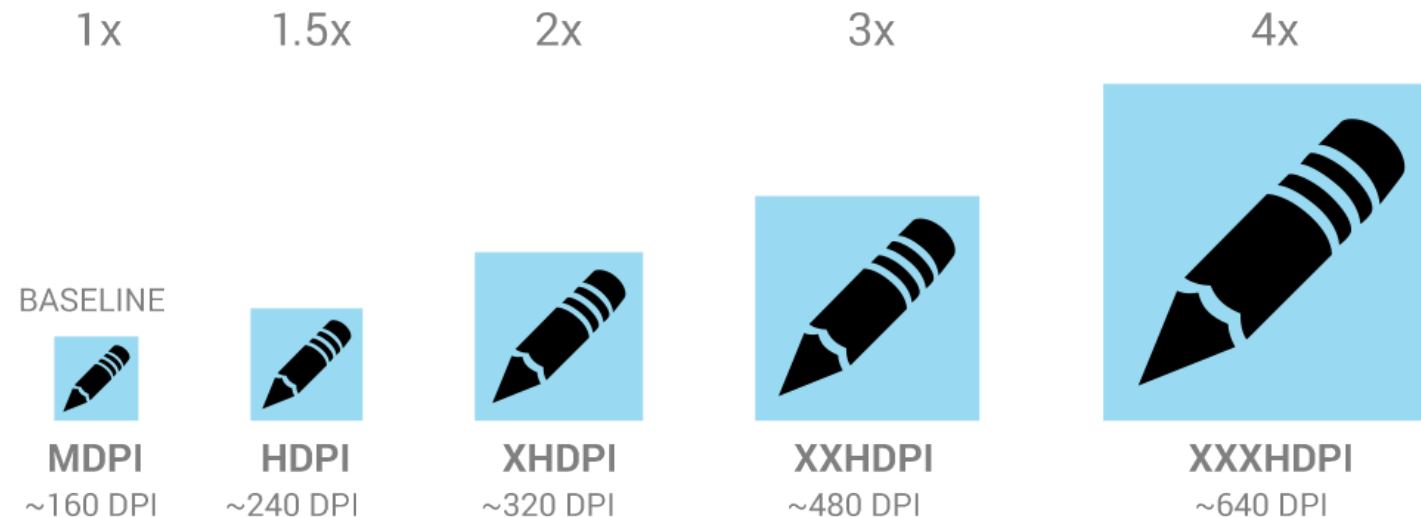
large landscape

main.xml

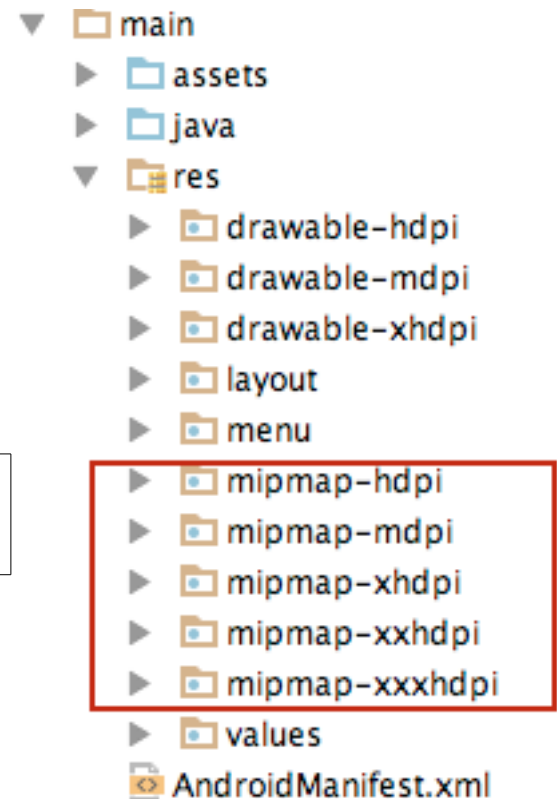
Rapport de tailles des images entre-elles

110

□ **mdpi**: 1.0 (baseline) : 48x48 pour icônes



Les icones : mipmap



- [button](#)
- <https://programmium.wordpress.com/2014/03/20/mipmapping-for-drawables-in-android-4-3/>

Click gauche sur res/ new Image Asset

-> Modification de manifests/AndroidManifest.xml

Gestion des tailles des images

112

MyProject/

res/

drawable-xxxxxxxhdpi/

awesomeimage.png

drawable-xxhdpi/

awesomeimage.png

drawable-xhdpi/

awesomeimage.png

drawable-hdpi/

awesomeimage.png

drawable-mdpi/

awesomeimage.png

drawable-ldpi/

awesomeimage.png

Changement dynamique de ressources

113

- On remplacer la lecture d'un fichier de ressource sur un événement en surchargeant dans l'Activity

```
public void onConfigurationChanged(Configuration newConfig) {...}
```

Déclencheurs : *mcc, mnc (évt carte SIM), locale, keyboardHidden, keyboard, fontScale, uiMode (voiture/jour/nuit), orientation, screenLayout(activation nouvel écran), screenSize (portrait/paysage), smallestScreenSize*

- rajouter les déclencheurs séparés par '|' dans <activity> (**Manifest**)
android:configChanges="screenSize | orientation | keyboardHidden »
- rajouter dans l'activité (**MainActivity.java**)

```
public void onConfigurationChanged(Configuration newConfig) {  
    super.onConfigurationChanged(newConfig);  
    if(newConfig.orientation==Configuration.ORIENTATION_LANDSCAPE)  
    ...  
}
```

- Créer l'activity **Ballade** contenant 2 tabs
 - ▣ Ecouter
 - ▣ Info
- Customisez la liste pour qu'elle contienne des images et du texte
- A partir des données récupérées sur le site pour 2 ballades
- Afficher les informations dans l'un des Fragments
 - ▣ Titre
 - ▣ Numéro
 - ▣ Durée
 - ▣ ...



115

Media Player

La classe Média Player permet de lire des videos et des fichiers audio qui peuvent provenir

1. Ressources locales : répertoire res/raw
2. URLs externes (streaming)
 - Dans ce cas il faut rajouter la permission
`<uses-permission android:name="android.permission.INTERNET" />`
3. Content Provider : Uri Interne

1. Ressources locales : répertoire res/raw

1^{er} cas : on connaît le nom de la ressource : `R.raw.balade_canelet`

```
MediaPlayer mediaPlayer = MediaPlayer.create(  
    getApplicationContext(), //fragment  
    R.raw.balade_canelet); // .mp3
```

Ou 2^{ième} cas : on doit retrouver la ressource à partir du nom : `"ballade_canelet"`

```
String soundName = "ballade_canelet";
```

```
MediaPlayer mediaPlayer = MediaPlayer.create(getContext(),  
    Uri.parse("android.resource://" + getContext().getPackageName()+"/raw/" + soundName));
```

```
// démarrage de la lecture dans les 2 cas
```

```
mediaPlayer.start();
```

2. URLs externes (streaming)

```
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
try {
    mediaPlayer.setDataSource(urlName);
    mediaPlayer.prepare(); // buffering, etc
    mediaPlayer.start();
} catch (IOException e) {
    e.printStackTrace();
}
```

Ne pas oublier de modifier le Manifest et faire

...

```
<uses-permission android:name="android.permission.INTERNET"/>
```

```
<application ...
```

Ecrire les tabs des infos des ballades : écouter, Infos, plan

Application en anglais et en français

Utilisation d'un style perso pour Ecouter, du thème du mobile pour les autres

Inverser les couleurs des textes quand on change l'orientation

Créez le fragment `FragmentPlay` qui permettra d'écouter une ballade.

Rajouter un bouton de type `ImageButton` qui déclenche le début et la pause de l'écoute.

Changer l'icone quand on joue ou on arrête la ballade

1. Récupération du fichier Json
http://www.laurent-freund.fr/cours/android/ballades_all.json
2. Recopie en local dans le répertoire Asset
3. Transformation du fichier Json en String
4. Parsing de la String pour récupérer les infos

Recopie en local dans Asset

120

- Créer le répertoire "assets"
File->New->Folder->Assets Folder
- Recopier .json dans le répertoire assets
- Lecture d'un fichier placé dans "assets"

```
BufferedReader br = new BufferedReader(new InputStreamReader(context.getAssets().open("data.json")));
```

Transformation du fichier Json en String

121

```
// build the string from json
```

```
try {
```

```
    BufferedReader br = new BufferedReader(new  
    InputStreamReader(getContext().getAssets().open("ballades.json")));
```

```
    StringBuilder sb = new StringBuilder();
```

```
    String line = null;
```

```
    while ((line = br.readLine()) != null) {
```

```
        sb.append(line + "\n");
```

```
    }
```

```
    String str = new String(sb.toString()) ;
```

```
} catch (IOException e) {
```

```
    e.printStackTrace();
```

```
}
```

Parsing du .json

122

```
// aller chercher la valeur du nom  
// dans l'objet bix, l'Array promenade et l'objet nom
```

```
JSONObject jsonObjConnection = new JSONObject(str);  
JSONObject jsonObjBix =  
    jsonObjConnection.getJSONObject("bix");  
JSONArray jsonA=jsonObjBix.getJSONArray("promenade");  
  
for(int i =0 ; i < jsonA.length() ; i++) {  
    JSONObject msg = (JSONObject)jsonA.get(i) ;  
    JSONObject nomObj = msg.getJSONObject("nom") ;  
    String nom = nomObj.getString("-val");  
}
```

123

Connection Internet

Les permissions: **AndroidManifest.xml**

Lancement d'une connection http

Parsing d'un fichier.json

Connection Internet

- **Rajouter dans le AndroidManifest une user permission**

`<uses-permission android:name="android.permission.INTERNET"/>`

Remarque : AndroidManifest.xml sert à décrire l'application et définir ses permissions

- **Interdiction de faire un appel réseau dans le thread principal**

-> `NetworkOnMainThreadException`

Il va donc falloir lancer une tâche parallèle, plusieurs techniques existent

Connection Internet

```

try {

    URL url = new URL(getString(R.string.ipDatabase)) ;

    // http connection

    HttpURLConnection urlConnection = (HttpURLConnection)url.openConnection();

    try {

        InputStream in = new BufferedInputStream(urlConnection.getInputStream());

        // conversion en String d'un fichier codé en UTF8. '\\A' : début du fichier

        String inputStreamString = new Scanner(in,"UTF-8").useDelimiter("\\A").next();

        // à coder ce qu'on fait de la String

        ...

    } finally {

        urlConnection.disconnect();

    }

} catch (MalformedURLException e){

    Log.d(TAGS, "Prb URL database connection");

} catch (IOException e){

    Log.d(TAGS, "Prb I/O database connection");

}

```

Les fichiers de données formatés

126

□ Utilisation de .JSON (plus concis que .xml)

```
{
  "ballades": [
    {
      "id": "12",
      "name": "Le Fantôme du Sémaphore",
      "file": "balade_canelet.mp3"
    },
    {
      "id": "19",
      "name": "Frioul",
      "file": "balade_frioul.mp3"
    }
  ]
}
```

Parsing d'un fichier JSON

127

```
JSONObject jsonObjConnection = null ;
try {
    jsonObjConnection = new JSONObject(inputStreamString );
    JSONArray jsonA=jsonObjConnection.getJSONArray( "ballades");
    String msg = jsonObjConnection.getJSONObject("msg").getString("reason");

    Log.d(MainActivity.TAGS, msg);
    ...
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}
```

Taches en arrière plan

Les Services : tâches longue démarrées, stoppées et contrôlées par d'autres composants d'application (Activité, récepteur d'intentions et autres services)

Exemple : play Music

Thread avec son **Handler**

AsyncTask

IntentService

Les Loaders

Les Alarm : déclenchement régulier ou à des heures précises

Les services

- *Les services s'exécutent de manière invisible*
- *On peut les créer les redémarrer, les arrêter, les lier à des activités*
- *Mécanisme complet et assez compliqué, il est préférable d'utiliser les autres techniques si cela est possible : `IntentService`, `AsyncTask`*
- *Doivent être déclarés dans le manifest en tant que fils de l'application*

```
<service android:enabled="true"  
    android:name="MyService"  
    android:permission="com.paad.MY_SERVICE_PERMISSION"/>
```

Démarrer un service

- *Lancer une intention explicite avec le nom du service.*

```
Intent intent = new Intent(MainActivity.this, MyService.class);
```

```
// ajouter des paramètres dans l'intent
```

```
startService(intent);
```

IntentService

- *Classe la plus pratique pour une tâche simple avec création d'un Thread asynchrone*
- *Passage des paramètres par un Intent*
- *Mise en place automatique d'une gestion de file d'attente des Intent*
- **LIMITATION :** *pas de communication retour avec la classe qui a créer l'IntentService.*
->A n'utiliser que pour un service indépendant

Classe qui hérite de IntentService

- Définit l'action à réaliser dans un thread indépendant

```
public class ConnectIntentService extends IntentService {
```

```
    /**...*/
```

```
    public ConnectIntentService() {  
        super(MainActivity.TAGS) ;  
    }
```

```
    /** méthode déclenchée par l'intention */
```

```
    @Override
```

```
    protected void onHandleIntent(Intent intent) {
```

```
        // récupération d'un paramètre de l'intent
```

```
        String user = intent.getStringExtra(MainActivity.EXT_MSG_USER);
```

```
        ...
```

Déclenchement de l'IntentService

134

- Création d'un intent, ajout de paramètres, démarrage du service

```
// création de l'intention explicite pour le service
```

```
Intent intent = new Intent(MainActivity.this, ConnectIntentService.class);
```

```
// ajout des paramètres
```

```
intent.putExtra(EXT_MSG_USER, "Bob");
```

```
...
```

```
// démarrage du service
```

```
startService(intent);
```

AsyncTask

- ❑ **Création d'un thread pour réaliser une tâche avec gestion de la fin, de la progression**
- ❑ **Possibilité d'interagir avec le thread de l'IHM**
- ❑ **Encore mieux : AsyncTaskLoader**
- ❑ **Création d'une classe qui hérite de `AsyncTask` qui surcharge**
 - ❑ `doInBackground()` // la tâche à réaliser hors thread IHM
 - ❑ `onPostExecute()` // après la tâche : dans thread IHM
 - ❑ `onPreExecute()` // facultatif : à faire dans l'IHM avant la tâche
 - ❑ `onProgressUpdate()` // facultatif : gestion progression
- ❑ **Les params peuvent être de nombre variable (mais même type)**
- ❑ **Les params passés à la tâche, la valeur retournée par celle-ci et la progression peuvent être facultatifs.**

AsyncTask

```
private class ConnectAsyncTask extends AsyncTask<String, Void, String> {
    /** executed in threads created specially for the task */
    protected String doInBackground(String...parameter) { // n params ...
        String result = null ;
        ... // ex : result = parameter[0]+" work so hard";
        return result ;
    }
    /** executed in UI thread */
    protected void onPostExecute(String result) {
        ... // ex : display result in a Toast
    }
}

// ailleurs, création de la task
new ConnectAsyncTask().execute(String1, String2, String3...); // n params
```

AsyncTask : gestion de la progression

137

```
private class ConnectAsyncTask extends AsyncTask<String, String, String> {  
    /** executed in threads created specially for the task */  
    protected String doInBackground(String...parameter) {  
  
        publishProgress( "On démarre à peine"); // call onProgressUpdate  
        ...  
        publishProgress( "On vient de passer la moitié..."); // call onProgressUpdate  
        ...  
    }  
  
    /** called by publishProgress, executed in the UI */  
    protected void onProgressUpdate(String ... msgs) {  
        Log.d(MainActivity.TAGS, msgs[0]);  
    }  
}
```

AsyncTask : résumé

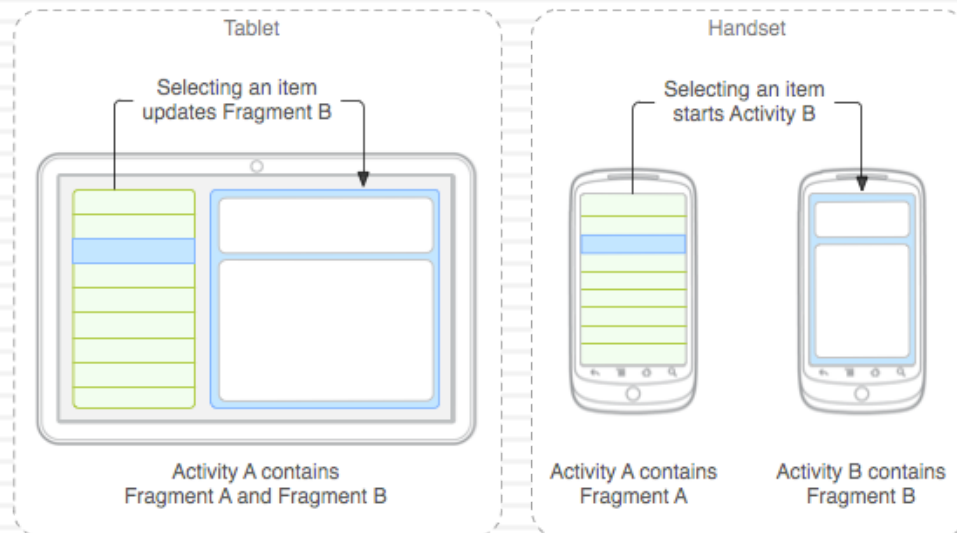
- ❑ You can specify the type of the parameters, the progress values, and the final value of the task, using generics
- ❑ The method `doInBackground()` executes automatically on a worker thread
- ❑ `onPreExecute()`, `onPostExecute()`, and `onProgressUpdate()` are all invoked on the UI thread
- ❑ The value returned by `doInBackground()` is sent to `onPostExecute()`
- ❑ You can call `publishProgress()` at anytime in `doInBackground()` to execute `onProgressUpdate()` on the UI thread
- ❑ You can cancel the task at any time, from any thread (`myTask.cancel(true);`)

Loaders

139

- ❑ Introduced in Android 3.0, loaders make it easy to asynchronously load data in an activity or fragment. Loaders have these characteristics:
- ❑ They are available to every Activity and Fragment.
- ❑ They provide asynchronous loading of data.
- ❑ They monitor the source of their data and deliver new results when the content changes.
- ❑ They automatically reconnect to the last loader's cursor when being recreated after a configuration change. Thus, they don't need to re-query their data.

Les Fragments

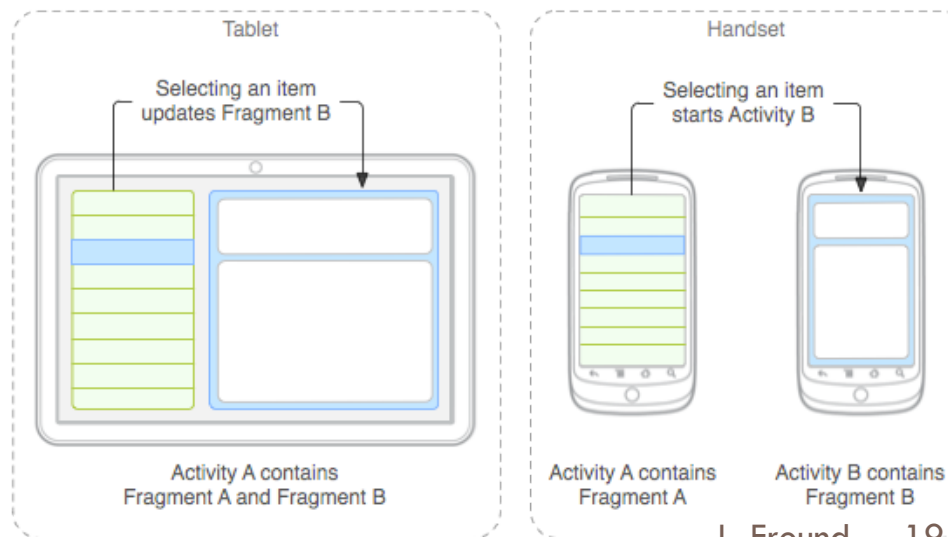


Building a Dynamic UI

141

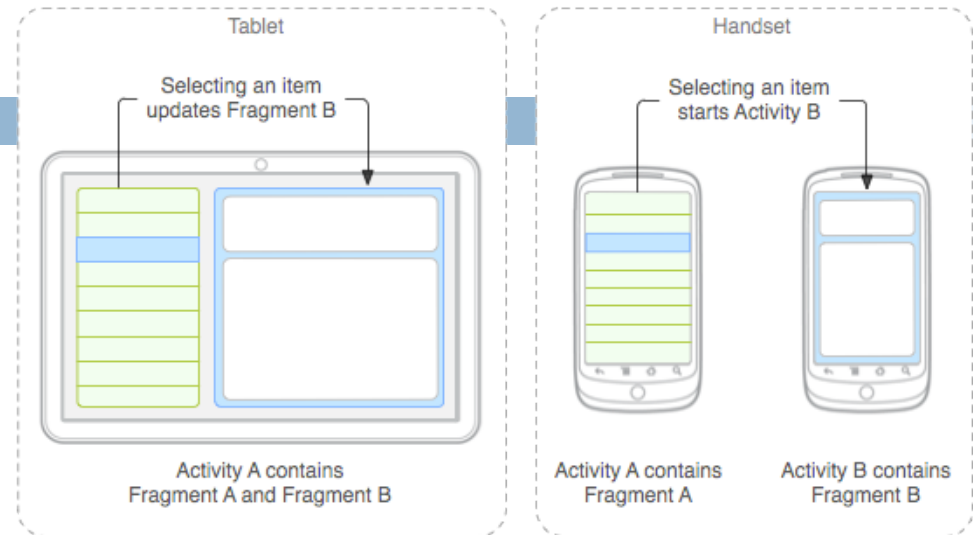
Un fragment est une sorte de sous activité que l'on peut ajouter ou supprimer statiquement ou dynamiquement dans une activité.

Les fragments ont été instaurés pour gérer les tailles des mobiles+tablettes :



Building a Dynamic UI with Fragments

142



- Activity A : "main activity"
 - ▣ Utilisation de **layouts** (.xml) différents en fonction de la taille de l'écran pour afficher 1 ou 2 fragment.
- Sur grand écran (tablette): **layout** de l'Activité A contient les 2 fragments. Activité B jamais utilisée
- Sur petit écran (mobile), **layout** Activity A contient uniquement le Fragment A (ListView). On doit lancer l'activité B pour voir le 2nd fragment
- Quand on sélectionne un élément de la liste (Frag A), si grand écran, on notifie le fragment B de la sélection, si petit écran on lance activité B

Quelques règles

143

- ❑ Un fragment ne communique jamais directement avec un autre fragment mais passe par l'activité.
- ❑ Faire attention à mettre le code dans les fragments et non dans l'activité (modularité)
- ❑ Une *interface* doit être déclarée dans chaque fragment pour que l'activité implémente un système de callback

Cycle de vie d'un fragment

144

- Cycle de vie sur le même modèle qu'une Activity

`onCreate()` : initialisation du fragment, y mettre les créations des objets

`onCreateView ()` : création de l'IHM

`onActivityCreated()` : appelé quand l'Activity mère et son IHM sont construites. **On peut alors interagir avec.**

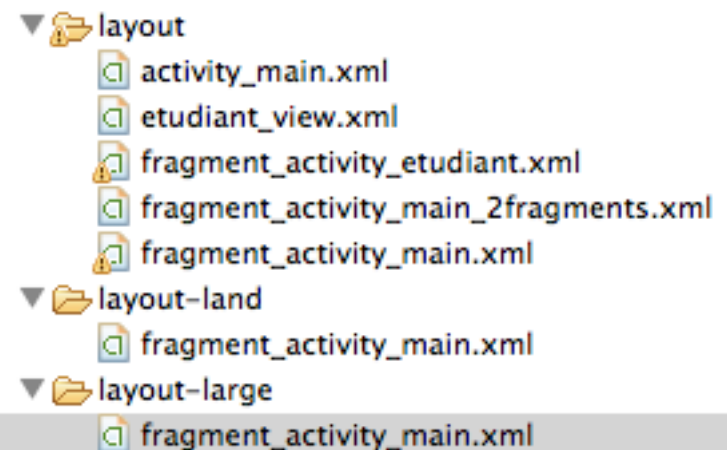
`onPause ()` : appelé quand le segment disparaît

- Les fragments suivent le cycle de vie de l'Activité mère :
pause/stop/destroy

Exemple de Fragment

145

- Exemple affichant la liste des promos d'étudiants. Le détail de la promo quand un élément est sélectionné
- Sur grand écran (tablette et écran horizontale) la liste et le détail de l'élément sont affichés (`ListFragment` et `EtudiantFragment`)
- Sur écran vertical de tél, seule la liste est affichée (`ListFragment`) quand on sélectionne, le détail est affiché en plein écran (`EtudiantFragment`).
- Nécessite 4 classes : `FragmentMainActivity`, `ListPromoFragment`, `EtudiantActivity`, `EtudiantFragment`
-
- Nécessite 6 fichiers .xml de layout



Activity englobant les Fragment (1/4)

: FragmentMainActivity

146

```
public class FragmentMainActivity extends FragmentActivity implements OnItemSelectedListener { // interface à définir
```

```
    static String POSITION_MSG = "Position";
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.fragment_activity_main);
```

```
    }
```

```
    /** callback appelé par list fragment quand on selectionne un élt */
```

```
    public void onItemSelected(int position) {
```

```
        EtudiantFragment etudiantFrag =(EtudiantFragment)getSupportFragmentManager().findFragmentById(R.id.etudiantFrag);
```

```
        if (etudiantFrag == null || ! etudiantFrag.isInLayout()) { // EtudiantFragment is not in the layout (handset layout),
```

```
            // so start EtudiantActivity and pass it the info about the selected item
```

```
            Intent intent = new Intent(this, EtudiantActivity.class);
```

```
            intent.putExtra(POSITION_MSG, position);
```

```
            startActivity(intent);
```

```
        } else // EtudiantActivity is in layout (tablet layout), tell the fragment to update
```

```
            etudiantFrag.updateContent(position);
```

```
    }
```

```
}
```

- Affichage de 1 ou 2 fragments en fct orientation et taille d'écran
- Création EtudiantActivity si nécessaire



Activity A contains
Fragment A and Fragment B

Layouts englobant les 2 Fragments

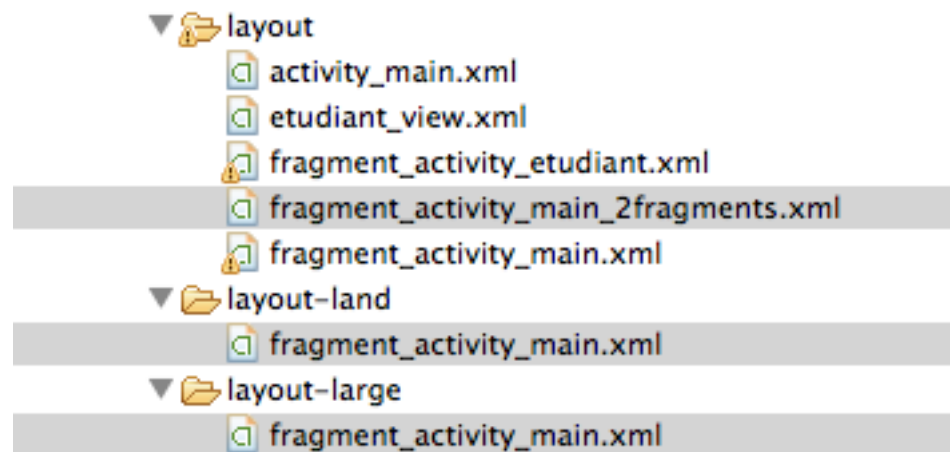
147

- pour avoir 2 layouts identiques : `layout-land/layout-large`
 - ▣ création d'un fichier layout unique : `layout/fragment_activity_main_2fragments.xml`
 - ▣ Création de 2 fichiers incluant le précédent
 - `layout-land/fragment_activity_main.xml`
 - `layout-large/fragment_activity_main.xml`

Eclipse : pour créer les xml, *new...> Other > XML Android File*



Activity A contains
Fragment A and Fragment B

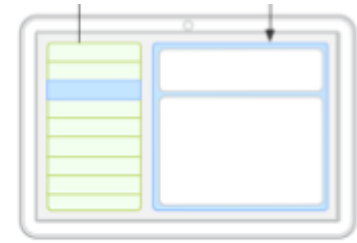


Layouts englobant les 2 Fragments

: `layout/fragment_activity_main_2fragments.xml`

148

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:baselineAligned="false"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="ismin.M_S13.trombismo.ListPromoFragment" // !!! CHANGER
        android:id="@+id/listPromo"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
    <fragment android:name="ismin.M_S13.trombismo.EtudiantFragment"// !!! CHANGER
        android:id="@+id/etudiantFrag"
        android:layout_weight="2"
        android:layout_width="0dp"
        android:layout_height="match_parent" />
</LinearLayout>
```



Activity A contains
Fragment A and Fragment B

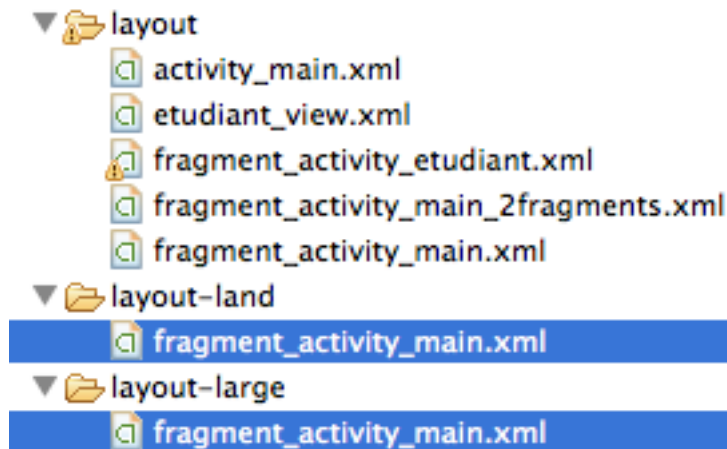


Layouts incluant *fragment_activity_main_2fragments.xml*

layout-land/fragment_activity_main.xml

layout-large/fragment_activity_main.xml

149



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

```
<merge>
```

```
    <include layout="@layout/fragment_activity_main_2fragments"/>
```

```
</merge>
```

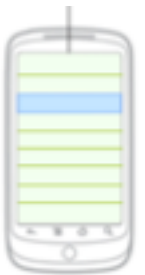
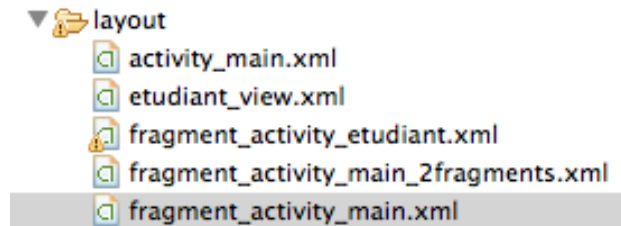
Layout pour un téléphone en portrait

layout/fragment_activity_main.xml

150

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:baselineAligned="false"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <fragment
        android:id="@+id/listPromo"
        android:name="ismin.M_S13.trombismo.ListPromoFragment" » // !!! CHANGER package

        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="@dimen/activity_left_margin"
        android:layout_marginRight="@dimen/activity_right_margin"
        android:layout_weight="1"
        tools:context=".FragmentMainActivity"
        tools:layout="@android:layout/list_content" />
</LinearLayout>
```



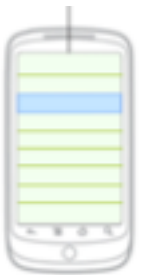
Activity A contains
Fragment A

Fabrication d'une ListFragment (2/4)

ListPromoFragment

151

- Classe spécialement conçue pour visualiser les listes dans un fragment
 - Équivalent des ListActivity (Activité avec ListView) pour les fragments
- Exemple : visualisation d'un tableau de String sous forme de liste (cf fig)
- Doit définir une interface pour décrire la méthode que l'activity doit exécuter quand on clique
- La FragmentMainActivity gère le redispatching aux autres fragments de la gestion de l'évènement



Activity A contains
Fragment A

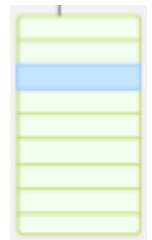
Fabrication d'une ListFragment (2/4)

ListPromoFragment

152

```
public class ListPromoFragment extends ListFragment{
    OnItemSelectedListener mListener; // I listener
    // Container Activity must implement this interface
    public interface OnItemSelectedListener {
        public void onItemSelected(int numSelected);
    }
    @Override
    /** Creation of the View */
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        setListAdapter(new ArrayAdapter<String>(getActivity(),
            android.R.layout.simple_list_item_activated_1,
            nameUsers)); // ToDo avec AsyncTask dans onCreate
        return super.onCreateView(inflater, container, savedInstanceState);
    }
    ...
}
```

- Fragment pour listes
- Utilisation ListAdapapter
- Définition une interface pour gestion click par Activity
- MainActivity gère le redispatching aux autres fragments

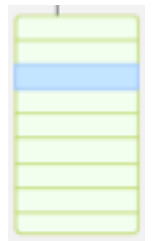


Fabrication d'une ListFragment (2/4)

ListPromoFragment

153

```
...  
  
public void onAttach(Activity activity) {  
    super.onAttach(activity);  
    try {  
        mListener = (OnItemSelectedListener) activity;  
    } catch (ClassCastException e) {  
        throw new ClassCastException(activity.toString() + " must implement  
OnItemSelectedListener");  
    }  
}  
  
@Override  
public void onListItemClick(ListView l, View v, int position, long id) {  
    mListener.onItemSelected(position); // Send the position to the host activity  
}  
  
}
```



Fabrication d'un Fragment (3/4)

EtudiantFragment

154

```
public class EtudiantFragment extends Fragment {
    private TextView logView ;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.etudiant_view, container,
false);
        logView =(TextView)view.findViewById(R.id.logView);// store view for
later
        return view;
    }

    /** update the content corresponding to the selected index */
    void updateContent(int position) {
        logView.setText("Selected :" +position);
    }
}
```



Layout de d'étudiant :

layout/etudiant_view.xml

155

```
<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView

        android:id="@+id/titleView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="@style/port_bigTxt"
        android:text="@string/msg_title" />

    <TextView

        android:id="@+id/logView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/titleView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="112dp"
        android:text="@string/etudiant_nom" />

</RelativeLayout>
```



Fabrication de l'activité du Fragment (4/4)

EtudiantActivity

156

```
public class EtudiantActivity extends FragmentActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if(getResources().getConfiguration().orientation==
Configuration.ORIENTATION_LANDSCAPE){
            // If landscape mode, we don't need this activity.
            finish();
            return;
        } else if (savedInstanceState == null) {
            setContentView(R.layout.fragment_activity_etudiant);
            EtudiantFragment etudiantFragment =
(EtudiantFragment)getSupportFragmentManager().findFragmentById(R.id.etudiantFrag);
            int indexSelected
=getIntent().getIntExtra(FragmentMainActivity.POSITION_MSG,0);
            etudiantFragment.updateContent(indexSelected);
        }
    }
}
```



Activity B contains
Fragment B

Layout de de l'activity de l'étudiant :

layout/fragment_activity_etudiant.xml

157

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:baselineAligned="false"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >

    <fragment
        android:id="@+id/etudiantFrag"
        android:name="ismin.M_S13.trombismo.EtudiantFragment" // !!! CHANGER
package
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="@dimen/activity_left_margin"
        android:layout_marginRight="@dimen/activity_right_margin"
        tools:context=".EtudiantActivity"
        android:layout_weight="1"/>
</LinearLayout>
```



Activity B contains
Fragment B

Quelques classes héritées de fragments

158

- DialogFragment : displays a floating dialog. Using this class to create a dialog is a good alternative to using the dialog helper methods in the Activity class, because you can incorporate a fragment dialog into the back stack of fragments managed by the activity, allowing the user to return to a dismissed fragment.
- ListFragment : displays a list of items that are managed by an adapter (such as a SimpleCursorAdapter), similar to ListActivity. It provides several methods for managing a list view, such as the onItemClick() callback to handle click events.
- PreferenceFragment : displays a hierarchy of Preference objects as a list, similar to PreferenceActivity. This is useful when creating a "settings" activity for your application.

Gestion dynamique des fragments

159

Il est possible de gérer dynamiquement la création (*add*), la destruction (*remove*), le changement de fragments (*replace*), en utilisant le **FragmentManager**, en démarrant une transaction et en faisant un `commit()` après l'action. Il est possible de rajouter des animations à ces transactions (*setTransition* ou *setCustomAnimations*)

```
FragmentManager ft = getSupportFragmentManager().beginTransaction();  
Ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN) ;  
ft.replace(R.id.my_idFrag, new MyFragment());  
ft.commit();
```

Infos sur les fragments

160

Un fragment n'a pas nécessairement d'UI

Un fragment peut utiliser les transitions

TP5 : connexion et Fragments

Utilisez les fragments pour gérer l'orientation, landscape-paysage, seule la liste apparaisse, ou bien les 2 fragments.

-> Remplacer `ListUserActivity` par les fragments.

(ne pas oublier de déclarer les `Activity` dans `AndroidManifest`)

Afficher le nom et prénom,... dans `EtudiantFragment`

Stockage des données (1 / 2)

Préférences partagées

Etat de l'interface

Fichiers

Base de données

Préférences partagées

163

- SharedPreferences : paires nom/valeur dans le contexte d'application
- Créer, éditer et sauvegarder

```
Final static String LOGIN_PREFS = "login_param" ;  
// création preferences ou ouverture si existe  
SharedPreferences sharedPref =  
    getSharedPreferences(LOGIN_PREFS, Activity.MODE_PRIVATE);  
SharedPreferences.Editor editor = sharedPref.edit();  
editor.putString("user", "bob");  
editor.commit(); // validate changes
```

Récupérer les valeurs

164

- Récupérer la valeur de la clef user

// preferences

```
SharedPreferences sharedPref =  
    getSharedPreferences(LOGIN_PREFS, Activity.MODE_PRIVATE);  
// get user value, "" if undefined  
final String userPref = sharedPref.getString("user", "");
```

- Tester si une clef existe

aller voir dans la doc de l'API la méthode *contains*

Stockage des données (2/2)

Préférences partagées

Etat de l'interface

Fichiers

Base de données

Base de données, moteur SQLite

166

- => **SQLiteOpenHelper** : classe abstraite à implémenter ; modèle pour la création, l'ouverture et la Mise à Jour
- La création est codée dans **SQLiteOpenHelper**
- Une fois crée on demande la **SQLiteDatabase** au **SQLiteOpenHelper** soit pour la lecture soit pour l'écriture
- **ContentValues** : lignes à insérer dans la base
- **Cursor** : pointeurs vers les résultats d'une requête

BalladesDBOpenHelper (1/2)

167

```
/** classe de creation de la base */
public class BalladesDBOpenHelper extends SQLiteOpenHelper {
    private static BalladesDBOpenHelper db ;
    private final static String DATABASE_NAME = "balladesDataBase.db" ;
    final static String DATABASE_TABLE = "Ballades" ;
    private final static int DATABASE_VERSION = 1 ;

    // def des champs de la base
    public static final String COLUMN_ID = "_id"; // id
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_PLAY = "play_file";

    // Database creation sql statement
    private static final String DATABASE_CREATE = "create table "
        + DATABASE_TABLE + "(" +
        COLUMN_ID + " integer primary key autoincrement, " +
        COLUMN_NAME + " text not null, " +
        COLUMN_PLAY + " text not null);";
}
```

...

BalladesDBOpenHelper (2/2)

168

```
public static BalladesDBOpenHelper getBalladesDBOpenHelper(Context context) { //Singleton
    if(db==null)
        db = new BalladesDBOpenHelper(context) ;
    return db ;
}

private BalladesDBOpenHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);

}

/** creation de la base, est appelée quand on cherche à lire ou écrire dans la base*/
@Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(DATABASE_CREATE);
    }

/** update à une nouvelle version de la base */
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.w(BalladesDBOpenHelper.class.getName(),
        "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
    onCreate(db);
}
}
```

Insertion dans la base : ContentValues

169

```
// open database to write => appel à onCreate
SQLiteDatabase db =
BalladesDBOpenHelper.getBalladesDBOpenHelper(
    getBaseContext()).getWritableDatabase() ;
// getBaseContext ou getContext/this/getActivity
// build ContentValues with data
ContentValues values = new ContentValues();
values.put(BalladesDBOpenHelper.COLUMN_NAME, "ballade1");
values.put(BalladesDBOpenHelper.COLUMN_PLAY, "ballades1.mp3"
);

// insert in db
long id =db.insert(BalladesDBOpenHelper.DATABASE_TABLE,
    null, values);
```

Récupération des data: Cursor

170

```
String [] allColumns = {BalladesDBOpenHelper.COLUMN_ID,
                        BalladesDBOpenHelper.COLUMN_NAME,
                        BalladesDBOpenHelper.COLUMN_PLAY} ;

// open database to read
SQLiteDatabase dbR =
BalladesDBOpenHelper.getBalladesDBOpenHelper(getBaseContext()).getReadableDatabase() ;

Cursor cursor = dbR.query(BalladesDBOpenHelper.DATABASE_TABLE,
                          allColumns, null, null, null, null, null);

cursor.moveToFirst();
while (!cursor.isAfterLast()) {
    Log.d(TAG, "id = "+cursor.getLong(0) +
        ", name = "+cursor.getString(1) +
        ", play= "+cursor.getString(2));
    cursor.moveToNext();
}

// make sure to close the cursor
cursor.close();
```

Update d'une valeur

171

```
String [] allColumns = {BalladesDBOpenHelper.COLUMN_ID,  
    BalladesDBOpenHelper.COLUMN_NAME,  
    BalladesDBOpenHelper.COLUMN_PLAY} ;  
  
// open database to read  
  
SQLiteDatabase db =BalladesDBOpenHelper.getBalladesDBOpenHelper(getBaseContext()).getWritableDatabase();  
    ContentValues values = new ContentValues();  
values.put(BalladesDBOpenHelper.COLUMN_NAME, " new name ");  
db.update(BalladesDBOpenHelper.DATABASE_TABLE, values, BalladesDBOpenHelper.COLUMN_ID + "=" + rowId, null);  
  
// make sure to close the cursor  
cursor.close();
```

TP : Base de données

Créer la classe `BalladesDBOpenHelper` pour gérer la création de la base qui contient nom de ballade, description...

Dans `onCreate` de `MainActivity`,

Appeler `loadJson`

- ouverture de la base

- pour chaque nom NON-trouvé dans le json, l'insérer dans la base

Afficher dans les log la base entière

Créer la `listView` en lisant directement dans la base

Ajouter un champ CMT dans la base.

Quand ce champ contient « new » on affiche une icône new

Sinon on affiche une note de musique

TP : connexion

- Plutôt que d'aller lire dans les 'assets ', aller lire le .json

<http://www.laurent-freund.fr/cours/android/ballades> [www.json](#)

Cela nécessite l'utilisation d'un service.

- Rajouter le mediaPlayer pour jouer la musique récupérée aux url (.json)

Content Provider

Pour rendre les données disponibles à d'autres activités (ou à son Activity), Android fournit le mécanisme de fournisseur de contenu. Il permet de faire une interface d'accès propre
fournit des outils supplémentaires

[CursorLoader](#) qui crée un Loader
moteur de recherche

Exemple : les contacts, est un [Content Provider](#)

Création d'un Content Provider

175

1. Création d'une classe qui hérite de **ContentProvider**
2. Implémentation de toutes les méthodes de la classe abstraite **ContentProvider**
 - Création de la base
 - Implémentation des requêtes
3. Déclaration du provider dans **AndroidManifest** pour le rendre visible à toutes les Activity

1) AnciensContentProvider extends Content Provider (1 / ...)

176

Héritage d'une classe abstraite

-> surcharge obligatoire de **onCreate**, query, update, delete, insert, getType

```
public class AnciensContentProvider extends ContentProvider {  
...  
  
/**  
 * Create the database of the Anciens with singleton for latter  
 * @see android.content.ContentProvider#onCreate()  
 */  
@Override  
public boolean onCreate() {  
    //use a singleton : méthode à créer dans AnciensDBOpenHelper  
    database = AnciensDBOpenHelper.getAnciensDBOpenHelper(getContext());  
    return false;  
}  
...
```

1) AnciensContentProvider extends Content Provider (2/...)

177

□ Exposition de l'URI de l'ensemble des éléments

```
private static final String AUTHORITY="fr.ismin.M_S13.provider_anciens"; // == MANIFEST
private static final String BASE_PATH = "elements";
public static final Uri CONTENT_URI = Uri.parse(
    "content://" + AUTHORITY + "/" + BASE_PATH);
```

Info : les éléments seront accessibles individuellement avec
"content://" + AUTHORITY + "/" + BASE_PATH + "/1"

Création d'un `URIMatcher` pour reconnaître le type d'URI

```
private static final int ALLROWS= 1;
private static final int SINGLE_ROW = 2;
private static final UriMatcher uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    static {
        uriMatcher.addURI(AUTHORITY, BASE_PATH, ALLROWS);
        uriMatcher.addURI(AUTHORITY, BASE_PATH + "/#", SINGLE_ROW );
    }
```

1) AnciensContentProvider définition query() (3/...)

178

/* execution de la requete SQL avec filtrage de l'URI en fonction de l'uriMatcher...

```
public Cursor query(Uri uri, String[] projection, String selection,
                    String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();
    queryBuilder.setTables(AnciensDBOpenHelper.DATABASE_TABLE);

    switch (uriMatcher.match(uri)) {
        case ALLROWS:
            break;
        case SINGLE_ROW:
            // adding the ID to the original query
            queryBuilder.appendWhere(AnciensDBOpenHelper.COLUMN_ID + "=" +
                                    uri.getLastPathSegment());
            break;
        default:
            throw new IllegalArgumentException("Unknown URI: " + uri);
    }
    SQLiteDatabase db = database.getWritableDatabase();
    Cursor cursor = queryBuilder.query(db, projection, selection,
                                       selectionArgs, null, null, sortOrder);

    return cursor;
}
```

1) AnciensContentProvider définition getType()

-> MIME (3/...)

179

```
/**
 *return the good MIME type
 * @see android.content.ContentProvider#getType(android.net.Uri)
 */
@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)) {
        case ALLROWS:
            return "vnd.android.cursor.dir/vnd.ismin.elemental" ;
        case SINGLE_ROW:
            return "vnd.android.cursor.item/vnd.ismin.elemental"
;
        default:
            throw new IllegalArgumentException("Unknown URI: " +
uri);
    }
}
```

1) AnciensContentProvider définition delete() (3/...)

180

```
public int delete(Uri uri, String selection, String[] selectionArgs) {
    SQLiteDatabase sqlDB = database.getWritableDatabase() ;
    String rowId = uri.getLastPathSegment() ;
    int rowsDeleted = 0;
    switch (uriMatcher.match(uri)) {
        case ALLROWS:
            rowsDeleted=sqlDB.delete(AnciensDBOpenHelper.DATABASE_TABLE, selection, selectionArgs);
            break;
        case SINGLE_ROW:
            if (TextUtils.isEmpty(selection)) {
                rowsDeleted = sqlDB.delete(AnciensDBOpenHelper.DATABASE_TABLE,
                    AnciensDBOpenHelper.COLUMN_ID + "=" + rowId, null);
            } else {
                rowsDeleted = sqlDB.delete(AnciensDBOpenHelper.DATABASE_TABLE,
                    AnciensDBOpenHelper.COLUMN_ID + "=" + rowId
                    + " and " + selection, selectionArgs);
            }
            break;
        default:
            throw new IllegalArgumentException("Unknown URI: " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return rowsDeleted;
}
```


1) AnciensContentProvider définition insert() (3/...)

181

```
/** insert one element*/
public Uri insert(Uri uri, ContentValues values) {
    SQLiteDatabase sqlDB = database.getWritableDatabase();
    long id = 0;
    switch (uriMatcher.match(uri)) {
        case ALLROWS:
            id=sqlDB.insert(AnciensDBOpenHelper.DATABASE_TABLE,
                           null, values);
            break;
        default:
            throw new IllegalArgumentException("Unknown URI:" +
uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return Uri.parse(BASE_PATH + "/" + id);
}
```

1) AnciensContentProvider définition update() (3/...)

182

```
/** update element(s) */
public int update(Uri uri, ContentValues values, String selection,
    String[] selectionArgs) {
    SQLiteDatabase sqlDB = database.getWritableDatabase();
    int rowsUpdated = 0;
    switch (uriMatcher.match(uri)) {
    case SINGLE_ROW:
        String rowId = uri.getLastPathSegment();
        selection = AnciensDBOpenHelper.COLUMN_ID + "=" +
            (!TextUtils.isEmpty(selection) ? " AND (" + selection + ')':"");
    default :
        break;
    }
    int updateCount = sqlDB.update(AnciensDBOpenHelper.DATABASE_TABLE,
        values, selection, selectionArgs);
    getContext().getContentResolver().notifyChange(uri, null); // notify
    return updateCount;
}
```

2) Déclaration AndroidManifest

183

- Définition de l'application : `AnciensContentProvider`
- Définition de l'autorité du fournisseur en reprenant l'esprit du package pour le nom :
`fr.ismin.M_S13.provider_anciens`

```
<application
    <!-- A placer après les Activity -->
    <provider
        android:name=".AnciensContentProvider"
        android:authorities= "fr.ismin.M_S13.provider_anciens">
    </provider>
</application>
```

Se servir du **ContentProvider**:

ajout d'une donnée

184

```
/** ajout d'un element */  
// get the content resolver  
ContentResolver cr = getContentResolver() ;  
Log.d(TAGS, "CProv="+cr.getType(AnciensContentProvider.CONTENT_URI) );  
  
ContentValues values = new ContentValues();  
values.put(AnciensDBOpenHelper.COLUMN_NAME, "Sinclar");  
values.put(AnciensDBOpenHelper.COLUMN_FORNAME, "bob");  
Uri uri = cr.insert(AnciensContentProvider.CONTENT_URI, values);
```

Accéder au contenu :

récupération des données

185

```
String []allColumns = {AnciensDBOpenHelper.COLUMN_ID,  
AnciensDBOpenHelper.COLUMN_NAME, AnciensDBOpenHelper.COLUMN_FORNAME};  
  
// recup d'un cursor  
Cursor cursor = cr.query(AnciensContentProvider.CONTENT_URI,allColumns, null,  
null, null);  
Log.d(TAGS, "cursor="+cursor);  
cursor.moveToFirst();  
while (!cursor.isAfterLast()) { // parcours de tous les elts  
    Log.d(TAGS, "id = "+cursor.getLong(0)+  
        ", name = "+cursor.getString(1)+  
        ", forname = "+cursor.getString(2));  
    cursor.moveToNext();  
}  
// make sure to close the cursor  
cursor.close();
```

TP : Content Provider

Créer le [ContentProvider](#) et l'utiliser

Pour cela, on pourra essayer d'ajouter un élément dans la base et d'afficher le contenu de toute la base en utilisant le content provider

Attention : tous les exemples des transparents utilisent « Anciens » au lieu de « Ballades » cela vous obligera à rentrer un peu dans le code de la base de donnée et du provider

Recherche dans un document

Barre de recherche

Dans la barre (étonnant non ?)

Vue de recherche : **le top**

Widget à placer dans une activity, ou dans la barre

Boite de recherche

Widget de l'écran d'accueil d'Android

Ergonomie

Recherche vocale

Suggestion par rapport aux dernières recherches

Corrélation avec les données de la base

Étapes

188

1. Créer le fichier des Méta-données (`searchable.xml`)
2. Insérer l'outil de recherche dans une vue
3. Déclarer la Searchable Activity
(Optionnel) Suggestions de recherche
4. Recevoir la requête de recherche
5. Chercher les données
6. Afficher les résultats



1) Meta-données à définir

189

- Configuration du search dialog ou widget :
 - ▣ Texte *search_hint*, voice search, search suggestion...

- Créer un fichier *res/xml/searchable.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<searchable
xmlns:android="http://schemas.android.com/apk/res/androi
d">
    android:label="@string/app_name"
    android:hint="@string/search_hint"
</searchable>
```

2) Ajout du Widget de Recherche

190

- Ajout du Widget dans la vue_souhaitée.xml



```
<SearchView
```

```
    android:id="@+id/searchView"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentRight="true"
```

```
    android:layout_alignParentTop="true" >
```

```
</SearchView>
```

(Optionnel) Suggestions de recherche

191

- Modifier search.xml : `android:searchSuggestAuthority`

```
<?xml version="1.0" encoding="utf-8"?>
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:label="@string/app_label"
    android:hint="@string/search_hint »

    android:searchSuggestAuthority="com.example.MyCustomSuggestionProvider">

</searchable>
```

(Optionnel) Suggestions de recherche

192

...

3) Déclarer la Searchable Activity

193

- Activité qui gère la recherche et qui l'affiche -> modif du Manifest
 1. Declare the activity to accept the ACTION_SEARCH intent, in an <intent-filter> element.
 2. Specify the searchable configuration to use, in a <meta-data> element.

- Manifest

<application ... >

<activity android:name=".SearchableActivity" >

<intent-filter>

<action android:name="android.intent.action.SEARCH" />

</intent-filter>

<meta-data android:name="android.app.searchable"
android:resource="@xml/searchable"/>

</activity>

...

</application>

4) Recevoir la requête de recherche

194

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.search);  
  
    // Get the intent, verify the action and get the query  
    Intent intent = getIntent();  
    if (Intent.ACTION_SEARCH.equals(intent.getAction())) {  
        String query = intent.getStringExtra(SearchManager.QUERY);  
        doMySearch(query);  
    }  
}
```

Recupération des key
Affichage de la map
Ajout d'éléments sur la carte

Créer une nouvelle activity Google Map

198

□ Créer l'activity

File/New/Google/Google Map Activity

□ Récupérer la Google Map API Key

-> Dans `res/values/goog_maps_api.xml`, cliquez sur le lien automatiquement généré

Ex :

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=F6:F2:9E:E3:BE:60:3F:84:9F:D4:E4:C3:49:DF:7C:F1:44:A2:C8:FD%3Bfr.grenouille.balladessonores.db.ballades

▣ Choisir *Créer un projet*

▣ Cliquer sur *créer* pour générer votre clef, exemple
`AlzaSyDU3LIKUNqFs9jru0vJLqDgChKnc7cNgxw`

□ il faut mettre la bonne clef dans **google_maps_api.xml**
À la place de `YOUR KEY HERE`

Manifest :Insertion de la clef et des permissions

199

□ Fait **automatiquement** dans le Manifest:

1. Ajout de la Key

```
<application
```

```
<meta-data
```

```
    android:name="com.google.android.geo.API_KEY"
```

```
    android:value="@string/google_maps_key"/>
```

2. Ajout des permissions pour un accès fin à la position

```
<uses-permission
```

```
    android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-permission
```

```
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Création de la Map

200

- Le Wizard a automatiquement crée une **Activity** Map
 - *activity_map.xml*
 - *MapsActivity.java*

- Dans notre cas, Il faut le transformer en **Fragment**
 - Le plus simple est de créer **MapsFragment** : un Blank Fragment **sans créer layout XML** et transformer son .java

[1/2] Ajout d'une Map dans 1 Fragment (au lieu d'Activity)

201

```
import android.support.v4.app.Fragment;

public class MapFragment extends Fragment implements OnMapReadyCallback {
    // onCreateView qui retourne une map
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.activity_maps, container, false);
    }

    private final static LatLng GARDANNE = new LatLng(43.455669, 5.470648999999998);

    @Override
    public void onActivityCreated (Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        SupportMapFragment supportMapFragment = ((SupportMapFragment)
        getChildFragmentManager().findFragmentById(R.id.map));
        supportMapFragment.getMapAsync(this);
    }
}
```

[2/2] Ajout d'une Map dans 1 Fragment (au lieu d'Activity)

202

```
@Override
public void onMapReady(GoogleMap googleMap) {
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(GARDANNE), 20000, null);

    if (ActivityCompat.checkSelfPermission(getActivity().getApplicationContext(),
        android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(getContext(), android.Manifest.permission.ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        return;
    }
    else
        googleMap.setMyLocationEnabled(true);
}
```

Quelques modifs

203

- *activity_maps.xml*, supprimer
tools:context="balades.isen16ballades.MapsActivity"
- MainActivity, ajouter le listener d'interaction avec le fragment :

Class MainActivity ... implements `FragmentManager.OnFragmentInteractionListener`,...

Dans son tél, donner les autorisations pour la localisation

Config. MapFragment or a MapView par XML

204

- **mapType**. This allows you to specify the type of map to display. Valid values include: `none`, `normal`, `hybrid`, `satellite` and `terrain`.
- **cameraTargetLat**, **cameraTargetLng**, **cameraZoom**, **cameraBearing**, **cameraTilt**. These allow you to specify the initial camera position. See [here](#) for more details on Camera Position and its properties.
- **uiZoomControls**, **uiCompass**. These allow you to specify whether you want the zoom controls and compass to appear on the map. See [UiSettings](#) for more details.
- **uiZoomGestures**, **uiScrollGestures**, **uiRotateGestures**, **uiTiltGestures**. These allow you to specify which gestures are enabled/disabled for interaction with the map. See [UiSettings](#) for more details.
- **zOrderOnTop**. Control whether the map view's surface is placed on top of its window. See [SurfaceView.setZOrderOnTop\(boolean\)](#) for more details. Note that this will cover all other views that could appear on the map (e.g., the zoom controls, the my location button).
- **useViewLifecycle**. Only valid with a **MapFragment**. This attribute specifies whether the lifecycle of the map should be tied to the fragment's view or the fragment itself. See [here](#) for more details.

In order to use these custom attributes within your XML layout file, you must first add the following namespace declaration (you can choose any namespace, it doesn't have to be `map`):

```
xmlns:map="http://schemas.android.com/apk/res-auto"
```

Exemple configuration activity_maps.xml

205

```
□ <?xml version="1.0" encoding="utf-8"?>
  <fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    map:cameraBearing="112.5"
    map:cameraTargetLat="43.296482"
    map:cameraTargetLng="5.369779999999992"
    map:cameraTilt="30"
    map:cameraZoom="13"
    map:mapType="normal"
    map:uiCompass="false"
    map:uiRotateGestures="true"
    map:uiScrollGestures="true"
    map:uiTiltGestures="true"
    map:uiZoomControls="true"
    map:uiZoomGestures="true"/>
```

Exemple **simple** dans une **Activity** : déplacement Caméra

206

Immédiat : `moveCamera( )`, animation : `animateCamera()`

```
public class MapAnciensActivity extends Activity {
    private GoogleMap mMap;
    private final static LatLng GARDANNE = new LatLng(43.455669,5.470648999999998) ;

    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_map);

        mMap= ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map)).getMap();
        if (mMap != null) {
            mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
            mMap.animateCamera(CameraUpdateFactory.newLatLng(GARDANNE),2000, null);
        }
    }
}
```


Placement d'un Marker

207

□ Du marqueur simple

```
mMap.addMarker(new MarkerOptions().position(GARDANNE));
```

□ Au plus custom

```
googleMap.addMarker(new MarkerOptions()  
    .position(GARDANNE)  
    .title("Gardanne")  
    .snippet("Population: beaucoup trop")  
    .alpha(0.7f)  
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.lampe_transp))  
);
```

□ Possibilité de rajouté des listener de click et de Drag

```
GoogleMap.setOnMarkerClickListener(OnMarkerClickListener), GoogleMap.setOnMarkerDragListener
```

Dessin de Shape

208

□ Polyline, Polygon, Circle

// exemple Cercle

```
CircleOptions circleOptions = new CircleOptions()
    .center(GARDANNE)
    .fillColor(Color.parseColor("#440000DD"))
    .radius(500); // In meters
```

// Get back the mutable Circle

```
Circle circle = mMap.addCircle(circleOptions);
```

Remarque : pour dessiner une image collée à la map et qui varie en fonction du zoom, de la rotation, on utilisera une [GroundOverlay](#)

Location Data

209

- Il est possible d'utiliser le Location Layer pour se positionner en appuyant sur l'icône

```
mMap.setMyLocationEnabled(true);
```

Par contre la version de la target doit être inférieure ou égale à 22 car sinon il faut demander dynamiquement les permissions à l'utilisateur

Utilisation du géocodeur

210

- Le **Geocoder** fait partir des Google play Services
- Géocodage Avant : lat,long -> add
- Géocodage Arrière: add -> lat,long

Utilisation du Add -> lat, long

211

- A placer dans une AsyncTask

```
List <Address> locations = null ;  
// utilisation du geocodage  
Geocoder geocoder = new Geocoder(getApplicationContext(),  
    Locale.getDefault());  
try {  
    locations = geocoder.getFromLocationName("15 rue paradis,  
marseille", 10);  
    Log.d(MainActivity.TAGS, "["+location.getLatitude()+ ","  
        +location.getLongitude()+"]");  
} catch (IOException e){  
    Log.e(MainActivity.TAGS, "IO Exception", e);  
}
```

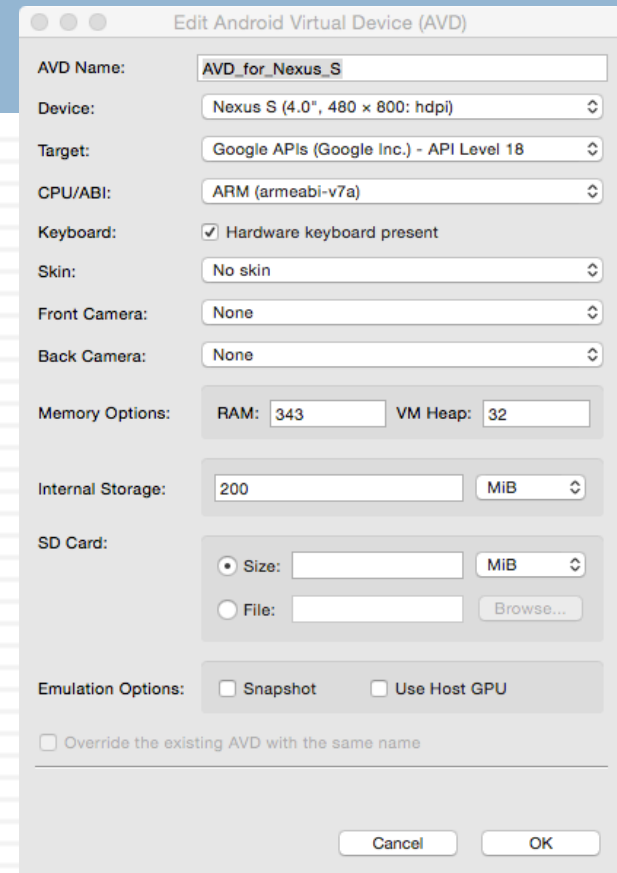
A partir du TP précédent, faire une activity qui contient une carte.

Rajouter les marqueurs des ballades

Rajouter les interactions quand on clique

Rajouter les map de chaque ballade

Pour la simulation sur Emulateur
Configurer l'AVD pour les Google API



WallPaper

Créer le projet WallPaperLogolsmin,
ne pas créer d'activité

Ajouter dans res/drawable l'image souhaitée

Créer la classe WallPaperLogolsmin extends **WallpaperService**

```
public class WallPaperLogolsmin extends WallpaperService {  
    /** Create The Engine */  
    public Engine onCreateEngine() {  
        return new WallpaperEngine();  
    }  
}
```


Xml pour la description

215

Créer `res/xml/wallpaperdescr.xml` qui servira à décrire le wallpaper :

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

```
<wallpaper
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:thumbnail="@drawable/ic_launcher"
```

```
    android:description="@string/wallpaper_description"/>
```

Rajouter dans `values/strings.xml`

```
<string name="wallpaper_description">WallPaper Logolsmin (2013)</string>
```

Modification du Manifest.xml

- Rajouter après `android:theme="@style/AppTheme" >`

```

<service
    android:label="@string/wallpaper_service"
    android:name=".WallPaperLogosmin"
    android:permission="android.permission.BIND_WALLPAPER">
        <intent-filter>
            <action
                android:name="android.service.wallpaper.WallpaperService"></action>
            </intent-filter>
            <meta-data
                android:name="android.service.wallpaper"
                android:resource="@xml/wallpaperdescr"></meta-data>
        </service>

```
- Rajouter dans `values/strings.xml`

```

<string name="wallpaper_service">Logo lsmín</string>

```

Rajouter la classe interne : WallPaperEngine

```

/** Engine for displaying WallPaper */
private class WallpaperEngine extends Engine {
    private Bitmap lampPic = BitmapFactory.decodeResource(getResources(), R.drawable.lampe_transp);

    /** Get the Canvas and draw*/
    public void onSurfaceChanged(SurfaceHolder holder, int format, int width, int height) {
        super.onSurfaceChanged(holder, format, width, height);

        Canvas canvas = null;
        try {
            canvas = holder.lockCanvas();
            if (canvas != null) {
                canvas.drawColor(Color.LTGRAY);
                canvas.drawBitmap(lampPic, (width - lampPic.getWidth())/2,
(height - lampPic.getHeight())/2, null);
            }
        } finally {
            if (canvas != null)
                holder.unlockCanvasAndPost(canvas);
        }
    }
}

```

Faire une image animée

Rajouter un Handler, un Runnable, un onTouchEvent

4 conseils pour développer vos ventes mobiles

219

1. Redonnez du **contrôle** au consommateur

Il est très difficile de proposer de nombreux produits à la vue du consommateur sachant que l'écran d'un téléphone est relativement petit. Les utilisateurs mobiles s'attendent donc à ce que vous leur proposiez des **filtres** et des façons de trier les listings de produits en fonction de leurs besoins.

Par exemple, pour un site dédiée à la mode. Proposez des filtres en fonction : homme/femme/enfants, en fonction des tailles, en fonction des marques,... Proposez ensuite de classer les résultats par ordre de prix croissant ou décroissant, par popularité, par nouveauté,... Autant de possibilités qui permettront aux internautes d'aller plus vite dans leurs achats et donc de vivre une expérience de shopping agréable.

4 conseils pour développer vos ventes mobiles

220

2. Simplifiez les **formulaires d'inscription** pour commander

Si vous souhaitez éviter de perdre trop de clients lors de la première inscription pour passer commande, évitez les nombreux champs à remplir dans les formulaires.

Il est urgent de se focaliser sur l'optimisation des formulaires afin de booster la première étape du tunnel d'achat.

Certains éléments peuvent largement être améliorés :

La **mise en majuscule automatique** de la première lettre cause de nombreuses **erreurs** de création de compte et d'authentification

Oubliez les listes avec 100 entrées parmi lesquelles choisir (exemple : choisir votre pays parmi la liste proposée)

Des **champs inutiles** au premier abord pour le consommateur : date de naissance par exemple.

4 conseils pour développer vos ventes mobiles

221

3. Optimisez à 100% **vosre page de localisation**

Les pages de localisation de votre point de vente physique, ou du point de vente où sont vendus vos produits sont souvent peu optimisées pour la lecture mobile. La plupart du temps, la carte de localisation défile dans tous les sens lorsque l'internaute souhaite scroller l'écran vers le bas. Il scrolle toute la carte au lieu de descendre simplement sur la page web. A une main cette manipulation est vraiment compliquée.

Faites en sorte que votre version mobile ne fasse apparaître que le minimum d'informations concernant la localisation. Faire apparaître une carte est une très bonne idée si cette dernière ne gâche pas l'expérience de navigation des internautes.

4 conseils pour développer vos ventes mobiles

222

Soyez obsédé par la **vitesse de chargement**

Bien souvent, les sites mobiles sont très lents, même lorsque les internautes sont connectés au Wi-Fi. Il est urgent de revoir le poids de vos images, d'optimiser le CSS et le code Javascript car ce sont ces éléments en particulier qui allongent les temps de chargement des sites mobiles.

Les consommateurs sont de moins en moins tolérants face à des sites peu accessibles sur mobile. Ils ne vont pas crier au scandale mais vont simplement quitter le tunnel d'achat à peine entamé. Les conversions ne seront alors pas au rendez-vous. Le trafic provenant des devices mobiles n'est plus à prendre à la légère. Optimisez tous les canaux de conversion !

Couleurs principales

Matériel Design

223

`@android:style/Theme.Material` (dark version)

`@android:style/Theme.Material.Light` (light version)

`@android:style/Theme.Material.Light.DarkActionBar`

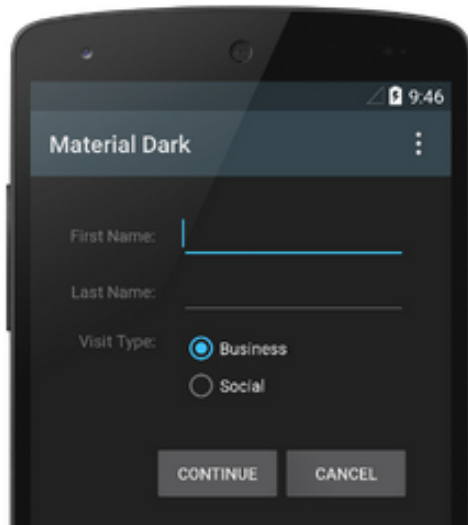


Figure 1. Dark material theme

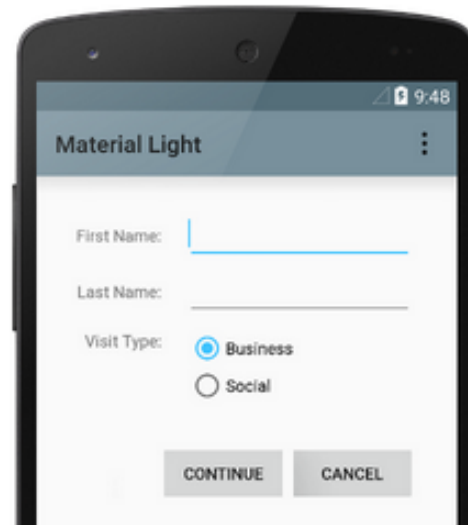


Figure 2. Light material theme

Définir son identité visuelle

224

styles.xml => **colors.xml**

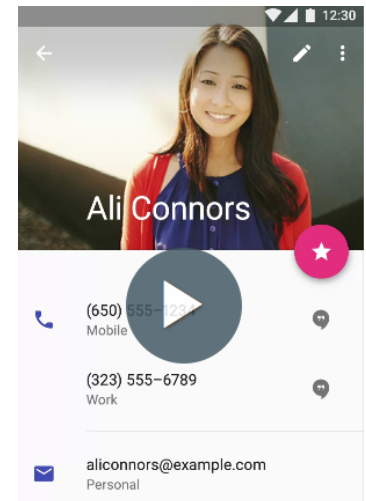
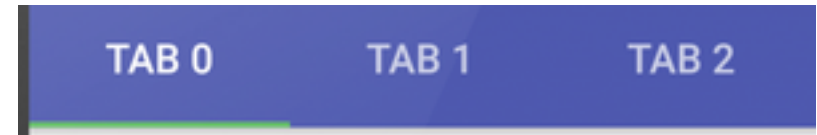
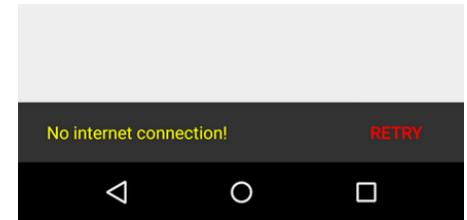
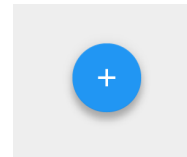
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

Prenez les couleurs de votre marque

Material Design

□ Les nouveaux composants

- FloatingActionButton
- Snackbar
- TabBarLayout
- CoordinatorLayout
- AppBarLayout
- CollapsingToolbarLayout

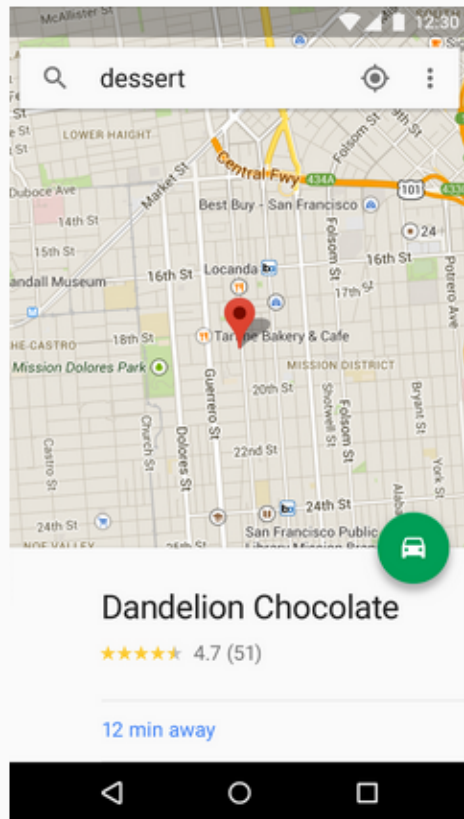


FloatingActionButton

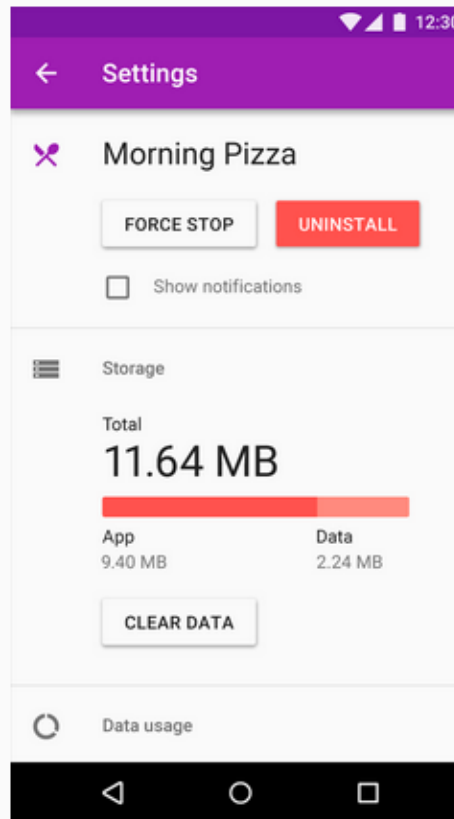


226

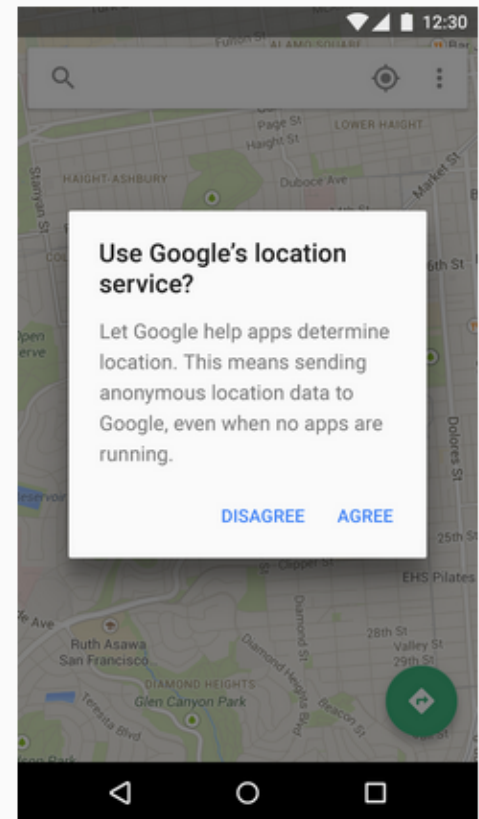
Doc officielle



Example of a floating action button



Example of a raised button



Example of a flat button

FloatingActionButton



227

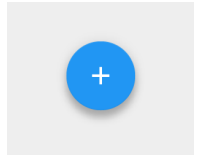
my_layout.xml

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:src="@android:drawable/ic_dialog_email" />
```

MainActivity.java

```
// floating Button
FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Votre action
    }
}
```

Interaction avec le Swipe du ViewPager : zooming + fading

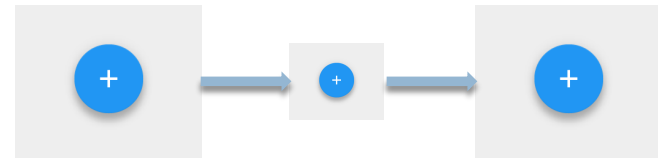


228

```
MainActivity implements ViewPager.OnPageChangeListener
... onCreate
mViewPager.addOnPageChangeListener(this);
...
@Override
public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels){
    FloatingActionButton fab = (FloatingActionButton)findViewById(R.id.fab);
    if (fab != null ) {
        if(positionOffset < 0.5) {
            fab.setScaleX(1 - 2*positionOffset); // zoom en fct position
            fab.setScaleY(1 - 2*positionOffset); // zoom en fct position
            fab.setAlpha(1 - 2*positionOffset); // alpha en fct position
        } else {
            fab.setScaleX(positionOffset);
            fab.setScaleY(positionOffset);
            fab.setAlpha(positionOffset);
        }
    }
}

@Override
public void onPageSelected(int position) {}

@Override
public void onPageScrollStateChanged(int state) {}
```



Les animations

229

- Avant 3.1 (Milieu HoneyComb) **View** Animations définies dans : `res/anim/`
- Depuis 3.1 (Milieu HoneyComb) **Property** Animations définies dans : `res/animator/`
 - ▣ Permet d'animer des propriétés : plus complet

Animations avec 2 buttons



230

Créer *fab_layout.xml*

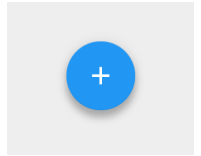
```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >>
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@android:drawable/ic_menu_compass"
        android:visibility="invisible"
        app:backgroundTint="@color/colorFAB"
        app:fabSize="mini" />
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab_2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@android:drawable/ic_menu_myplaces"
        android:visibility="invisible"
        app:backgroundTint="@color/colorFAB"
        app:fabSize="mini" />
</FrameLayout>
```

L'insérer dans le .xml au dessus du FAB existant

□ `<include layout="@layout/fab_layout"/>`

Créer les animations :

apparition du bouton plus



231

Créer *res/anim/fabplus_show.xml*

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

    <!-- Rotate -->
    <rotate
        android:duration="500"
        android:fromDegrees="30"
        android:interpolator="@android:anim/linear_interpolator"
        android:pivotX="50%"
        android:pivotY="50%"
        android:repeatCount="4"
        android:repeatMode="reverse"
        android:toDegrees="0"></rotate>

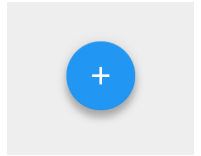
    <!--Move-->
    <translate
        android:duration="1000"
        android:fromXDelta="170%"
        android:fromYDelta="25%"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXDelta="0%"
        android:toYDelta="0%"></translate>

    <!--Fade In-->
    <alpha
        android:duration="2000"
        android:fromAlpha="0.0"
        android:interpolator="@android:anim/decelerate_interpolator"
        android:toAlpha="1.0"></alpha>

</set>
```

Créer les animations :

disparition du bouton plus



232

Créer *res/anim/fabplus_hide.xml*

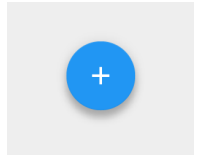
```
□ <?xml version="1.0" encoding="utf-8"?>
  <set xmlns:android="http://schemas.android.com/apk/res/android"
      android:fillAfter="true">

    <!--Move-->
    <translate
      android:duration="1000"
      android:fromXDelta="-170%"
      android:fromYDelta="-25%"
      android:interpolator="@android:anim/linear_interpolator"
      android:toXDelta="0%"
      android:toYDelta="0%"></translate>

    <!--Fade Out-->
    <alpha
      android:duration="2000"
      android:fromAlpha="1.0"
      android:interpolator="@android:anim/accelerate_interpolator"
      android:toAlpha="0.0"></alpha>

  </set>
```

Modifier le *Main.java* pour déclencher les anims



233

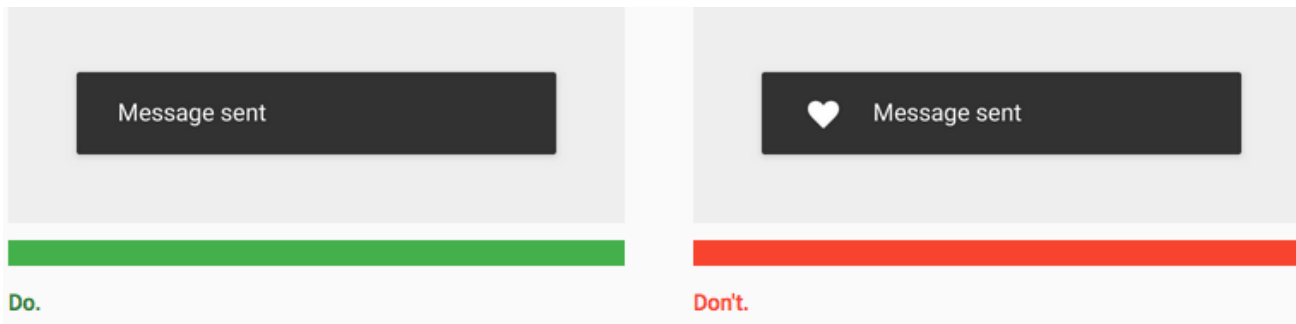
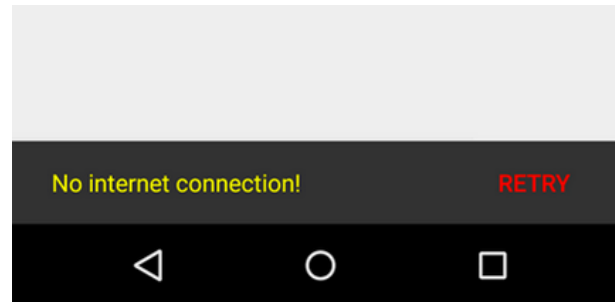
```
//Animations
Animation show_fab_1 = AnimationUtils.loadAnimation(getApplication(), R.anim.fabplus_show);
Animation hide_fab_1 = AnimationUtils.loadAnimation(getApplication(), R.anim.fabplus_hide);

...
```

SnackBar

234

- ❑ Permet l'affichage temporaire d'un message
- ❑ + Action si souhaité : 1 seule action
- ❑ Texte Only



SnackBar

235

```
SnackBar snackbar = SnackBar
    .make(findViewById(R.id.coordinatorLayout)
        , "Message is deleted", SnackBar.LENGTH_LONG)
    .setAction("UNDO", new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            SnackBar snackbar1 = SnackBar.make(coordinatorLayout,
                "Message is restored!", SnackBar.LENGTH_SHORT);
            snackbar1.show();
        }
    });

snackbar.show();
```

TabBarLayout

TAB 0

TAB 1

TAB 2

236

- ▣ Affiche le titre des tabs d'un ViewPager

my_layout.xml

```
<android.support.design.widget.AppBarLayout
...
    <android.support.design.widget.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorPrimaryDark" />
</android.support.design.widget.AppBarLayout>
```

MainActivity.java

```
...
mViewPager.setAdapter(fragmentsPagerAdapter);
TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
tabLayout.setupWithViewPager(mViewPager);
```

- ▣ Pour ne pas avoir un Txt sur 2 lignes : scrollable
`app:tabMode="scrollable"`
- ▣ Pour appliquer un style pour le tab
`style="@style/MyCustomTabLayout »`
- ▣ Définir le style du tab, texte, indicator dans *style.xml*

```
<style name="MyCustomTabLayout" parent="Widget.Design.TabLayout">
    <item name="tabMaxWidth">@dimen/tab_max_width</item>
    <item name="tabIndicatorColor">?attr/colorAccent</item>
    <item name="tabIndicatorHeight">2dp</item>
    <item name="tabPaddingStart">12dp</item>
    <item name="tabPaddingEnd">12dp</item>
    <item name="tabBackground">?attr/selectableItemBackground</item>
    <item name="tabTextAppearance">@style/MyCustomTabTextAppearance</item>
    <item name="tabSelectedTextColor">?android:textColorPrimary</item>
</style>
<style name="MyCustomTabTextAppearance" parent="TextAppearance.Design.Tab">
    <item name="android:textSize">10sp</item>
    <item name="android:textColor">@color/colorPrimary</item>
    <item name="textAllCaps">>false</item>
</style>
```

CoordinatorLayout

238

- **FrameLayout** qui permet de regrouper des vues (Child) afin de coordonner leur déplacements en fonction du CoordinatorLayout (Parent)
- On peut utiliser le DefaultBehavior ou un autre

AppBarLayout

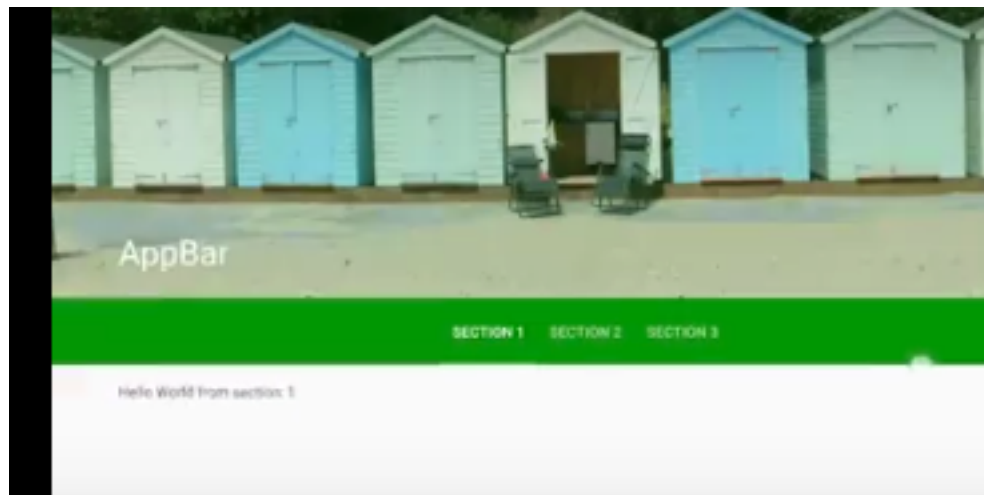
TAB 0

TAB 1

TAB 2

239

- ▣ Permet de gérer les éléments barre et les actions de scrolling associées
- ▣ A inclure dans un **CoordinatorLayout**
- ▣ Y ajouter un **CollapsingToolbarLayout** pour regrouper des éléments avec effet parallax



.xml

240

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="isen.guitare.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fitsSystemWindows="true"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:layout_width="match_parent"
            android:layout_height="300dp"
            app:contentScrim="?attr/colorPrimary"
            android:fitsSystemWindows="true"
            app:expandedTitleMarginEnd="@dimen/activity_horizontal_margin"
            app:expandedTitleMarginStart="@dimen/activity_horizontal_margin"
            app:layout_scrollFlags="scroll|exitUntilCollapsed">
            <ImageView
                android:id="@+id/toolbar_image"
                android:layout_width="match_parent"
                android:layout_height="300dp"
                android:adjustViewBounds="true"
                android:contentDescription="@null"
                android:scaleType="centerCrop"
                android:src="@drawable/vanessa"
                app:layout_collapseMode="parallax"/>
            <android.support.v7.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                app:layout_collapseMode="pin"
                app:popupTheme="@style/AppTheme.PopupOverlay"/>
            </android.support.design.widget.CollapsingToolbarLayout>

            <android.support.design.widget.TabLayout
                android:id="@+id/tabs"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="@color/colorPrimaryDark"
                app:tabMode="scrollable"
                app:layout_scrollFlags="scroll"
                android:layout_gravity="bottom"
                style="@style/MyCustomTabLayout"
            />
        </android.support.design.widget.AppBarLayout>

        <include
            layout="@layout/content_main" />

    </android.support.design.widget.CoordinatorLayout>
```

Explications

241

app:layout_scrollFlags

- **scroll** : ce flag doit être ajouté aux vues dont vous voulez qu'elles suivent le scroll du CoordinatorLayout. Sans ce flag, les vues ne seront pas déplacées
- **enterAlways** : ce flag permet d'activer le quick-return de cette vue. Elle disparaît avec le scroll up et réapparaît lors du scroll down
- **enterAlwaysCollapsed** : Lorsque votre vue possède une **minHeight** et que vous utilisez ce flag, celle-ci va apparaître initialement avec sa taille minimale, puis va suivre le scroll pour atteindre sa taille finale (collapse -> visible)
- **exitUntilCollapsed** : ce flag, à l'inverse du **enterAlwaysCollapsed**, fait apparaître la vue à sa taille initiale, puis la réduit jusqu'à sa **minHeight** lors du scroll (visible -> collapse)

app:layout_collapseMode

- **pin** : cette vue sera fixée en haut de l'écran lors du scroll
- **parallax** : la vue disparaîtra avec un effet parallax, donc avec une vitesse différente du scroll actuel
- en l'absence d'un **collapseMode**, la vue disparaît totalement de l'écran lors du scroll, comme vous en avez l'habitude

<http://www.tutos-android.com/material-design-support-library>

La référence

242

- <https://www.google.com/design/spec/style/icons.html#>
- Utiliser les icones dans les ressources
- <https://github.com/google/material-design-icons/releases/tag/1.0.1>

Annexes

UX

Excellent retour d'expérience sur le montage d'une équipe de User eXpérience

<http://www.pompage.net/traduction/experience-utilisateur-etude-de-cas>

Des tutos

Des trucs bien sympa

<http://code.tutsplus.com/categories/android-sdk>

Quelques outils pratiques

244

- Affichage du mobile sur un ordi: <http://droid-at-screen.org/>
- Emulateur rapide: <https://www.genymotion.com/>
- Plateforme Cloud mobile pour les notifications & Co :
 - [Parse](https://www.parse.com/) : <https://www.parse.com/>
 - Azure de Microsoft
- Outils de resize :

https://github.com/codepath/android_guides/wiki/Working-with-the-ImageView

<https://github.com/asystat/Final-Android-Resizer/blob/master/Executable%20Jar/Final%20Android%20Resizer.jar?raw=true>