

Compte rendu TP2 : Les interfaces sous Android
---

**Sommaire :**

1) Contexte .....	2
2) Tutoriel sur les groupement de vues .....	2
2.1 Mise en page avec des LinearLayout .....	2
2.2 Mise en page avec un TableLayout .....	8
2.4 Mise en page avec un ConstraintLayout .....	9
3) Exercices .....	10
a) Saisie d'un livre : livre.xml .....	10
b) Saisie d'un livre : repas.xml .....	12
4) Conclusion .....	13

## 1) Contexte

Ce TD avait pour objectif de découvrir différentes ressources telle que "LinearLayout", "TableLayout", "TableRow" et "RelativeLayout" en confectionnant différentes pages sous AndroidStudio.

## 2) Tutoriel sur les groupement de vues

### 2.1.Mise en page avec des LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="136dp"
        android:layout_marginTop="16dp"
        android:text="OK"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_marginStart="143dp"
        android:layout_marginTop="203dp"
        android:text="Annuler"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />
```

```
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_marginStart="114dp"
    android:layout_marginTop="107dp"
    android:text="Annuler tout"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Cela donne une fois sur l'émulateur :



Il fallait ensuite examiner la mise en page en comparant avec les propriétés des vues :

- Mettez `match_parent` pour `android:layout_width` de l'un des boutons, puis annulez le changement.

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="136dp"
    android:layout_marginTop="16dp"
    android:text="OK"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />
```

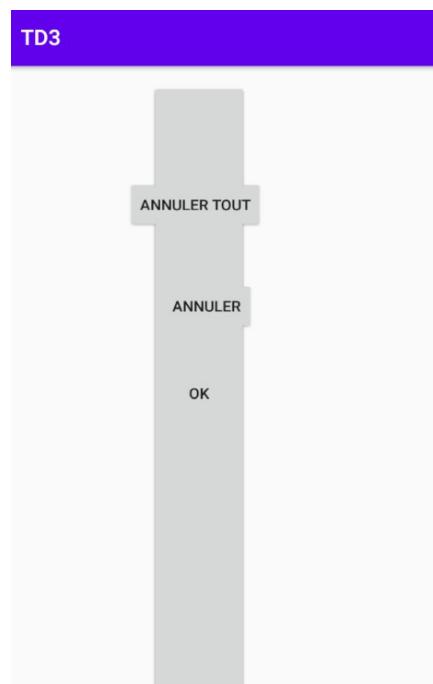
Cela agrandi la largeur du bouton Ok.



- Mettez `match_parent` pour `android:layout_height` de l'un des boutons, regardez son effet, puis annulez.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginStart="136dp"
    android:layout_marginTop="16dp"
    android:text="OK"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />
```

Cela agrandi la hauteur du bouton Ok.



- Changez android:orientation en horizontal pour le LinearLayout, laissez ce changement. Passez en mode paysage si les vues sont trop encombrantes.

```
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:orientation="horizontal"  
tools:context=".MainActivity">
```

Cela donne une fois sur l'émulateur :



Il fallait ensuite supprimer les 2e et 3e boutons, ne laisser que le OK. Ajoutez un EditText devant le bouton OK. C'est à dire que l'EditText prend presque toute la place en largeur, le bouton OK est minimal.

Pour cela, il faut mettre un poids nul au bouton OK mais aussi lui donner une taille minimale.

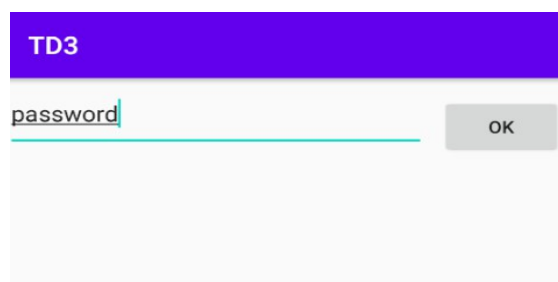
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="316dp"
        android:layout_marginTop="16dp"
        android:layout_weight="1"
        android:text="OK"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />

    <EditText
        android:layout_width="305dp"
        android:layout_height="59dp"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="5dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Cela donne une fois dans l'émulateur :

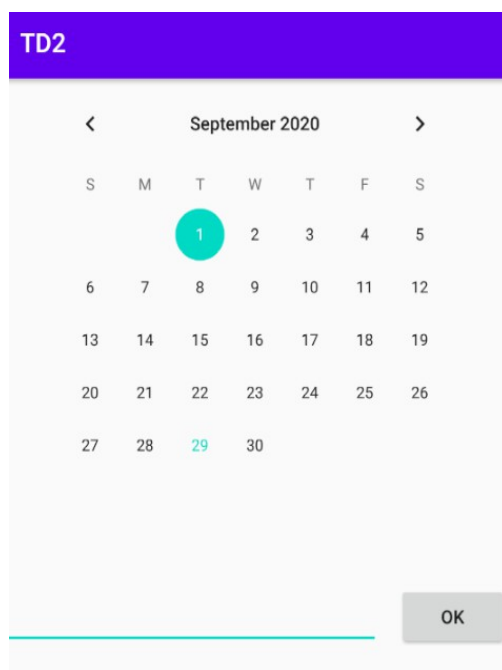


Il fallait ensuite rajouter un calendrier au code précédent.

```
<CalendarView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />
```

« CalendarView » a permis de rajouter le calendrier mais il fallait que celui-ci devait se trouver au-dessus de la zone de texte et du bouton codés plutôt. J'ai donc créé deux « LinearLayout » afin que le calendrier se trouve dans une ligne réservée uniquement à son nom et que la seconde ligne ne concerne que le mot de passe et le bouton

Cela nous donne la page suivante :



## 2.2 Mise en page avec un TableLayout

« TableLayout » permet de disposer un tableau. Il m'a fallu codé des « Button » dans un « TableRow » afin que les boutons soient disposés justement, dans un tableau :

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:shrinkColumns="0"
    android:stretchColumns="1"
    android:layout_height="match_parent">

    <TableRow>
        <Button android:text="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <Button android:text="col2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </TableRow>

    <TableRow>
        <Button android:text="2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <Button android:text="col2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </TableRow>
</TableLayout>
```

Cela nous donne :

1	COL2
2	COL2



## 2.4 Mise en page avec un ConstraintLayout

« ConstraintLayout » permet de disposer de manière dynamique des widgets dans des dispositions dynamiques. Il nous est demandé de disposer 3 boutons dans un ordre spécifique lorsque le portable est en mode paysage :

```
<?xmlversion="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="BTN1"
        app:layout_constraintHorizontal_bias="0.504"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="159dp"
        android:layout_height="675dp"
        android:layout_marginTop="4dp"
        android:text="BTN2"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="0.356"
        app:layout_constraintRight_toRightOf="0.356"
        app:layout_constraintTop_toBottomOf="@id/button1" />
```

J'ai utilisé des contraintes pour que certains boutons se trouvent, par rapport à d'autres, à un endroit en particulier

Voilà la page une fois sur l'émulateur



### 3) Exercices

#### a) Saisie d'un livre : livre.xml

Ce premier exercice consiste à créer un écran de saisie pour un livre avec la possibilité d'y insérer le titre du livre, l'auteur, l'année, l'isbn et la notation du livre en question.

Pour la notation, il m'a fallu utiliser un « RatingBar » qui représente une barre de note en étoiles permettant de juger la qualité du livre. La notation devait être de 0,5 en 0,5, j'ai donc utilisé un « android:stepSize'0,5' ».

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Auteur : "
    android:textColor="#000000"
    android:textSize="14sp"
    android:textStyle="bold"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="106dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Année : "
    android:textColor="#000000"
    android:textSize="14sp"
    android:textStyle="bold"
    tools:layout_editor_absoluteX="16dp"
    tools:layout_editor_absoluteY="254dp" />
```

```

<EditText
    android:id="@+id/Annee"
    android:layout_width="216dp"
    android:layout_height="47dp"
    android:hint="Entrez l'année du livre"
    android:inputType="numberDecimal"
    tools:layout_editor_absoluteX="97dp"
    tools:layout_editor_absoluteY="239dp" />

<EditText
    android:id="@+id/Titre"
    android:layout_width="216dp"
    android:layout_height="47dp"
    android:hint="Entrez le titre du livre"
    android:inputType="numberDecimal"
    tools:layout_editor_absoluteX="97dp"
    tools:layout_editor_absoluteY="16dp" />

<EditText
    android:id="@+id/ISBN"
    android:layout_width="216dp"
    android:layout_height="47dp"
    android:hint="Entrez l'ISBN du livre"
    android:inputType="numberDecimal"
    tools:layout_editor_absoluteX="97dp"
    tools:layout_editor_absoluteY="163dp" />

```

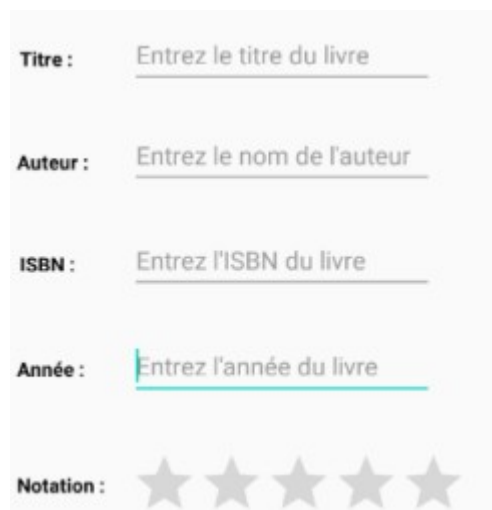
```

<RatingBar
    android:id="@+id/rating"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:saveEnabled="true"
    tools:layout_editor_absoluteX="97dp"
    tools:layout_editor_absoluteY="322dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="ISBN : "
    android:textColor="#000000"
    android:textSize="14sp"
    android:textStyle="bold"
    tools:layout_editor_absoluteX="18dp"
    tools:layout_editor_absoluteY="179dp" />

```

Voila le résultat obtenu une fois sur l'émulateur.



#### b) Saisie d'un livre : repas.xml

Cet exercice, consister à créer une interface pour saisir un repas. J'ai insérer des « RadioButton » pour indiquer s'il s'agissait d'un déjeuner ou d'un repas (les « RadioButton » sont des boutons que l'on peut cocher pour saisir une réponse), j'ai également utiliser une zone de saisie texte pour l'entrée, une pour le plat principal et deux cases à cocher (« RadioButton ») pour indiquer si on a mangé du fromage et un fruit (on doit pouvoir cocher les deux). Enfin, j'ai rajouter une vue pour saisir la date, pour ce faire, j'ai utiliser un « DatePicker » qui est presque semblable à un « CalendarView » sauf qu'il permet de sélectionner une date précise :

```
<DatePicker
    android:id="@+id/simpleDatePicker"
    android:layout_width="413dp"
    android:layout_height="330dp"
    android:layout_marginTop="288dp"
    android:datePickerMode="spinner"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/Entrée"
    android:layout_width="match_parent"
    android:layout_height="47dp"
    android:layout_marginTop="124dp"
    android:hint="Entrée"
    android:inputType="numberDecimal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<RadioButton
    android:id="@+id/radio_ninjas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:layout_marginTop="83dp"
    android:onClick="onRadioButtonClicked"
    android:text="Diner"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<RadioButton
    android:id="@+id/radio_pirates"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:layout_marginTop="51dp"
    android:onClick="onRadioButtonClicked"
    android:text="Déjeuner"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Voila le résultat une fois sur l'émulateur :

The screenshot shows a mobile application interface for logging meals. At the top, it says "Saisie d'un repas :". Below this, there are two radio buttons: "Déjeuner" and "Dîner". Underneath, there are two text input fields: "Entrée" and "Plat principal". Below these, there is a section titled "Vous avez mangé :" with two radio buttons: "Fromage" and "Fruits". At the bottom, there is a calendar for September 2020. The calendar shows the days of the week (S, M, T, W, T, F, S) and the dates. The date 29 is highlighted in green.

Saisie d'un repas :

☐ Déjeuner

☐ Dîner

Entrée

Plat principal

Vous avez mangé :

☐ Fromage ☐ Fruits

< September 2020 >

Aug	28	31	S	M	T	W	T	F	S
					1	2	3	4	5
Sep	29	2	6	7	8	9	10	11	12
Oct	30	3	13	14	15	16	17	18	19
			20	21	22	23	24	25	26
			27	28	29	30			

#### 4) Conclusion

Ce TP m'a posé de nombreuses difficultés, mais m'a permis d'en apprendre plus sur le développement d'applications mobiles et sur le fonctionnement d'Android Studio.