

Dans ce TD, nous commençons le développement d'une application d'information géographique. Nous allons le faire de manière graduelle, en ajoutant des fonctionnalités au fur et à mesure qui couvriront plusieurs aspects du développement Android (multi-activités, requêtes réseau, internationalisation, accès à des données locales ou distantes, géolocalisation, partage d'information entre applications, etc.).

Ce TD est moins guidé que le précédent. Vous devez vous servir d'Internet. Les exemples disponibles sur *Stack Overflow* sont souvent très pertinents, de même que la documentation officielle sur `developer.android.com`. Dans les deux cas, google est un bon point de départ.

1 Interface graphique (piste bleue : moyen)

Décidez d'un nom – si possible unique – pour votre application de géolocalisation.

1. Créez un nouveau projet d'application vide (*Empty*, pas *Blank*) que vous nommerez ainsi.
2. Ajoutez un champ texte editable pour saisir un nom de ville
3. Ajoutez un bouton en dessous
4. Lorsque le bouton est pressé, vous affichez le nom de la ville dans LogCat. (Pour détecter les clics, on utilisera l'attribut `onClick` dans le fichier XML du layout.)
5. Créez une seconde activité
6. Désormais, l'appui du bouton dans la première activité lance la seconde activité. (“*starting another activity*”.)
7. Idem, avec le nom de la ville passé en paramètre à la seconde activité
8. Concevez le layout de votre seconde activité de sorte à afficher :
 - (a) Le nom de la ville en titre
 - (b) Quelques informations relatives à cette ville : Pays, Région, Superficie, Population, Maire. Pour chacun, on veut en fait deux champs : un champ de libellé (p.ex. “Pays :”) et un autre champs qui contient la valeur correspondante. (Conseil : donnez-leur des noms utilisez des `LinearLayout` directement dans le code XML.)
 - (c) Le texte des libellés peut être rempli directement dans le layout XML, il ne changera pas.
 - (d) Pour les valeurs, créez une méthode qui remplit tous les champs depuis le code java. Pour l'instant, mettez des valeurs de votre choix. (Utilisez `findViewById` pour récupérer un composant graphique à partir de son identifiant.)
9. Donnez du style à vos deux activités.

2 Internationalisation (piste verte : facile)

Toutes les chaînes de caractères de votre application devraient être placées dans le fichier `res/values/strings.xml`. Cela vaut aussi bien pour les chaînes présentes dans le code java que pour celles dans le code XML.

Une fois les chaînes de caractère centralisées dans un même fichier, il est très facile de traduire l'application dans une autre langue. Faites une version en français et une autre en anglais à l'aide de ces fichiers (voir la page *Supporting Different Languages* sur developer.android.com).

Vous pouvez tester le résultat en changeant la langue du téléphone (ou de l'émulateur) dans les paramètres. Le résultat est-il conforme à ce que vous attendez ?

3 Orientation (piste rouge : non-guidé)

Si la séance (de deux heures) est terminée, vous pouvez passer directement au TD suivant. Sinon voici quelques améliorations à réaliser sur votre application.

1. Faîtes en sorte que votre application s'affiche correctement dans les deux orientations (portrait et paysage). Pour tester la rotation dans l'émulateur, selon les versions, c'est soit Ctrl+F11, soit Ctrl+F12, soit la touche 7 du clavier numérique déverrouillé.
2. Notez que les informations disparaissent lorsque l'écran est tourné. Comprenez pourquoi et corrigez ce problème.
3. Continuez à personnaliser le style de votre application jusqu'à la fin de la séance.

4 Divers

Si la séance (de deux heures) est terminée, vous pouvez passer directement au TD suivant. Sinon voici quelques améliorations à réaliser sur votre application.

1. Faîtes en sorte que l'application se souvienne de la ville lorsqu'elle est fermée puis relancée.
2. Étudiez le site Wikidata, que nous utiliserons la semaine prochaine.

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`.