Semaine thématique Android

Layouts et Widgets

TextView

- Une TextView permet d'afficher une chaine de caractères que l'utilisateur ne peur pas modifier.

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/textView"
    android:textSize="8sp"
    android:textColor="#112233" />
```

EditText

- Un EditText est utilisé pour permettre a l'utilisateur d'écrire des textes.

- Il s'agit en fait d'un TextView éditable.

```
<EditText
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:hint="@string/editText"
  android:inputType="textMultiLine"
  android:lines="5" />
```

Button

- Il s'agit d'un bouton
- On peut l'assimiler à un TextView cliquable

```
<Button
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="@string/button" />
```

CheckBox

- Une case qui peut avoir deux états : cochée ou pas
- Utile pour représenter un booléen lors de l'affichage

```
<CheckBox
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="@string/checkBox"
   android:checked="true" />
```

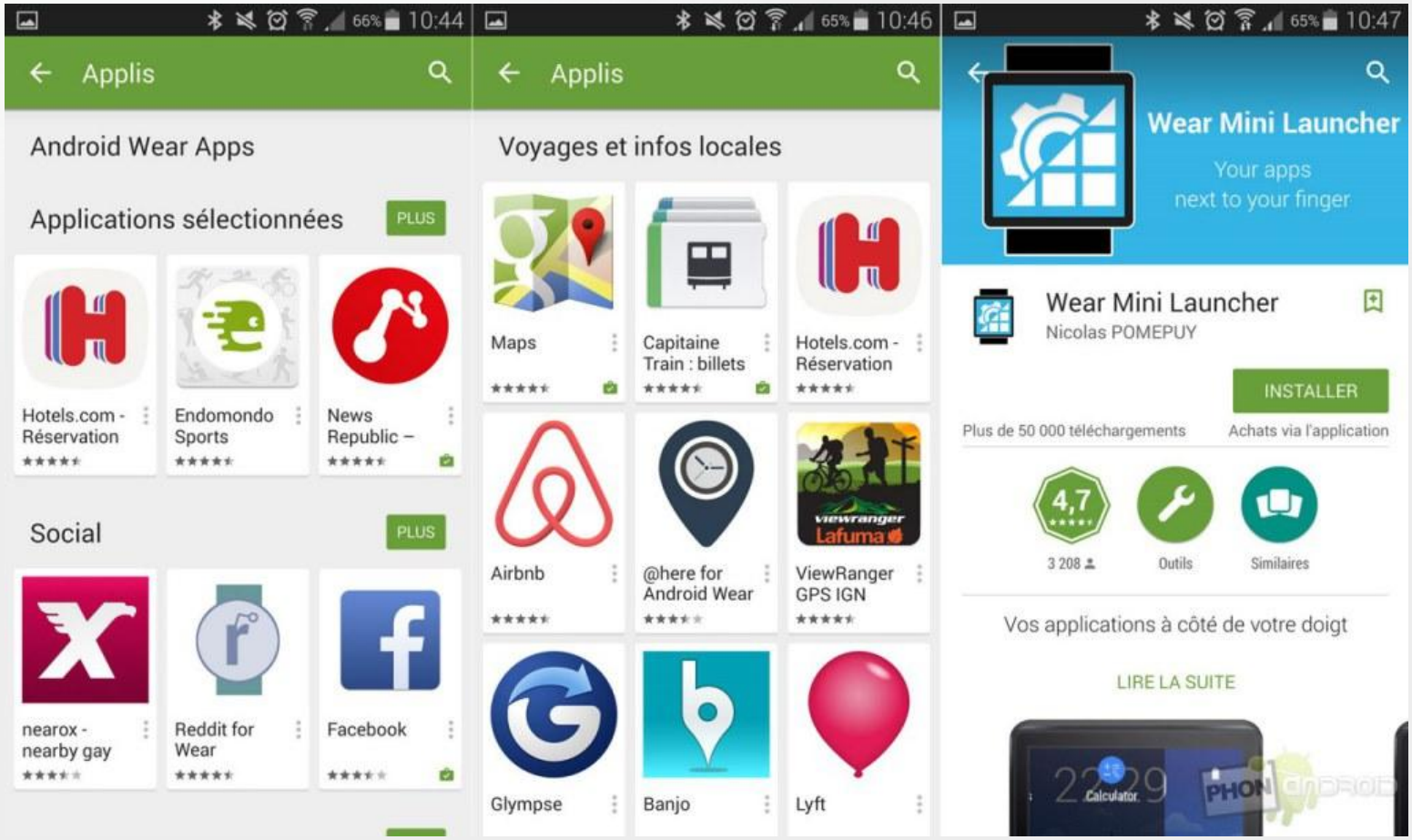
Récupération des composants

- On récupère les objets de l' Activity dans le code java au sein de la méthone onCreate().

- On utilise la méthode *findViewById(int id)*

```
Button btn = (Button) findViewById(R.id.myButtonId);
```

Layouts


Applis
Android Wear Apps
Applications sélectionnées
PLUS
Hotels.com - Réservation
Endomondo Sports
News Republic –
Social
PLUS
nearox - nearby gay
Reddit for Wear
Facebook
Applis
Voyages et infos locales
Maps
Capitaine Train : billets
Hotels.com - Réservation
Airbnb
@here for Android Wear
ViewRanger GPS IGN
Glympse
Banjo
Lyft
Wear Mini Launcher
Your apps next to your finger
Wear Mini Launcher
Nicolas POMEPUY
INSTALLER
Plus de 50 000 téléchargements
Achats via l'application
4,7
3 208
Outils
Similaires
Vos applications à côté de votre doigt
LIRE LA SUITE

Layouts

- Les layouts (à traduire par conteneurs) peuvent contenir des objets graphiques (boutons,…) mais aussi d'autres conteneurs
- Tous les layouts dérivent de la classe ViewGroup
- Les layouts ainsi que les objets graphiques sont appelés des View

LinearLayout

- Ce layout se charge de mettre les différentes vues sur une même ligne.
- On est obligé de préciser son orientation
- L'orientation est précisée à l'aide de l'attribut: android:orientation qui peut prendre les valeurs suivantes:
  - vertical: disposition de haut en bas (en colonne)
  - horizontal : disposition de gauche à droite (en ligne)

LinearLayout

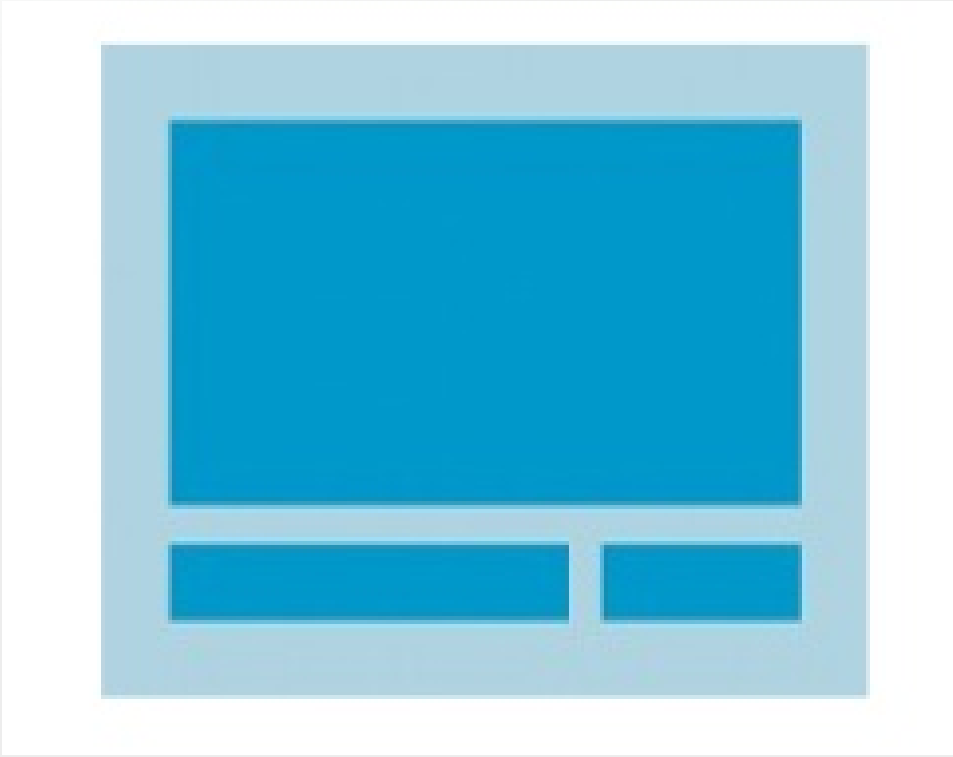
RelativeLayout

- Ce layout place les View les unes par rapport aux autres.

- Il est possible de le placer par rapport au RelativeLayout parent.

- Les attributs utilisables sont:
  - layout_alignParentTop
  - layout_alignParentLeft
  - layout_alignLeft
  - layout_alignRight
  - layout_below
  - layout_centerHorizontal

RelativeLayout



FrameLayout

- Il s'agit de positionner les objets View à un endroit précis de l'écran.
- Si l'on met plusieurs View dans un FrameLayout, ils vont tous se positionner dans le coin supérieur gauche et se superpositionner les uns sur les autres.
- Utilisé pour afficher un contenu spécifique par-dessus l'écran en cours sans avoir a modifier en profondeur l'écran.

FrameLayout