

Exercice 3.1

Objectif

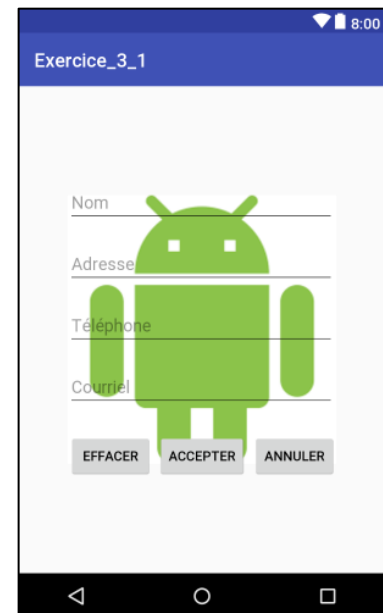
Se familiariser avec *Android Studio* et *ConstraintLayout*.

Problème

Créez une application *Android Studio* demandant à l'utilisateur d'entrer ses coordonnées :

Voici les spécifications de cette application :

- Les quatre champs de lecture doivent afficher un indicateur (*hint*) de ce que l'utilisateur doit y entrer. Ces champs doivent aussi imposer un format d'entrer approprié selon ce qui doit y être lu.
- Les quatre champs doivent être positionnés verticalement l'un sous l'autre, et les deux boutons doivent être alignés horizontalement sous les champs.
- Tous les widgets de l'activité doivent être centrés dans l'écran, affichés sur le logo d'*Android*.
- Des identificateurs significatifs doivent être attribués aux widgets (plutôt que `editText2`, `button2`, ...).
- Le bouton « Effacer » doit être fonctionnel : il doit effacer le contenu des quatre champs. Vous n'avez pas à programmer les deux autres boutons.
- Vous devez corriger tous les avertissements indiqués par *Android Studio* (identifiés par l'icône ⚠).



Matériel fourni

Récupérez de *eCité* le fichier d'image `android.png`.

À soumettre

Une fois l'exercice solutionné, démontrez à l'instructeur que ça fonctionne bien.

Suggestions

Voici les tâches à accomplir pour solutionner cet exercice :

1. Créer un nouveau projet *Android Studio* basé sur le gabarit « *Empty Activity* ».
2. Intégrez le fichier **android.png** à votre projet par copier-coller dans le répertoire `app » res » drawable24`.
3. Désactivez *Autoconnect* avant de concevoir votre interface.
4. Concevez l'interface en y déposant et configurant les widgets requis.
 - 4.1. Il est parfois difficile de sélectionner dans l'éditeur *Design* un widget spécifique. Dans ces circonstances vous pouvez utiliser le panneau *Component Tree* pour sélectionner des widgets.
5. Une fois tous les widgets adéquatement positionnés tel que demandé, utilisez le bouton *Infer constraints* pour attribuer les contraintes appropriées aux widgets
 - 5.1. Premièrement, attribuez des contraintes au `ImageView` afin qu'il demeure centré dans l'activité
 - 5.2. Ensuite, sélectionnez le reste des widgets (via le panneau *Component Tree*) et utilisez *Infer constraints* pour leur attribuer des contraintes
6. Vous constaterez probablement que la fonction *Infer constraints* ne donne pas des contraintes appropriées. Si c'est votre cas, vous devrez manuellement modifier les contraintes afin d'obtenir les résultats exigés.
7. Finalement, il ne reste qu'à programmer le bouton « Effacer » afin qu'il efface le contenu des quatre `EditText`
 - 7.1. Dans le code *Java* de la classe de l'activité, ajoutez la fonction membre suivante à cette classe :

```
public void effacerButtonOnClick(View v) {  
}
```

Notez que *Android Studio* va afficher un message d'erreur indiquant qu'il ne reconnaît pas le type `View`. Il vous suggèrera cependant quel `import` ajouter au fichier afin d'en obtenir la définition.

- 7.2. Un widget `Button` dispose d'un attribut **`onClick`** permettant de lui attribuer une fonction (de comme format ci-dessus) à exécuter lorsqu'il est touché. Attribuez la fonction ci-dessus à cet attribut du bouton « Effacer ».
- 7.3. Il ne reste plus qu'à compléter le code source de la fonction de sorte que le contenu des quatre `EditText` soient effacés lorsqu'on touche le bouton. Nous avons déjà vu en classe comment récupérer en *Java* un widget via la classe `R` et la fonction `findViewById()`. Une fois le widget récupéré, utilisez sa fonction `setText()`.