université de BORDEAUX

Développement Android

Interface graphique

TD2

Dans la série de TDs à venir, nous allons développer une application de météo pour les stations de ski. Pour cela vous aurez accès a un service web donnant les informations suivantes :

— L'état de la station de ski (ouverte ou fermée)
— Le temps (beau, couvert ou neigeux)
— La température le matin (en degré celsius)
— La température l'après midi (en degré celsius)
— La vitesse du vent (en km/h)
— Le niveau de neige (en cm)

Après avoir lancé l'application, l'utilisateur doit pouvoir rentrer le nom d'une station de ski dans un champ texte puis valider à l'aide d'un bouton. L'application va alors interroger le service web pour récupérer les informations à propos de la station et les afficher a l'écran.

1 Création du projet

Lancer l'Android Studio et créez un nouveau projet. Nommez votre application comme vous voulez (mais évitez les noms bateau, il faut que votre application ait un nom différent de celui des autres). Poursuivez la création avec les options par défaut, puis choisissez `Empty Activity`, et enfin `Next` et `Finish`.

2 Design et implémentation de l'interface graphique

Avec les informations dont vous disposez, imaginez l'interface graphique de votre application. Typiquement il vous faudra un champ pour rentrer le nom de la station de ski, un bouton pour valider l'envoi de la requête et des champs texte pour les autres informations. Vous pouvez aussi placer un composant d'image, vide pour l'instant, que nous utiliserons plus tard pour indiquer si le temps est beau ou couvert.

Ouvrez le fichier de layout qui se trouve dans *res/layout*. C'est ici qu'est défini l'interface graphique de votre Activité principale. Essayez d'implémenter l'interface se rapprochant le plus possible de votre design. Utilisez les composants fournis dans l'API d'Android (*EditText*, *TextView*, *Button*, *ImageView*, etc.) pour définir votre UI. Vous pouvez utiliser des layouts tels que *LinearLayout* ou *RelativeLayout* pour placer les widgets correctement : dans un *RelativeLayout*, on indique la position des composants les uns par rapport aux autres (*p.ex.* en dessous de..., à droite de...), tandis que dans un *LinearLayout*, on les place séquentiellement, les uns après les autres (soit verticalement, soit horizontalement). Vous pouvez imbriquer plusieurs layouts pour créer une interface bien structurée.

3 Interaction avec l'interface graphique

La logique de votre application se trouve dans votre Activité principale (MainActivity.java). Ouvrez ce fichier. Une méthode est déjà surchargée : *onCreate*. Cette méthode est appelée à chaque fois que l'activitée est lancée, c'est donc là que nous allons initialiser certaines variables. Constatez, comme précédemment, qu'un appel à la méthode *setContentView* permet de lier l'activité à son interface graphique définie en XML. Il vous faut à présent interagir avec les widgets que vous avez déclaré.

→ Récupérez une référence à *chacun* de vos widgets grâce à *findViewById* (vu au TD 1). Veillez à ce que les variables correspondantes soient aussi accessibles à l'extérieur de `onCreate`.

Vous pouvez maintenant interagir avec chaque widget directement depuis le code java. En utilisant la documentation de la classe Button (*c.f.* `developer.android.com`), créez un listener pour détecter les clics de l'utilisateur sur votre bouton. Faites en sorte que lorsqu'un clic se produit, des données météorologiques factices s'affichent. Testez votre application en l'exécutant dans l'émulateur ou sur votre téléphone.

4 Jusqu'à la fin de la séance

Personnalisez votre interface graphique et donnez lui du style !

À rendre par courriel :

Une archive zip nommée `td2-nom1-nom2.zip` où `nom1` est le nom de famille alphabétiquement le plus petit de votre binôme, et `nom2` le suivant. L'archive doit contenir :

1. Votre fichier de layout `.xml`
2. Le fichier source de votre activité `.java`
3. Votre fichier strings `.xml`.