PROGRAMMATION MOBILE

ANDROID

TD 1

PREMIERE APPLICATION

charles.meunier@u-bourgogne.fr

INFORMATIONS GENERALES

OBJECTIFS

A travers les exercices qui suivent, vous verrez comment :

- Créer une application Android,
- Mettre en place des composants visuels
- Gérer des événements

CONVENTIONS

- Les informations entre crochets (ex : `[username]`) sont à remplacer dans les exemples fournis.
- Les consignes à réaliser sont précédées par ➢.

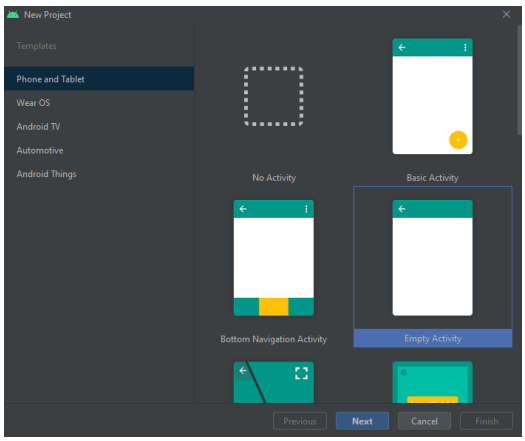
N'oubliez pas que votre enseignant est là pour répondre à vos questions.

ANDROID STUDIO

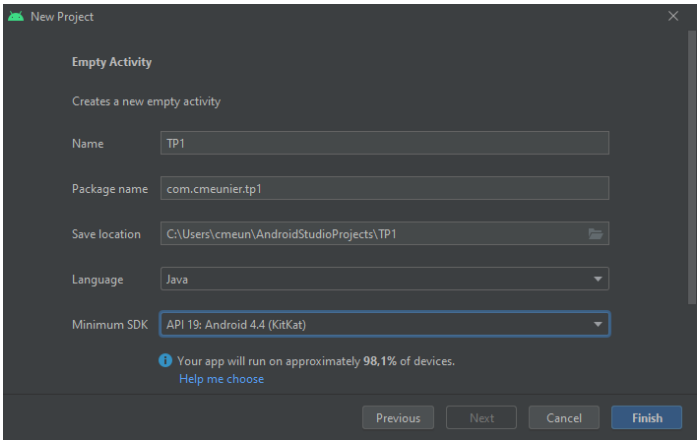
Si vous n'avez pas encore installé Android Studio, suivez les instructions du guide « Installation Android Studio ».

NOUVEAU PROJET

- ➢ Démarrez Android Studio
- ➢ Créez un nouveau projet
- ➢ Dans l'onglet « Phone / Tablet », choisissez « **Empty Views Activity** »

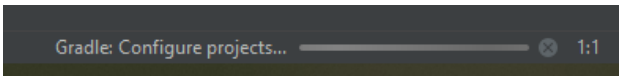


- ➢ Saisissez les informations suivantes pour le projet :
  - Name : **TD 1**
  - Package Name : **com.[your-name].td1**
  - Language : **Kotlin**
  - Minimum SDK : **API 25 : Android 7.1 (Nougat)**

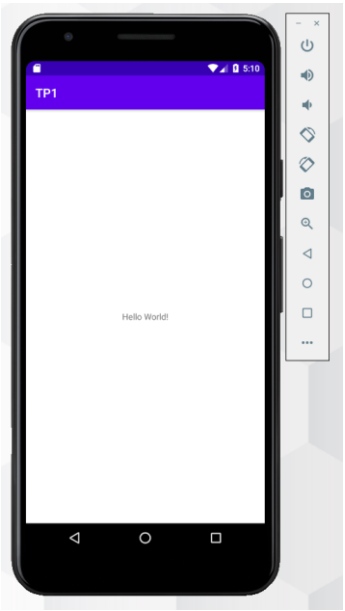


- Android en est à la version 13. Pourquoi est-il demandé de se limiter à la version 7.1 ?

- Patientez le temps que Gradle configure le projet.



- Exécutez l'application pour être certain que tout fonctionne.

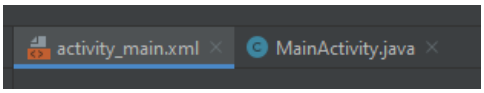


INTERFACE GRAPHIQUE

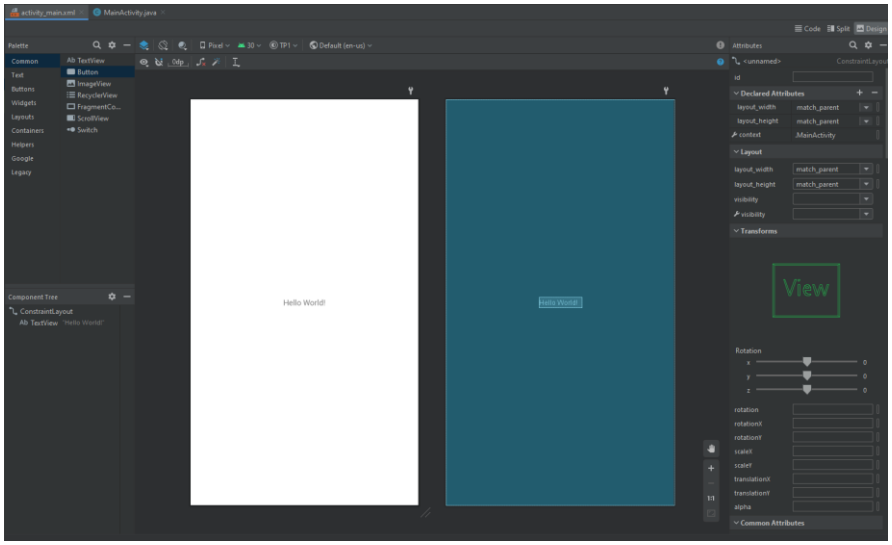
PREMIERES OBSERVATIONS

Le fichier « activity_main.xml », qui est généré automatiquement, est un layout, c'est-à-dire un fichier de mise en page qui détermine les composants graphiques à afficher à l'écran et leurs caractéristiques.

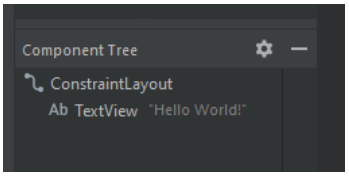
- Cliquez sur le fichier « activity_main.xml »



- Cela fera apparaître l'éditeur d'interface utilisateur.



Comme vous pouvez le voir, des composants ont été ajoutés par défaut :



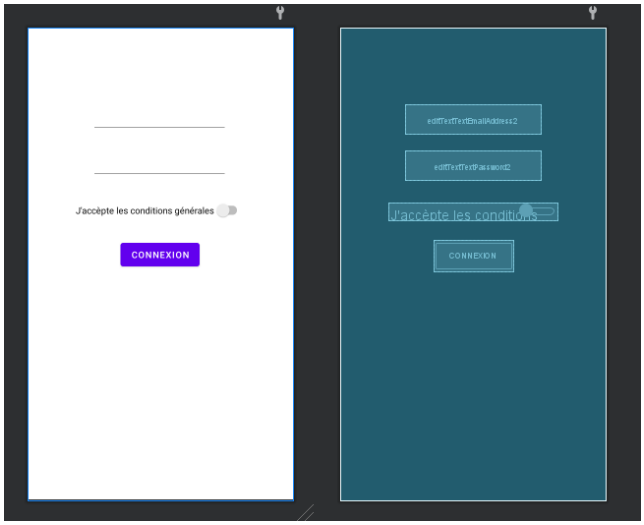
- Le composant `ConstraintLayout` est un conteneur. Son rôle est de contenir d'autres composants et de définir la manière avec laquelle ils vont se positionner. Un conteneur peut également contenir d'autres conteneurs. Il existe différents types de conteneurs (`ConstraintLayout`, `LinearLayout`, `TableLayout`, …), chacun offrant un mode de positionnement différent.
- Le composant `TextView` qui contient un texte statique (l'utilisateur ne peut pas le modifier).

➢ Quelles sont les différences entre les conteneurs `ConstraintLayout` et `LinearLayout` ?
➢ Sélectionnez tous les composants et supprimez-les. Que constatez-vous ? Pourquoi selon vous ?

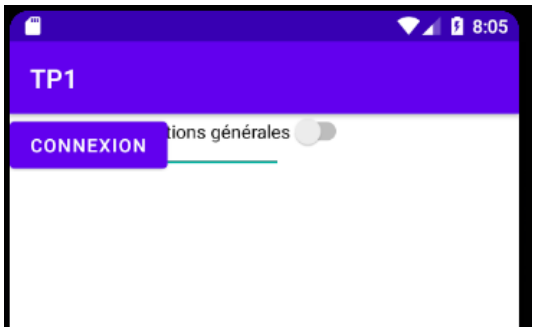
FORMULAIRE DE CONNEXION

Vous allez à présent créer un formulaire de connexion.

➢ Ajoutez dans le `ConstraintLayout` :
  - 1 x champ texte `Email`
  - 1 x champ texte `Password`
  - 1 x bouton Switch intitulé « J'accepte les conditions générales »
  - 1 x bouton intitulé « Connexion »

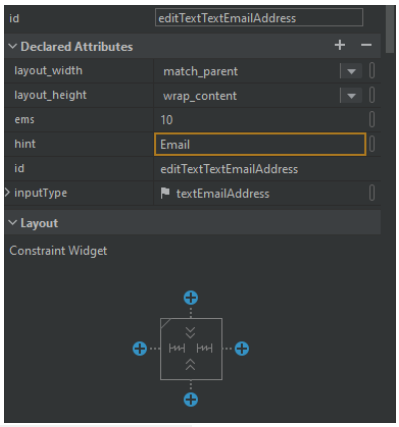


- Exécutez l'application. Vous devriez obtenir quelque chose comme ceci :



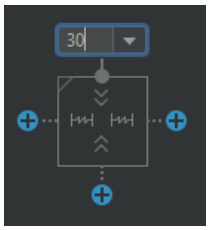
Tous les composants sont placés les uns sur les autres. Cela vient du `ConstraintLayout` qui place les composants les uns par rapport aux autres. Vous n'avez pas encore défini de règle de placement, ce qui explique cet affichage « raté ».

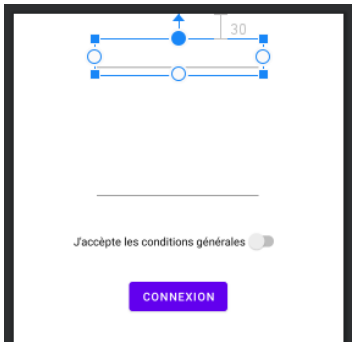
- Sélectionnez le premier champ texte (`Email`), puis explorez les propriétés du composant sur la droite de l'écran.



- Repérez la propriété `Constraint Widget`
- Cliquez sur le bouton « + » du dessus. Cela indique que le composant sera placé en dessous du composant précédent, à une distance fixe. Ici, le composant texte est le premier, il se placera donc tout en haut de l'écran.

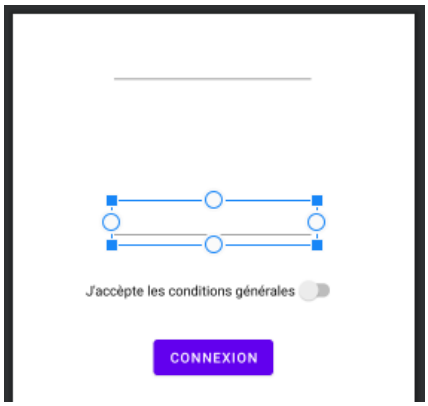
- Indiquez « 30 » pour placer le champ texte à 30 pixels du bord de l'écran.



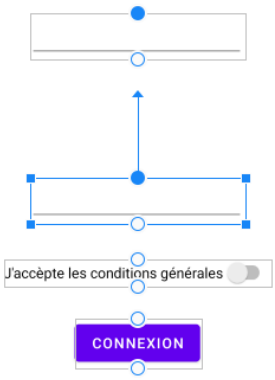


A présent, il faut placer les autres composants les uns par rapport aux autres.

- Sélectionnez le second champ texte (`Password`) :



- Cliquez sur le rond blanc supérieur et maintenez le clic de la souris enfoncé. Puis déplacer votre souris jusqu'à atteindre le rond blanc inférieur du champ `Email`. Une flèche suis le mouvement de la souris :

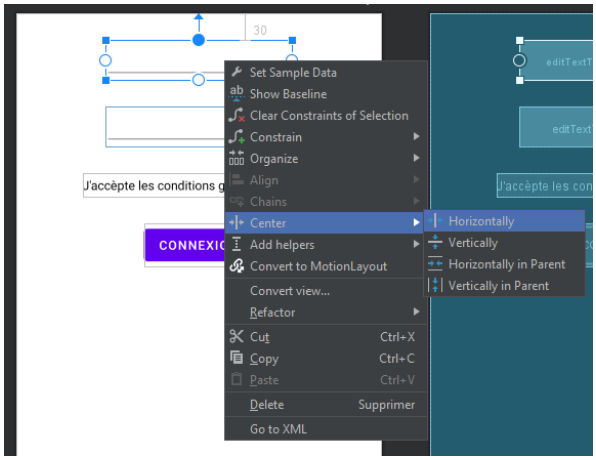


  Cela crée un lien entre les deux composants, `Password` se positionnant par rapport à `Email`.

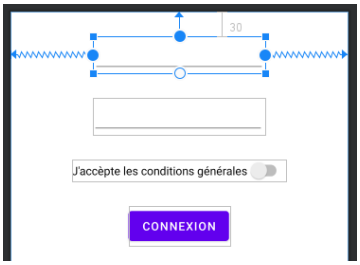
- Spécifiez « 20px » pour la distance supérieure dans le `Constraint Widget`.
- Faites de même pour le Switch et le bouton « Connexion » avec les composants qui les précèdent.
- Exécutez de nouveau l'application. Vous devriez obtenir quelque chose comme ceci :

C'est mieux, mais ce n'est pas parfait. Améliorons encore un peu les choses…

- Sélectionnez le champ `Email` et faites un clic droit dessus.
- Dans le menu, sélectionnez *Center* > *Horizontaly*



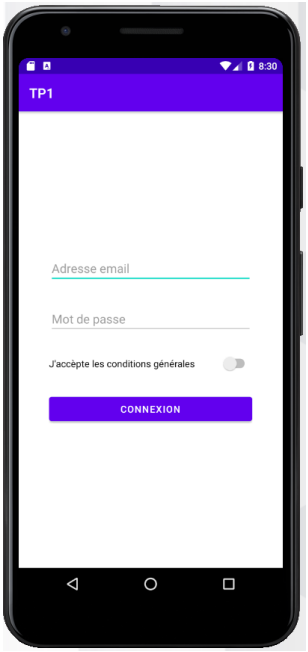
Cela contraint le composant à se centrer dans le layout :



- Faites de même pour les autres composants.
- Exécutez l'application à nouveau. Cette fois, cela devrait être bon :

- Faites pivoter l'écran du smartphone. Vous devriez toujours avoir un affichage correct :

- Modifiez votre interface pour obtenir le même résultat que ci-dessous :



ACTIVITE

PREMIERES OBSERVATIONS

Comme vu précédemment, le fichier layout « activity_main.xml » représente les composants graphiques de l'activité et leurs caractéristiques.

Un second fichier a été généré avec votre projet : « MainActivity.java ». Il s'agit de la classe représentant le code de l'activité.

- Affichez le contenu du fichier « MainActivity.java »

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

- Selon vous, que fait la méthode `onCreate` de `MainActivity` ?

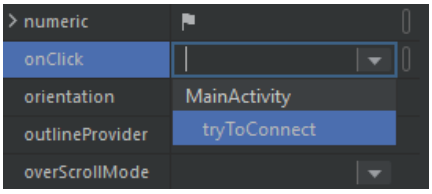
EVENEMENT ONCLICK

Vous allez à présent ajouter une action lorsque l'utilisateur clique sur le bouton « Connexion » de la fenêtre.

- Ajoutez à la classe `MainActivity` la méthode suivante :

```
public fun tryToConnect(view: View)
{
    AlertDialog
        .Builder( context: this)
        .setTitle("Demande de connexion")
        .setMessage("Nouvelle demande de connexion")
        .show()
}
```

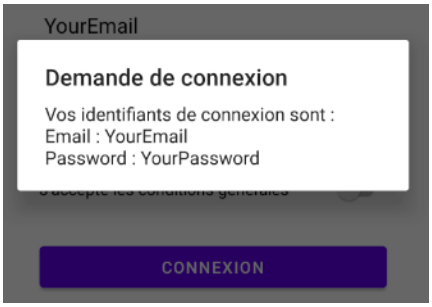
- Corrigez les éventuelles erreurs.
- Retournez dans le fichier « activity_main.xml »
- Sélectionnez le bouton « Connexion »
- Recherchez sa propriété `onClick`
- Attribuez la méthode `tryToConnect` de `MainActivity`



- Exécutez l'application et testez.

RECUPERATION D'INFORMATIONS

- Modifiez le code de la méthode `tryToConnect` afin que la boite de dialogue affiche le message suivant :



Où `YourEmail` et `YourPassword` correspondent aux valeurs saisies dans les champs `Email` et `Password`.

```
//Récupère un champ texte de l'interface
val textField = findViewById<EditText>(R.id.[id_du_champ_text])

//Récupère la valeur du champ texte
val value = textField.text.toString()
```

- Exécutez l'application et testez.

Fin de l'échauffement, on passera aux choses sérieuses avec le second sujet !