

# TP 1: Introduction au développement sous Android

Laurent Provot < laurent.provot@uca.fr >

Le but de ce premier TP sera d'appréhender les notions de bases pour créer sous Android Studio un projet simple pour une plateforme Android de type smartphone. Vous vous aiderez pour cela principalement de la documentation que vous trouverez à l'URL suivante :  $\label{lem:html} \text{html}$   $\frac{\text{developer.android.com/develop/index.html}$ 

#### **Objectif**

L'objectif du TP est de mettre au point un petite application de Quiz à laquelle on répond aux questions par «Vrai» ou par «Faux» (comme sur l'exemple ci-dessous).



L'application étant très simple, le but n'est pas de la terminer le plus rapidement possible, mais au contraire de bien prendre le temps d'explorer les possibilités d'Android Studio, les layouts disponibles sous Android et la documentation relative aux classes utilisées.

**Notions abordées :** Projet Android Studio, Layouts, Activity, Toast, Button, TextView, Listener, Classes Anonymes, resources (strings, drawables), LogCat, Cycle de vie d'une activité

#### **Préliminaires**

- 1. Créez un projet applicatif Android : vous l'appellerez Quiz, vous le mettrez dans un package fr.iut.pm (le reste n'est normalement pas à changer)
- 2. Créez une Empty Activity et nommez-là QuizActivity
- 3. Avant de rentrer dans le vif du sujet et de créer la vue du Quiz, commencez par manipuler et comprendre les différents *Layout* et leurs paramètres. Notamment, vous devez maîtriser le FrameLayout, le LinearLayout et le RelativeLayout (pas de ConstraintLayout pour l'instant). De même vous devez connaître les principaux paramètres qui leur sont liés, directement ou indirectement (en l'occurence orientation, gravity, layout\_width, layout\_height, layout\_gravity, layout\_weight)

### Mise en place de l'application de base

- 1. Créez la vue du Quiz en vous inspirant de l'exemple précédent (faites le ménage dans le fichier activity\_quiz.xml auparavant)
- 2. Expliquez comment (et où) sont gérés les textes qui apparaissent dans le GUI
- 3. Ajoutez différents textes pour les boutons Vrai / Faux ainsi qu'une question en dur
- 4. Explorez le code de la classe QuizAcivity pour comprendre commment le layout XML est relié au GUI
- 5. Profitez en pour lire la doc de la classe Activity et explorer le fichier de ressource R et comprenez comment fonctionnent les ID de ressources
- 6. Ajoutez un comportement à vos boutons pour qu'ils affichent un Toast (consultez la documentation pour plus d'infos, findViewById et classe Toast) affichant si le choix de la réponse est correct ou pas (vous mettrez dans un premier temps cela en dur dans votre code)

Testez votre application en la lançant sur un téléphone (vous pouvez aussi tester sur un émulateur avec un AVD Nexus, mais il faudra auparavant installer une image système car aucune n'est présente sur l'installation de base de l'IUT — à faire chez vous donc).

# Réfactoring

- 1. Gérer les questions en dur dans le GUI ne respecte pas vraiment les bonnes pratiques que nous vous avons apprises jusqu'à présent. Créez donc une classe métier TrueFalseQuestion pour gérer une question
- 2. Modifiez si nécessaire votre TextView pour qu'il possède un ID et soit ainsi manipulable dans le code
- 3. Ajoutez aussi un nouveau bouton centré sous les 2 précédents pour passer à la question suivante
- 4. Ajoutez différents intitulés de questions dans le fichier strings.xml
- 5. Ajoutez un ensemble de TrueFalseQuestion en dur dans votre QuizActivity en l'initialisant avec les questions correspondantes aux intitulés que vous venez de créer
- 6. Faites en sorte que votre TextView affiche la première question et que le bouton «Suivante» permette de passer à la question suivante
- 7. Ajoutez le code permettant de vérifier que la réponse donnée est correcte. Utiliser un Toast pour le signifier à l'utilisateur

(Avez-vous pensé à refactorer vos sources si un code identique est présent à différents endroits? Suivez-vous les bonnes pratiques qui vous ont été enseignées?)

# Étude du cycle de vie et persistance

- 1. Ajoutez une icône à votre bouton «Suivant» (res/drawable-xxx)
- 2. Lancez votre application, passez 1 ou 2 questions et faites une rotation de l'appareil. Que remarquez-vous ? Comment l'expliquez-vous ?
- 3. Ajoutez à QuizActivity une constante TAG dont la valeur est le nom de la classe
- 4. Surchargez les 6 méthodes du cycle de vie d'une activité en loggant le passage par ces méthodes (cf. classe Log)
- 5. Jouez avec le Home button, le Back button et faites des rotations de l'appareil, observez, déduisezen la réponse à la question du point 2.
- 6. Profitez-en pour manipuler un peu LogCat et testez ses fonctionnalités (notamment le filtrage).
- 7. Ajoutez un layout différent pour le mode paysage (res/layout-land) en faisant en sorte que le bouton «Suivant» soit collé en bas à droite.
- 8. Vérifiez que celui-ci est utilisé lors du passage en mode paysage.
- 9. Gérez la sauvegarde des données pour qu'après une rotation de l'appareil le quiz ne reparte pas de la première question.

Indice: onSaveInstanceState(Bundle outState) et onCreate(Bundle savedInstanceState)