



TP n°1 Initiation au développement sur Android

Objectif

Ce TP permet de vous introduire au développement d'application Android ainsi que vous initialiser à l'environnement de développement. En effet, dans ce TP vous allez réaliser les premiers pas pour apprendre les principes de bases et le minimum nécessaire du développement des applications Android natives.

Partie I : Ma première Activité :



1. Création d'un nouveau projet :

Pour créer un nouveau projet vous devrez suivre les étapes suivantes :

- a. Lancez Android Studio et créez un nouveau projet nommé « tp1_yourname » ayant comme nom de package « eniso.android.tp1_yourname » : *File → New → New Project.*
- b. Choisissez "Empty Activity" → Next.
- c. Editez les informations nécessaires dans le formulaire affiché et donnez le nom « FirstActivity » à l'activité à créer → Finish.

2. Prise en main de l'environnement

NB : Vous pouvez exécuter votre projet sur un émulateur ou un téléphone.

- a. Lancez cette application en cliquant sur le bouton  après avoir créé un émulateur via l' **AVD Manager** . D'où vient le texte "Hello World !"?
- b. Remplacez ce texte affiché par défaut, par une chaîne de votre choix parmi celles existantes dans le fichier *res/values/strings.xml*.
- c. Ajoutez une ressource chaîne de caractères et affichez son contenu.
- d. Modifiez la taille de la police de ce texte (20sp, 35sp, etc.) ainsi que sa couleur (#0000FF, #C7A855, etc.).



- e. Refaites la question précédente en utilisant les deux fichiers XML (*dimens.xml* et *colors.xml*) pour stocker respectivement la nouvelle taille de la police et la nouvelle couleur.
- f. La figure.1 présente un code qui affiche un message (Toast) lors d'un clic sur un bouton. Illustrez de ce code pour afficher un Toast contenant un message de bienvenue lors du lancement de l'activité.

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

btn.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        Toast.makeText(v.getContext(), "Le bouton \"Valider\" a été cliqué", Toast.LENGTH_LONG).show();
    }
});
```

Figure 1: Gestion d'un clic sur un bouton "btn"

Partie II : Création d'applications

1. Interface d'authentification :

Ajoutez au projet « TP1_yourname » une nouvelle activité « **AuthenticationActivity** » de type EmptyActivity.

Clic droit sur la racine du projet (app) → New → Activity → Empty Activity.

NB : Ceci ajoutera à la fois, un fichier java sous src et un fichier xml sous res/layout

Commencez par modifier le layout qui a été généré par l'activité « AuthenticationActivity » pour qu'il ressemble à la figure 2. Sur cette figure, nous avons : deux champs de texte (EditText), deux labels (TextView) et un bouton (Button).

Cette activité représente une interface d'authentification composée par un **Login**, un **Password** et un bouton « **Connecter** » pour s'authentifier.

- Si l'authentification de l'utilisateur est correcte, un toast doit être affiché pour informer l'utilisateur que les données sont correctement saisies.
- Sinon, un toast doit être affiché pour signaler un message d'erreur.

NB :

- Tous les textes affichés dans les différentes vues (*TextView*, *EditText*, *Button*) doivent correspondre à des chaînes stockées dans le fichier **strings.xml**.
- La couleur des *TextView* (couleur bleu = #0000FF) est à lire à partir du fichier **colors.xml** déjà existant.
- La taille des mots **Login** et **Password** (15sp) est à lire à partir du fichier **dimens.xml** déjà existant.



- La gestion du clic sur le bouton **Connecter** se fait selon les étapes suivantes
 - o Récupération de l'objet bouton :

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

// Créez un attribut dans votre activité de type Button et l'initialisez en lui associant l'**ID** du bouton créé dans le layout.

- o Association d'un listener à ce bouton:

```
btn.setOnClickListener(...)
```

// Utilisez un listener au bouton pour définir son comportement lors d'un clic.

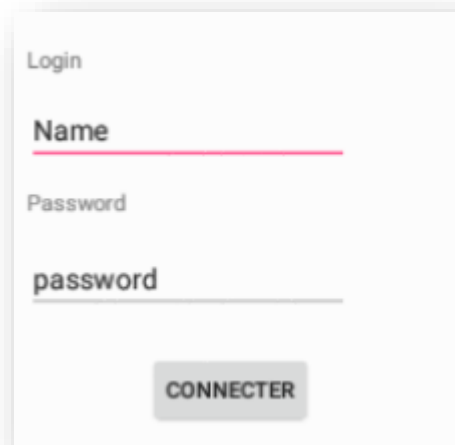


Figure 2 : L'interface correspondante à l'activité «AuthenticationActivity»

2. Passage d'une activité à une autre :

Ajoutez au projet «TP1_yourname» une autre activité «**ComputeActivity**» de type EmptyActivity. Cette activité permet de rechercher les nombres premiers dans un intervalle compris entre une valeur minimale et une valeur maximale.

Modifiez le layout qui a été généré par l'activité «**ComputeActivity**» pour qu'il ressemble à la figure 4. Sur cette figure, nous avons :

- trois champs de texte (EditText),
- quatre labels (TextView) : « Rechercher les nombres premiers », « Valeur minimale », « Valeur maximale » et « Résultat ».
- un bouton (Button) : « Rechercher »



Lorsqu'on clique sur le bouton « Rechercher », les nombres premiers doivent être affichés dans le champ associé au (TextView) Résultat.

NB :

- L'affichage des résultats se fait dans le dernier TextView comme le montre la figure 4.
- Le passage vers «**ComputeActivity**» se fait suite au succès de l'authentification au niveau de la première activité (Si les données du compte sont correctes, on passe, sinon, on garde le toast du message d'erreur).
 - o Ce passage de "AuthenticationActivity" vers "ComputeActivity" se fait grâce à un objet Intent comme le montre la figure 3.

```
Intent intent = new Intent(this,ActiviteADemarrer.class);
startActivity(intent);
```

Figure 3: Création d'un intent

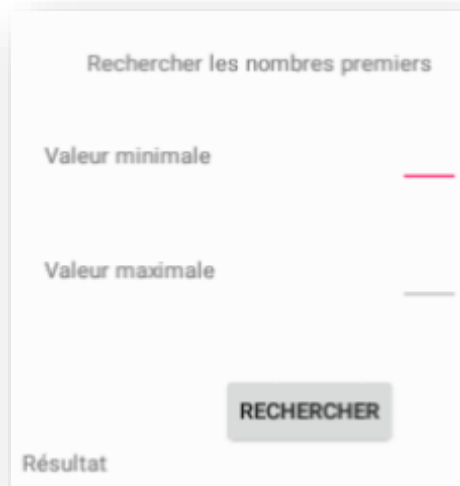


Figure 4: L'interface correspondante à l'activité «ComputeActivity»