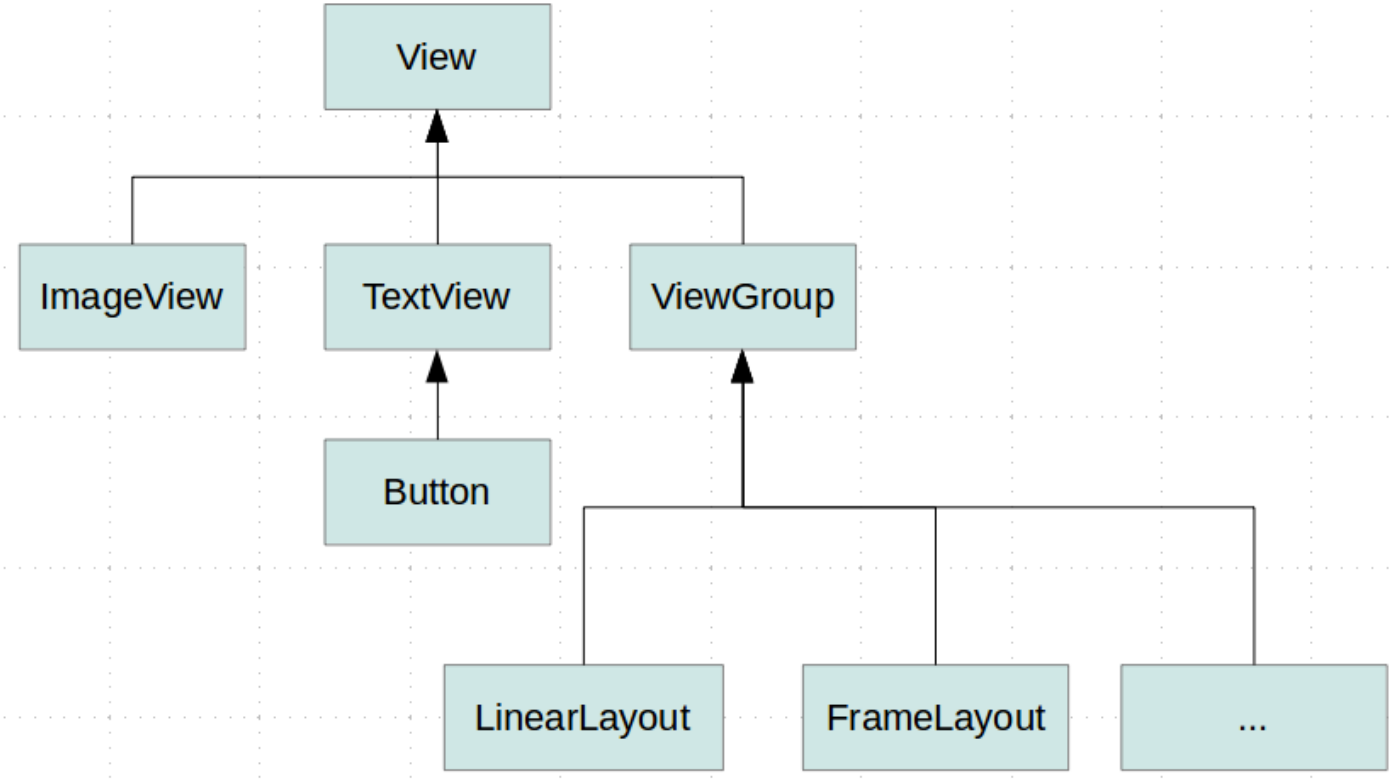
coaxys

POEC Android

Pierre Duchemin (@pierreduchemin)

Les layouts

- Sous Android, l'IHM est décrite en XML
- Les ViewGroup permettent d'organiser les vues ou d'autres ViewGroups



Propriétés des Views

- Les propriétés width et heigth sont obligatoires
- Elles peuvent prendre les valeurs :

> match_parent : prend toute la place disponible dans la View parente
> wrap_content : la View prend la place minimale requise
> ~~fill_parent~~

Le LinearLayout

- Il permet simplement de mettre des Views les unes à la suite des autres
- Gère des pourcentages via la layout_weight et weight sur ses enfants
- Voir : orientation, gravity

Le RelativeLayout

- Il permet de positioner des Views les unes par rapport aux autres ou par rapport à lui-même
- Il est moins performant
- Voir : below, above, toRightOf, toLeftOf, alignParentTop, alignParentBottom...

Le TableLayout

- Utile pour afficher des données très structurées
- Le TableLayout contient des TableRow
- Chaque TableRow doit contenir le même nombre d'éléments

Color.xml

- Fichier où les couleurs de l'application sont stockées.
- Permet de les centraliser car elles seront souvent utilisées

```
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

Exercice 1

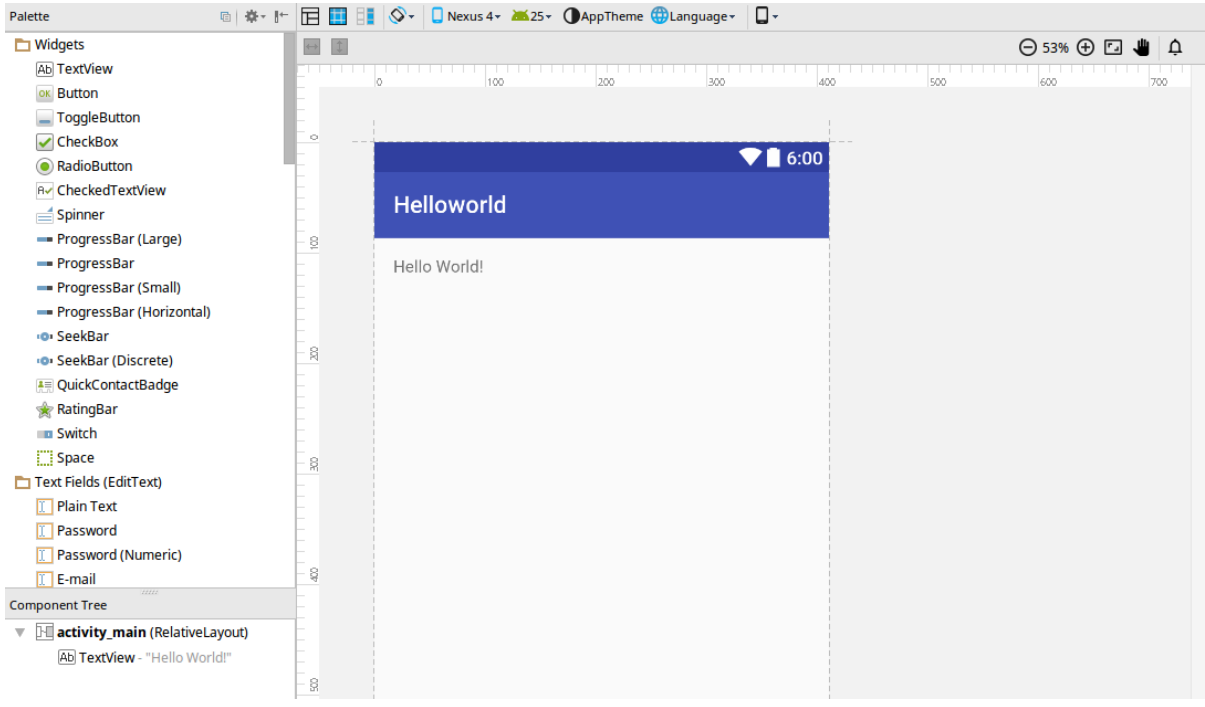
- Faire cette interface :

Exercice 2

- Faire cette interface :

L'éditeur WYSIWYG

- Android Studio dispose d'un éditeur WYSIWYG (What You See Is What You Get)
- Il est utile pour découvrir les différentes Views
- ⚠️ Cet éditeur produit un code peu maintenable

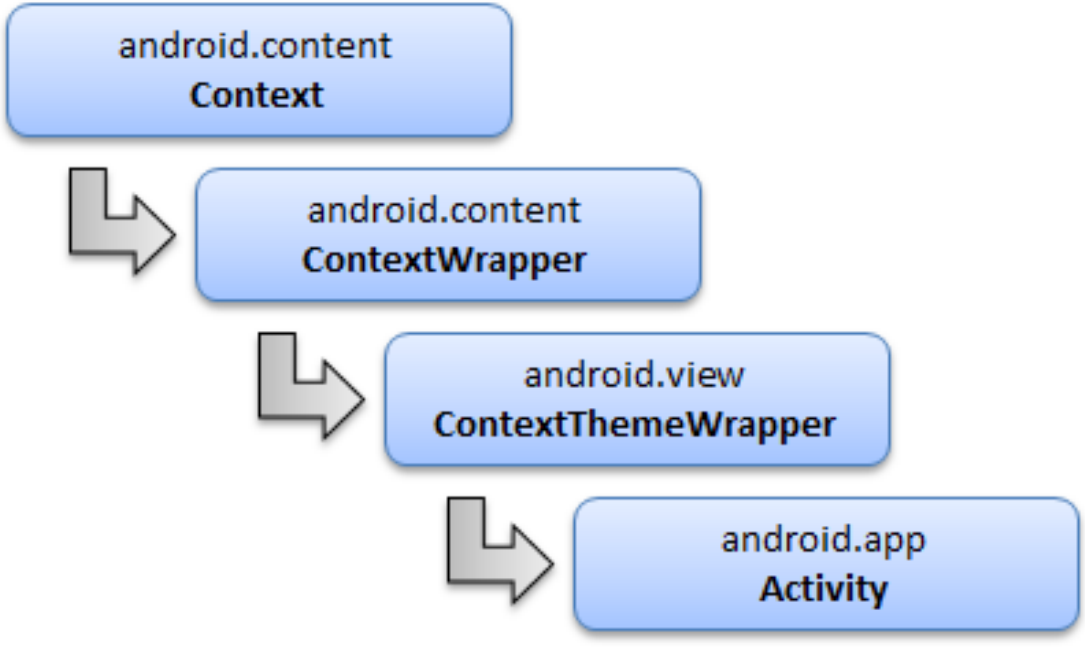


Le contexte

> Il s'agit de déterminer l'état dans lequel se trouve l'application

- Il est possible d'utiliser un contexte global ou plus spécifique en fonction du besoin

> Dans un Contexte (Application, Activity, Service...) : this
> Pour avoir le contexte global : getApplicationContext()
> Sinon : le passer en paramètre



Réagir au clic

- Il est possible d'ajouter un OnClickListener sur toutes les Views

```
public class MainActivity extends AppCompatActivity {

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

        View tvHello = findViewById(R.id.tvHello);
        tvHello.setOnClickListener(new View.OnClickLister
            @Override
            public void onClick(View v) {
                Toast.makeText(MainActivity.this, "View c
            }
        });
    }
}
```

Réagir au clic

Il est aussi possible d'utiliser "onClick" en xml :

```
    <Button
        android:id="@+id/reset_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClickReset"
        android:text="@string/reset" />
```

Java 8 et Android

- Android ne supporte pas encore totalement java 8
- RetroLambda et Lightweight-Stream-API permettent d'utiliser les lambda et les streams

Exercice 3

- Faire une application pour gérer un quiz basique

Exercice 4

- Rendre les questions et réponses paramétrables

Gagner du temps

- Android Studio propose de **nombreux raccourcis**
- **CTRL + Shift + A** : accéder aux fonctionnalités d'Android Studio
- Theme Editor : pour **facilement customiser l'application**
- Translation Editor : permet de modifier simplement les fichiers de traduction. Permet de **voir les traductions manquantes**
- Configurer l'auto-format