

SYM

Labo01

1.

Il faut ajouter un dossier values-(country code) dans les ressources, contenant un fichier strings.xml.

Ceci permet d'avoir une application multi-lingue.

La langue par défaut est la langue contenue dans le fichier strings.xml sous le dossier "values".

2.

Il faut mettre l'image dans le dossier ressource "drawable". Physiquement il existe plusieurs dossier pour différentes résolutions.

Les images sont gérées selon la résolution du téléphone utilisé. Comme la version d'android ne change pas beaucoup d'un téléphone à l'autre contrairement au matériel (et notamment l'écran), il faut pouvoir supporter différentes résolutions pour l'application.

D'où la gestion via différents folders.

3.

Sans rien faire, l'action de cliquer sur le bouton lors de la 2ème activité fait quitter l'application, alors qu'elle devrait simplement nous ramener à l'écran de login. Afin de palier à ce problème, nous avons utilisé la fonction `OnBackPressed()` dans la `BasicActivity` (2ème activité) afin de lui dire de retourner à l'écran de login en cas de pression sur le bouton Back.

4.

Afin de faire ceci, au moment de créer l'activité, il ne faudra pas utiliser la fonction `startActivity()` mais plutôt la fonction `startActivityForResult()` ce qui nous permettra de récupérer une valeur par la suite dans la première activité. Dans la 2ème activité, nous pourrons faire les 2 fonctions `getIntent().putExtra("ResultValue", "0");` et `setResult(Activity.RESULT_OK, getIntent());` afin de dire que le résultat est une string comme clé "ResultValue" et valeur "0" et également lui dire que l'activité a correctement quitté (`RESULT_OK`). Ensuite, dans la 1ère activité, il suffira de handle le retour grâce à la fonction `onActivityResult()` qui nous permettra de récupérer toutes les informations que nous avons passé lors du retour de la 2ème activité.

5.

Il faut créer un second layout, en mode paysage (depuis la gui de android studio par exemple).

Ensuite ce layout doit être déposé dans un folder "layout-land" qui permet d'utiliser l'application en mode paysage et avoir un autre affichage.

6.

L'utilisation d'un RelativeLayout n'est pas très compliquée. Cependant il nous faut passer en mode text de l'éditeur d'interface, car l'éditeur graphique est limité. Pour utiliser le RelativeLayout à la place du LinearLayout il suffit de substituer le nom du 2ème par le 1er. De retour en mode graphique nous pouvons voir que le layout a changé, tout est en désordre. Il faut réaligner les éléments les uns par rapport aux autres car nous sommes en positionnement relatif.

7.

Voilà une brève explication du moment où chaque fonction est appelée : - onCreate(): Appelée au moment où l'activité est créée (mais non démarrée) - onStart(): Appelée au moment où l'activité démarre - onResume(): Appelée au moment où l'activité a été mise en pause et se relance (L'utilisateur retourne à une activité sur son écran) - onPause(): Appelée au moment où l'activité est mise en pause (Ex: une autre activité arrive sur l'écran) - onStop(): Appelée au moment où l'activité s'arrête (Elle n'est plus visible à l'écran) - onRestart(): Appelée au moment où l'activité redémarre (L'utilisateur navigue sur l'activité à nouveau) - onDestroy(): Appelée au moment où l'activité est détruite (L'activité est finie ou est détruite par le système volontairement, pour un crash par exemple)

Si nous utilisons des systèmes externes, nous pourrions utiliser la fonction onCreate() pour créer le lien avec le système et la fonction onDestroy() pour détruire ce lien quand l'activité est finie, ce qui éviterait ainsi des fuites de mémoire ou de laisser des liens vers des systèmes ouverts.

Authors

Yves Athanasiadis & Samuel Darcey