POLYTECHNIQUE
MONTRÉAL

Département de génie
informatique et génie logiciel

INF3405

Informatique Mobile

Hiver 2018 - Rapport du Travail Partique 1

Travail soumis à Mikaela Ngamboé

Par

Philippe Carphin 1693091

Mounia Nordine 1749881

Reph Dauphin Mombrun 1662298

1er mars 2018

Table des matières

1 Introduction

Dans le premier travail pratique d'informatique mobile, nous avons du apprendre à trouver des points d'accès WiFi et le placer sur une carte Google Map.

Pour ce faire, nous avons réalisé une application Android qui présente une liste des points d'accès disponibles ainsi que leurs informations et offre quelques autres fonctionnalités comme le partage et l'ajout de favoris.

Le rapport présente trois parties. La présentation technique montre l'interface de l'application et décrit son fonctionnement interne. Les sections suivantes présentent des difficultés rencontrées et des critiques et suggestions.

2 Présentation technique

Dans cette section nous présentons, d'une part, l'interface de notre application. Ensuite, nous présentons quelques classes de librairies importantes utilisées par l'application. Finalement, nous présentons les classes que nous avons écrites pour l'application elle même.

2.1 Interface utilisateur

L'écran d'accueil présente trois zones. La zone de carte géographique, la barre du haut et la barre de droite (voir figure 1). Initialement, la barre de droite présente une liste des réseaux disponibles. Cette liste peut être mise à jour en cliquant sur le bouton scan.

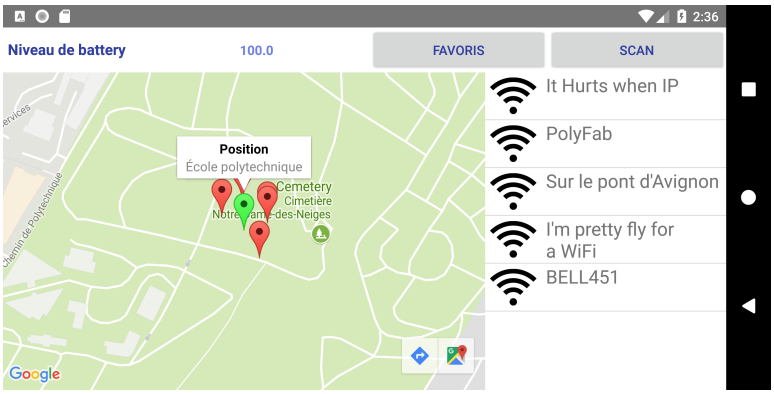

FIGURE 1 – Écran d'accueil de l'application

Lorsqu'on clique soit sur un marqueur de la carte ou su un élément de la liste de la barre de droite, les informations du point d'accès sélectionné sont affichées dans la barre de droite (voir figure 2).

La barre de droite contient aussi des boutons pour ajouter le point d'accès aux favoris ou le partager. Le bouton partager permettra de partager sur Facebook, Gmail ou un autre service installé sur notre appareil.

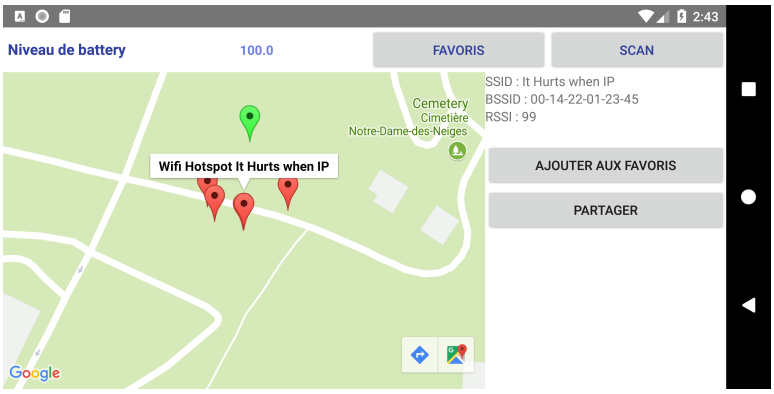

FIGURE 2 – Cliquer sur un marqueur ou élément de la liste fait apparaître les détails de celui-ci

Le bouton Ajouter Aux Favoris permettra d'ajouter le point d'accès à une liste de favoris. Cette liste est accessible en cliquant sur le bouton Favoris de la barre du haut.

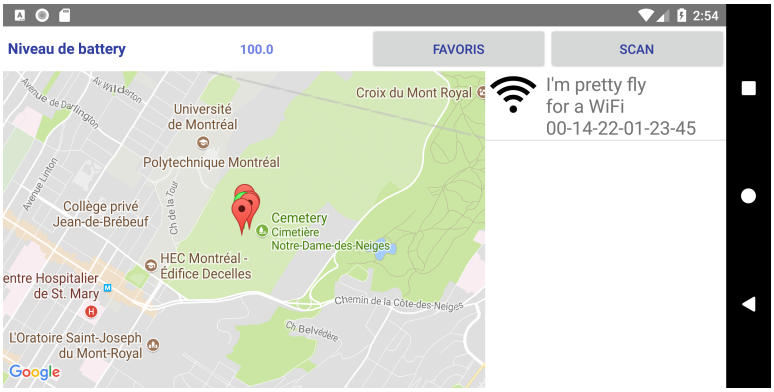

FIGURE 3 – Cliquer sur le bouton Favoris montre la liste des favoris

2.2 Classes de librairie

Les classes importantes utilisées sont la classe wifiManager et ScanResult pour la recherche des points d'acces WiFi à proximité. Pour la carte et l'API Google Maps, nous utilisons la classe GoogleMap et les classes GoogleMap.Marker et GoogleMap.MarkerOptions.

2.2.1 Classes utilisées pour le Wifi

La recherche de points d'accès est faite à l'aide de la classe `WifiManager`. Celle-ci demande les points d'accès disponible au dispositif avec la fonction `getScanResults()` qui retourne

une liste d'objet `ScanResult`.

Ceci requiert une certaine mise-en-place, notament la création d'un `WifiScanReceiver` et l'enregistrement de celui-ci avec la fonction `registerReceiver()` de la classe `Activity`.

2.2.2 Classes utilisées pour les Maps

La classe `GoogleMap` est notre point d'accès aux APIs de carte géographique de Google. On pose un fragment dans le `layout` de notre activté. Ceci demande que notre classe implémente l'interface `onMapReadyCallback` qui demande que la fonction ait une méthode `onMapReady()`.

2.3 Classes propres à l'application

Dans cette section nous présentons les classes créées pour notre application ainsi que leurs méthodes. Nous expliquerons aussi les choix de conception pertinents pour comprendre ces classes.

Notre application fonctionne avec NN classes principales. La plus importante est la classe `MapsActivity` qui est la classe maitresse de l'application.

Parmis ses attributs, les suivants sont les plus importants :

`mMap` Un objet de type `com.google.android.gms.maps.GoogleMap` qui nous offre la communication avec les service de Google et nous donne un carte géographique avec laquelle nous interagissons.

`PointsAcces` Une liste de points d'accès trouvés par `WifiManager`. La classe `PointAcces` est un adaptateur de la classe `ScanResult`.

`FragmentListePointsAcces` Un fragment qui hérite de `ListFragment` et qui s'occupe d'afficher les éléments de notre liste de `PointAcces`.

`FragmentDetailsPointsAcces` Ce fragment afficher les détails d'un `PointAcces` dans une vue et des boutons dans une autre vue. Un fragment de ce type est créé lorsqu'un point d'accès est sélectionné.

Ces classes coopèrent ensemble pour changer lorsqu'un point d'accès est sélectionné. La figure 4 résume les interactions qui arrivent lorsqu'un point d'accès est sélectionné soit en

cliquant sur un marqueur dans la carte ou en cliquant sur un élément de la liste.

On peut sélectionner un point d'accès en cliquant dans la liste ou en cliquant sur son marqueur dans la carte. Cette action cause un appel à la méthode `onPointAccesSelected()` de `MapsActivity`. Cette fonction remplace le fragment de droite par un fragment contenant les détails du point d'accès ainsi que quelques boutons pour permettre des actons.

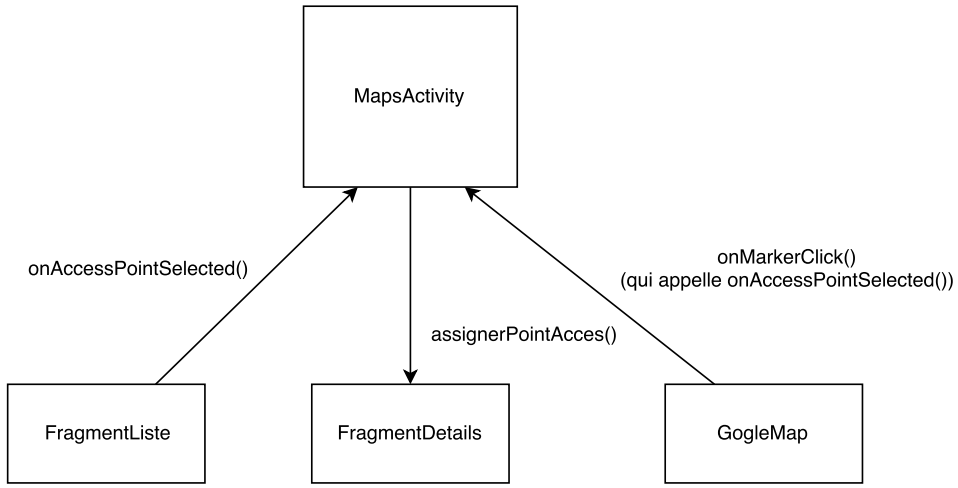

FIGURE 4 – Diagramme des interactions de sélection de point d'accès

3 Difficultés rencontrées

Les trois auteurs n'avaient pas d'expérience préalable avec le dévelopement Android. Nous avons donc eu une bonne mesure de difficultés. Cette section les énumère en détail.

3.1 Conflits de versions

« There's more than one way to skin a cat. »

Mark Twain (Connecticut Yankee in King Arthur's Court) 1889

L'écosystème Android est en constante évolution et la compatibilité vers l'arrière n'est pas toujours maintenue. Certaines classes de support nous permettent d'utiliser d'anciennes classes lorsqu'on en a besoin.

Par contre, étant donné notre inexpérience avec Android, nous avons du recourir à des tutoriels en-ligne. Il est arrivé souvent que solution donnée par un tutoriel ne fonctionne qu'avec les fragments de type `android.app.Fragment` alors on change le `import` au début du fichier.

Mais hélas, ceci cause des erreurs dans des bouts de code déjà écrits qui, sans qu'on le sache, utilisent des fragments de type `android.support.v4.app.Fragment`.

Heureusement, l'IDE Android Studio est très bon pour nous montrer ces erreurs et même souvent, il peut nous a proposé des solutions.

3.2 Accès à la BD de Google

Puisque nous n'avons pas accès à la base de donnée de Google qui fait correspondre des adresses MAC à des coordonné géographiques, nous avons du distribuer les marqueurs aléatoirement autour de l'emplacement de l'usager.

3.3 Méthodes de cycle de vie

Une des difficultés majeure de l'apprentissage d'Android est la compréhension des méthodes de cycle de vie des activités et fragments.

4 Critiques et suggestions

Voici quelques critiques et suggestions pour ce travail pratique. D'une part, il serait bien inviter les étudiants à lire de la documentation de base avant de commencer. Ensuite, on pourrait considérer d'autres plateformes de développement. Et finalement, laisser plus de liberté quant à la façon d'implémenter l'application.

4.1 Documentation suggérée

La documentation suggérée s'adresse à des gens qui sont déjà familiers avec le fonctionnement d'Andoid. Nous avons fait l'erreur de croire qu'elle serait suffisante. Nous avons donc commencé à coder notre application avec une compréhension déficiente. Ceci a mené à une

architecture mal conçue. Nous aurions du commencer par lire quelque guides de la page Android Developer Guides Google (2018a).

Par exemple, nous avons consulté les guides Android Lifecycle Methods Google (2018c) et Android Fragments Guide Google (2018b) mais nous avons consulté ces guides après avoir écrit beaucoup de code.

Il aurait été bien de conseiller fortement aux étudiants impatients de commencer par ces guides.

4.2 Plateforme de développement

Il aurait été intéressant de permettre d'utiliser d'autres plateformes comme React-Native ou Xamarin pour le développement de l'application. Nous comprenons que la chargée de laboratoire doit être familière avec la plateforme utilisée pour pouvoir corriger notre TP donc il est normal que cette suggestion ne puisse pas être prise en compte.

4.3 Fragments

Pour des néophytes d'Android, l'utilisation de fragments est plutôt difficile à apprendre. Il aurait été peut-être plus facile de réaliser l'application demandée en utilisant plusieurs activités.

5 Conclusion

Dans ce TP nous avons acquis beaucoup de connaissances sur le fonctionnement d'Android. Les erreurs qui nous ont causé du trouble ne serons pas répétées dans la prochaine application.

Nous avons aussi approfondis nos connaissances sur les APIs de GoogleMaps et WifiManager.

Références

Google. (2018a). *Android developper guides.* Consulté sur `https://developer.android.com/guide/index.html`

Google. (2018b). *Android fragments guide.* Consulté sur `https://developer.android.com/guide/components/fragments.html`

Google. (2018c). *Android lifecycle methods.* Consulté sur `https://developer.android.com/guide/components/activities/activity-lifecycle.html`