

TP Android : PPM 2012/2013

Etienne Renault et Harris Bakiras

L'objectif de ce TP est de fournir une introduction rapide aux concepts vus lors du cours Android. Ce TP fournit les éléments de base nécessaires à la réalisation du projet

1 Exercice 1 : Rating Bar App

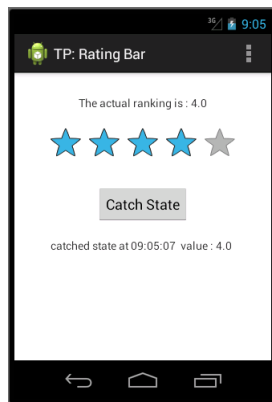


Figure 1: Vue principale Rating Bar App.

*Cet exercice constitue le classique “Hello World” légèrement élaboré pour pouvoir aborder les concepts liés à la rotation. Le but est de créer une application de notation via l'utilisation d'un *RatingBar*.*

Question 1. Tout d'abord nous souhaitons mettre en place une interface graphique minimaliste contenant deux champs texte (*TextView*), un bouton (*Button*) et un champ de notation utilisant les étoiles (*RatingBar*). Le premier champ *TextView* permet de connaître la valeur numérique associée au composant *RatingBar*. Le second champs *Text view* permet de connaître

l'heure de la dernière capture de la notation au sein du *RatingBar*. La valeur par défaut de ce dernier champ est "NEVER UPDATED". La Fig. 1 montre l'aspect global de cette activité.

Question 2. Que se passe-t-il lorsque l'on effectue une rotation de l'appareil (via ctrl-F12 sur le simulateur)?

Question 3. Faire en sorte que la rotation ne provoque pas la perte des données, ni ne change la date de dernière modification par l'utilisateur. Pour cela il faut utiliser la persistance légère au travers des méthodes *onSaveInstanceState*¹ et *onRestoreInstanceState*².

2 Exercice 2 : Mini Music Player

Cet exercice constitue l'occasion de manipuler les listes au travers d'une mini-bibliothèque statique de musique. Cet exercice permettra de manipuler différentes vue via l'utilisation d'intent, et la persistance longue.

Question 1. Construire la vue principale de manière statique pour les différents morceaux qui seront joués par le Player. Cette vue est une *ListActivity*. Au lieu de fixer en dur dans le code ce tableau, la philosophie Android vise plutôt à définir un *StringArray* *mymusic* par exemple dans le fichier *res/values/string*. La récupération se fait ensuite via:

```
getResources().getStringArray(R.array.mymusic);
```

Question 2. Construire l'interface graphique en XML qui sera utilisée pour pour la vue détaillée des différents morceaux (Fig. 3).

Question 3. Construire l'activité secondaire qui permettra d'associer des actions aux différents boutons et mettra à jour dynamiquement le nom de la chanson dans la vue détaillée. Pour passer des paramètres à une activité il faut utiliser les méthodes *getStringExtra*³ et *putExtra*⁴ de la classe *Intent*.

¹[http://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState\(android.os.Bundle\)](http://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))

²[http://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState\(android.os.Bundle\)](http://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))

³[http://developer.android.com/reference/android/content/Intent.html#getStringExtra\(\)](http://developer.android.com/reference/android/content/Intent.html#getStringExtra())

⁴[http://developer.android.com/reference/android/content/Intent.html#putExtra\(java.lang.String, android.os.Bundle\)](http://developer.android.com/reference/android/content/Intent.html#putExtra(java.lang.String, android.os.Bundle))

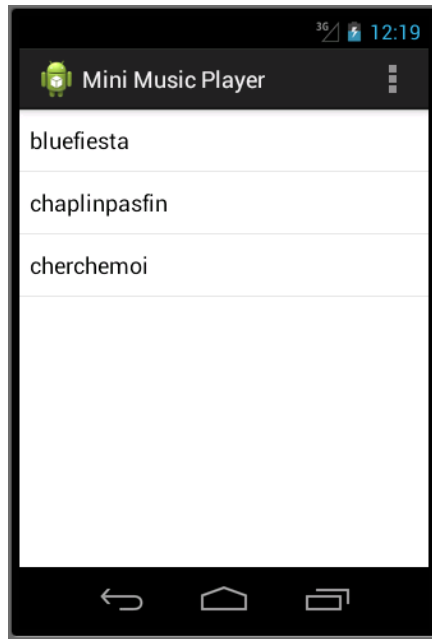


Figure 2: Vue principale

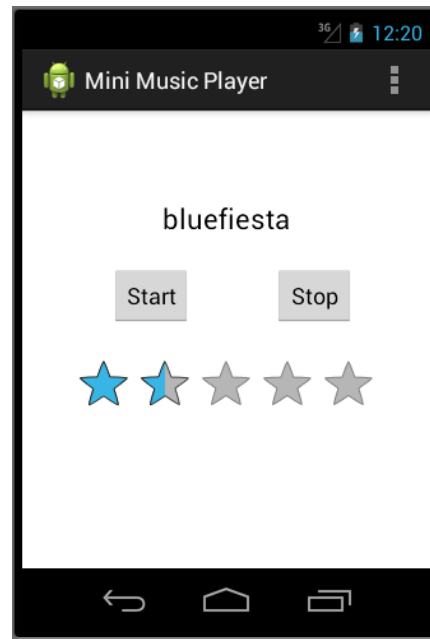


Figure 3: Vue détaillée

Question 4. Déclarer l'activité dans le fichier `Android-Manifest.xml`, puis implémenter l'évènement 'clic sur un item' qui se charge de lancer la vue détaillée de la chanson sélectionnée.

Question 5. Gérer la persistance temporaire lorsque l'application est tuée lors d'une rotation.

Question 6. Pour que l'application soit persistente, nous voulons pouvoir sauvegarder de manière durable (même si l'application est détruite) la valeur contenue dans le composant *RatingBar*. Pour cela il faut utiliser *getSharedPreferences*⁵.

Question 7. Que se passe-t-il lorsque l'on demande de jouer un morceau puis que l'on effectue une rotation sans avoir stoppé la musique? A quoi cela est-il du? Proposer une solution à ce problème.

⁵[http://developer.android.com/reference/android/content/Context.html#getSharedPreferences\(java.lang.String,int\)](http://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String,int))