Travail final (partie 2)

Spotify, Spotify, dis-moi qui est le plus … ?

Si vous avez ce document, c'est que vous avez réussi à récupérer une information dans votre activité, félicitations !

Dans ce document, on va parler brièvement de :

- Téléchargement d'images
- Séparation de la Vue et du Modèle
- Grille de correction sommaire

Téléchargement des images

Placer des images des artistes / albums choisis pour vos questions amène beaucoup de richesse à votre app. Téléchargez / afficher une image provenant d'un serveur amène des défis mais encore une fois Volley peut simplifier le processus.

- Utilisez des NetworkImageViews plutôt que des ImageViews (à éditer dans le xml de positionnement **com.android.volley.toolbox.NetworkImageView**
- On peut se servir du ImageLoader de votre singleton :
    - Récupérer la String représentant l'url de l'image à afficher dans la réponse de votre requête Volley / Spotify
    - Utiliser la méthode Volley setImageUrl de la classe NetworkImageView
    - setImageUrl prend 2 paramètres :
        - la String représentant l'image à afficher
        - l'imageLoader que vous pover récupérer de votre instance de votre Singleton (faire un get si nécessaire)

Pour plus d'infos :

https://cypressnorth.com/web-programming-and-development/setting-android-google-volley-imageloader-networkimageview/

https://stackoverflow.com/questions/65153571/spotify-api-error-401-no-token-provided

Séparation de la Vue et du Modèle

On a un singleton qui sépare partiellement la vue du modèle mais on peut aller plus loin. Différentes techniques sont possibles, en gardant en tête qu'on doit attendre que la réponse soit reçue pour afficher les résultats contenus dans la réponse dans la Vue. C'est un défi.

Techniques possibles :

1. Codez les requêtes et recevoir les réponses dans les activités, utiliser l'objet RequestQueue et ImageLoader du singleton → de base, n'obtiendra pas tous les 10 points consacrés à la séparation Modèle / Vue.

2. **Faire les requêtes / réponses dans une classe du modèle (une par question par exemple) et, à l'image du tp1, utiliser le contexte qu'on peut transtyper en Activité pour faire le lien entre les infos contenues dans la réponse et la vue → correct**

3. **Ajouter un écouteur qui mettra en œuvre notre propre interface qui sera appelée lorsque la réponse est reçue et qui permettra ainsi de mettre à jour la vue seulement lorsque la réponse est reçue → très bien, bonus possible**

   Exemple d'interface :

```
public interface RequeteTermineeListener {
        public void requeteterminee(JSONObjecresponse);
}
```

Exemple de classe mettant en oeuvre l'interface réponse (je l'ai fait dans une classe interne plutôt qu'une expression lambda) :

```
public class ObjRep2 implements Response.Listener<JSONObject>{
    private RequeteTermineeListener listener;
    public ObjRep2(RequeteTermineeListener listener)
    {
        this.listener=listener;
    }
    @Override
    public void onResponse(JSONObject response) {
        listener.requeteTerminee(response);
```

Exemple du code de l'activité (les requêtes sont lancées de l'activité) :

```
q.faireRequete(this, Request.Method.GET, url1, listener);

où q : objet du modèle
   faireRequete : méthode contenant la requête Volley
   listener : objet mettant en œuvre l'interface
RequeteTermineeListener
```


listener étant défini dans l'activité comme ceci :

```
RequeteTermineeListener listener = new RequeteTermineeListener() {
    @Override
    public void onRequestCompleted(JSONObject response) {

        //Afficher les résultats de la réponse dans vos widgets
        graphiques

}
```

Grille de correction sommaire

Questions - originalité	/5
Questions – aléatoires (on n'a pas tjrs les mêmes choix de réponse)	/5
Requêtes fonctionnelles (récupère	/15

informations de spotify)	
Images récupérées de Spotify	/10
Gestion des événements / afficher bonne ou mauvaise réponse à partir de résultats de requêtes	/15
Comptabiliser les bonnes réponses	/5
Animation ObjectAnimator	/10
Séparation entre le Vue et le modèle (efforts, autant que possible)	/10
Apparence des activités	/10
Utilisation du Singleton	/5
Stratégies de codage (pas de répétition, conventions)	/5
Commentaires pertinents	/5
Utilisation de GSON pour une question (ou plus)	+10