

Listes horizontales



Défilement horizontal

Dans `Play_activity.xml`, remplacer la `ListView` par

```
<HorizontalScrollView
    android:layout_width="match_parent"
    android:layout_height="128dp"
    android:layout_gravity="center"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:id="@+id/token_list_view2"/>
</HorizontalScrollView>
```

Le conteneur `token_list_view2` recevra les jetons.

Images réactives

Sous classe de `ImageView` qui implémente `View.OnClickListener`

```
public class TokenView extends ImageView
                        implements View.OnClickListener {
    TheApplication app;
    Integer id;
    public TokenView(Context c, AttributeSet as) {
        super(c,as);
        app = (TheApplication)c.getApplicationContext();
    }
    void setTokenId(Integer id) { this.id = id; }
    Integer getTokenId() { return id; }
    @Override
    public void onClick(View v) {
        app.setSelectedToken(id);
    }
}
```

Une vue pour les jetons

Nouvelle ressource xml : token_view.xml

```
<include xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/imageItem"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="3dp"/>
```

Servira à créer les images pour peupler la LinearLayout (token_list_view2).

Créer la liste d'images réactives

Dans `PlayActivity(2)`, méthode `onCreate`

```
LinearLayout tokenListView =
    (LinearLayout)findViewById(R.id.token_list_view2);
Context cx = this.getApplicationContext();
for(Integer id: app.getGame()) {
    DrawableToken d = new DrawableToken(cx, id);
    LayoutInflater inf = LayoutInflater.from(cx);
    TokenView v =
        (TokenView).inf.inflate(R.layout.token_view, null, false);
    v.setImageDrawable(d.get());
    v.setTokenId(id);
    v.setOnClickListener(v);
    tokenListView.addView(v);
}
```

Chaque vue de jeton s'écoute elle-même...

Positionnement

Attributs

- ▶ `gravity` pour positionner le contenu
- ▶ `layout_gravity` pour se positionner vis-à-vis de son conteneur

Valeurs : `left right top bottom center center_horizontal center_vertical`

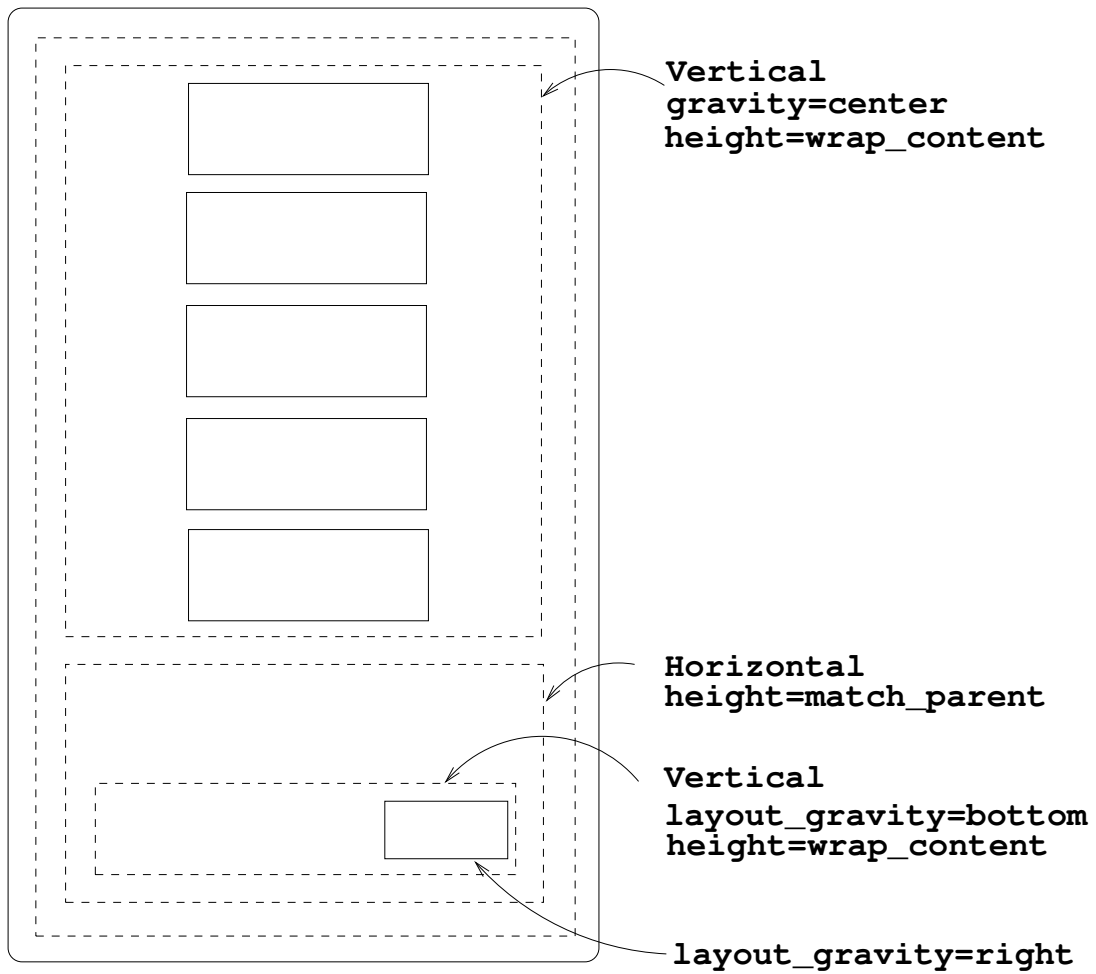
DÉMO :

1. `gravity`
2. `layout_gravity` in vertical `LinearLayout`
3. `layout_gravity` in horizontal `LinearLayout`

<http://stackoverflow.com/questions/3482742/gravity-and-layout-gravity-on-android>

«gravity doesn't work well on a `RelativeLayout` but can be useful with a `LinearLayout`»

Composer les positions



Orientation du terminal

Un *layout* pour chaque cas :

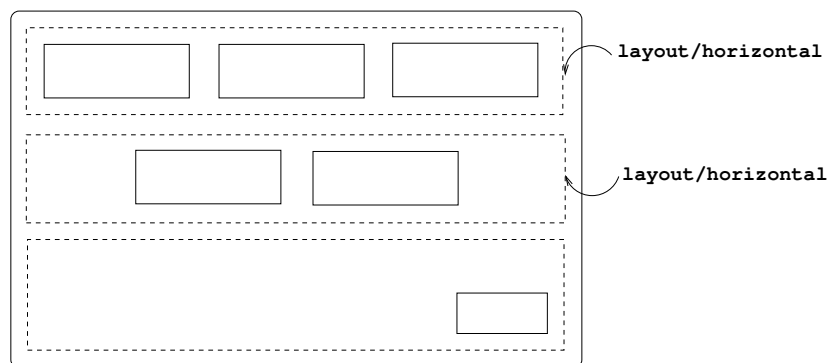
- ▶ par défaut : portrait
- ▶ créer une ressource XML spécifique pour l'orientation paysage

Système de fichier :

- ▶ `res/layout/activity_main.xml`
- ▶ `res/layout-land/activity_main.xml`

Android Studio : démo

1. sur *layout* : clic droit/new/Layout resource file
2. *File name* : `activity_main`
3. *Available qualifiers* : `orientation/landscape`



Proportions

Hauteurs ou largeur modulables : poids relatifs

Attribut : `android:layout_weight`

DÉMO : `LinearLayout`

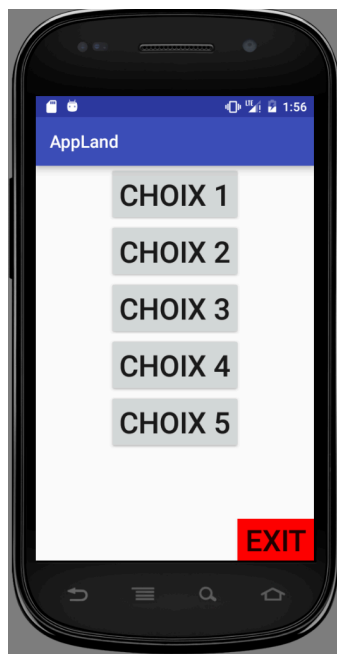
- ▶ `vertical` : partager la hauteur
`height=0dp` le système la calculera
- ▶ `horizontal` : partager la largeur
`width=0dp` le système la calculera

<http://stackoverflow.com/questions/31344768/how-to-give-layout-weight-in-relativelayout>

«You cannot use percentages to define the dimensions of a View inside a RelativeLayout. The best ways to do it is to use LinearLayout and weights, or a custom Layout.»

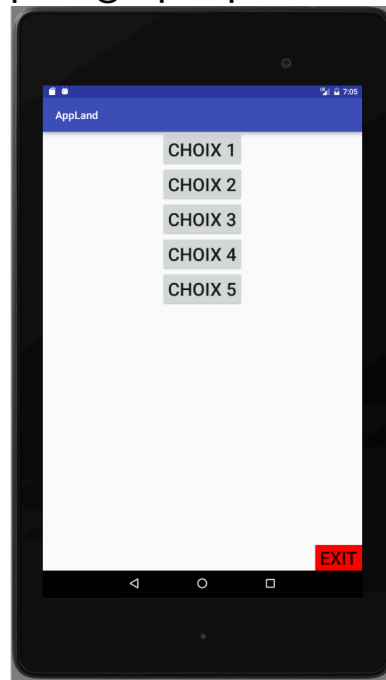
Taille et densité

2 terminaux aux caractéristiques graphiques différentes



normal-hdpi

480×800 - ≈240dpi



large-xhdpi

1200×1920 - ≈320dpi

<https://developer.android.com/guide/topics/resources/providing-resources.html>

Dimensions

Constantes symboliques plutôt que constantes numériques

1. `res/values/dimens` :

créer des variantes pour `normal-hdpi` et `large-xhdpi`
(clic droit/new/Values resource file/dimens/size+density)

2. définir une taille pour chaque écran

`normal-hdpi` : `<dimen name="textSize">"34dp"</dimen>`

`large-xhdpi` : `<dimen name="textSize">"68dp"</dimen>`

3. utiliser les noms dans les `layout/*.xml`

`android:textSize="@dimen/textSize"`

(Démo)