ENSSAT
LANNION

LOGICIEL EMBARQUE

Android

IMR3 — 2016/2017
REGNAULT Antoine
NANTEL Maëlig

Sommaire

1. Introduction: application “GeoQuest”

L’objectif de ce projet est de développer un jeu de piste pour Android. L’application consistera en une interface graphique qui accompagne l’utilisateur durant sa recherche d’indices (sous la forme de message et/ou de photos) qui lui permettent d’avancer progressivement sur un itinéraire balisé par des coordonnées GPS vers une destination finale. L’utilisateur pourra également créer de nouveaux itinéraires et y ajouter des balises. Les données devront être persistantes d’une exécution à l’autre de l’application.

2. Conception initiale

Cette section présente de manière très brève la conception de l'application en phase de lancement du projet. Comme nous le verrons dans la section 3 de ce document, les solutions retenues ont évolué durant la phase de développement.

2.1. Affichage du fond de carte

Pour permettre à l'utilisateur de visualiser sa position, il est nécessaire de pouvoir afficher un fond de carte. Il existe deux principales solutions pour cela :

- Google Map API
- Open Street Map

OpenStreetMap étant une solution open source et libre de droits, c'est vers elle que nous nous sommes tournés. Pour son utilisation, nous avons décidé d'utiliser la librairie OSMDroid (déjà utilisée lors d'un des TP).

2.2. Modélisation d'un itinéraire

Une fois la carte affichée dans l'application, il est nécessaire de pouvoir représenter un itinéraire. C'est le parcours que devra suivre l'utilisateur de l'application pour naviguer de balise en balise. Il existe plusieurs formats textuels permettant de représenter les itinéraires et nous avons décidé d'utiliser le JSON. Nous utiliserons plus précisément le format GeoJSON. Ce dernier permet de normaliser la représentation de données géographiques. Nous l'avons déjà manipulé dans le cadre d'un autre module.

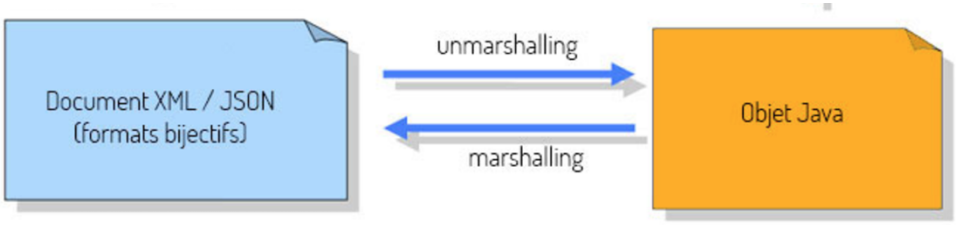
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "name": "Stade de Bel Air",
        "indice": "Un espace sportif après une longue montée depuis le
        "image_b64": "
        "image_url": "http://www.letelegramme.fr/ar/imgproxy.php/PhotoI
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          -3.4769153594970703,
          48.72391072616036
        ]
      }
    },
    {
      "type": "Feature",
      "properties": {
        "name": "ENSSAT",
        "indice": "Une prestigieuse école de génis",
        "image_b64": "
        "image_url": "http://www.letelegramme.fr/ar/imgproxy.php/PhotoI
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          -3.46268892288208,
          48.729770708236785
        ]
      }
    },
```


Figure 1 — Exemple d'un itinéraire en GeoJSON

Pour chaque balise du jeu, nous allons renseigner :

- Un indice textuel
- Ses coordonnées au format GPS (longitude et latitude)
- Une image associée à l'indice (encodée en base 64 dans le JSON)

Pour pouvoir exploiter les données GeoJSON, il est nécessaire de pouvoir les convertir en objets Java. Pour cela, nous avons décidé d'écrire un parseur en utilisant la librairie Jackson (version 2).



En mode édition, l'utilisateur pourra créer ou modifier un itinéraire. Pour cela, il devra sélectionner des points sur la carte puis renseigner un nom, un indice et éventuellement une photo. L'ordre des balises dans l'itinéraire sera fixe et dépend de l'ordre d'ajout dans l'éditeur.

3. Réalisation

3.1. Évolution de la conception

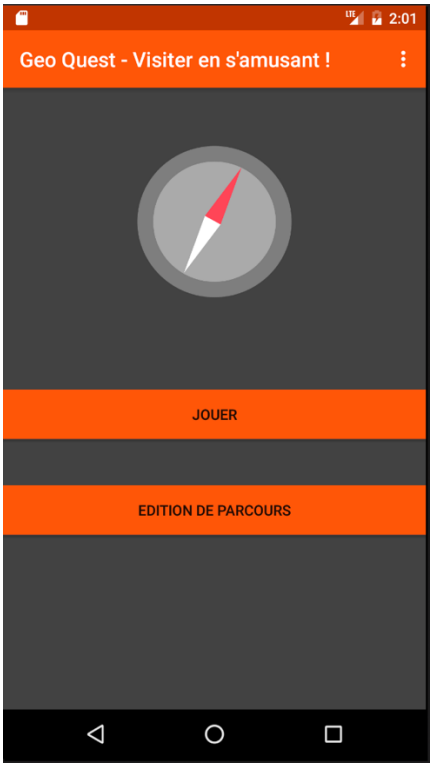
Après quelques tests, nous avons finalement décidé d'utiliser Google Map et non OSMDroid (OpenStreetMap). La principale raison est que nous allons avoir besoin d'utiliser l'API Google Map sur une application Android dans le cadre de notre projet de Génie Logiciel. Nous avons donc souhaité appréhender cette solution.

De même, nous nous sommes rendu compte que le format GeoJSON apportait beaucoup trop de lourdeur comparée à ce qui était réellement nécessaire. Nous avons donc utilisé un autre format de JSON :


```
{
  "beacons": [
    {
      "coordinates": {"latitude": 48.731483...},
      "hintString": "La Mie Câline <3",
      "hintImage": "/9j/4AAQSkZJRgABAQEBLAEsAAD/2wBDAA
    },
    {
      "coordinates": {
        "latitude": 48.72983319999999,
        "longitude": -3.4625674000000117
      },
      "hintString": "Prestigieuse école d'ingénieur"
    },
    {
      "coordinates": {
        "latitude": 48.75758922536711,
        "longitude": -3.455328941345215
      },
      "hintString": "La tour du Mordor",
      "hintImage": "/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAA
    }
  ],
  "name": "Example"
}
```


3.2. Résultats obtenus

Ci-dessous les principaux écrans de l'application et quelques explications sur leur fonctionnement. L'application a été testée sur un LG Nexus 5X, un LG Nexus 4 et une tablette ASUS Nexus 7.



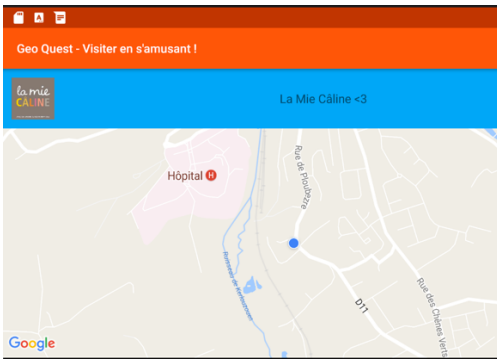
Lors du lancement de l'application, l'utilisateur est invité à choisir entre le mode « Jeu » et le mode « Édition de parcours ».

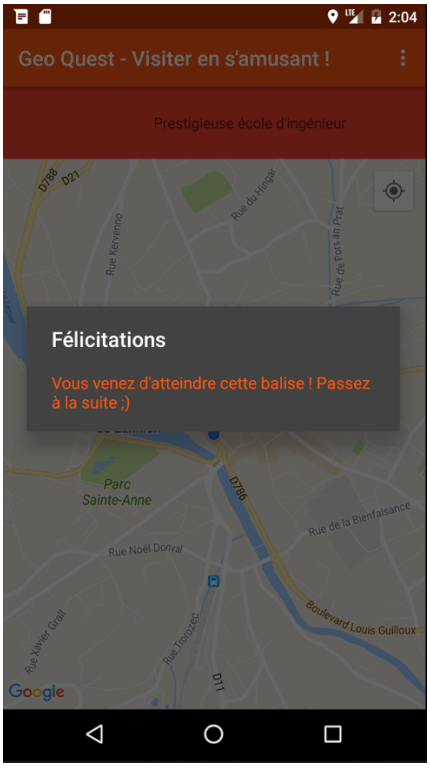
Cette activité est la seule de l'application à forcer l'utilisation du mode portrait.

En mode jeu, l'utilisateur visualise la carte (Google Map). Il peut visualiser sa position avec un point bleu. En haut de l'écran se trouve l'indice de la balise courant (texte) et éventuellement une image.

La couleur du fond contenant les indices évolue en fonction de la distance de l'utilisateur à la balise :

- Rouge à moins de 250 mètres
- Orange entre 250 et 600 mètres
- Bleu à plus de 600 mètres

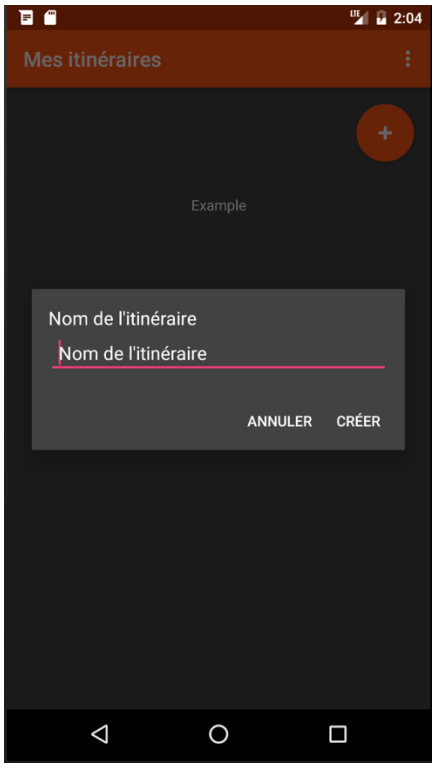




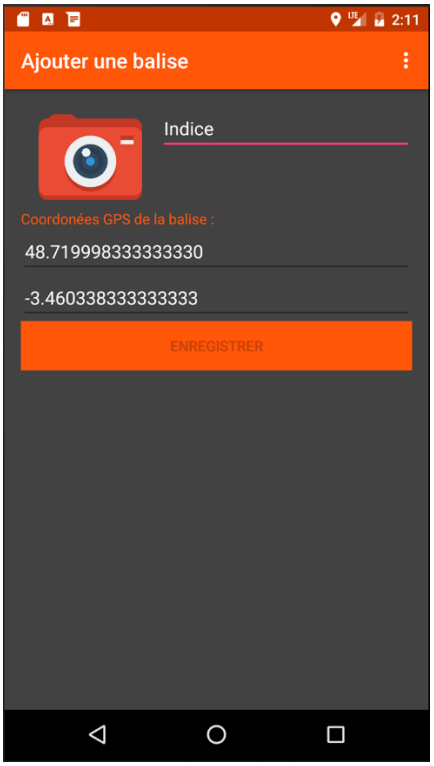
Lorsque l'utilisateur atteint les coordonné
balise courante, un modal (alert dialog)
affiché pour le féliciter. Un « touch » en de
ce modal le fermera et permets de relanc
avec la nouvelle balise.

Pour des raisons de précisions lié
coordonnées GPS, une marge de 20 m
ajoutée (on considère une balise comme
si l'utilisateur se trouve à moins de 20
d'elle).

L'application gère un mode « mutli-scénarios » pour l'expérience de jeu. Un utilisateur peut donc facilement en créer de nouveau (saisie uniquement d'un nom).



Il existe un itinéraire fourni par défaut dans l'application (« Example »). À partir du moment où l'utilisateur a créé au moins un autre scénario, il n'est plus possible d'accéder à l'itinéraire « Example ». Les itinéraires de l'utilisateur sont stockés en JSON dans le répertoire « geoquest » de la mémoire utilisateur (SD CARD).

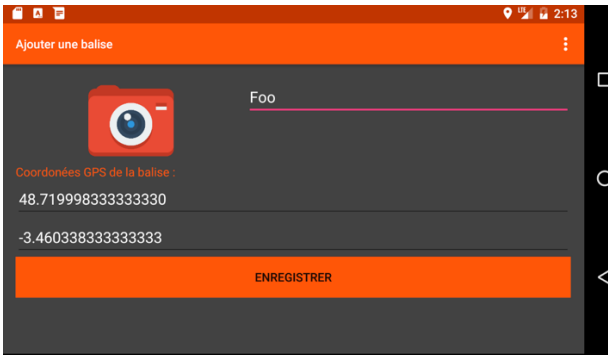


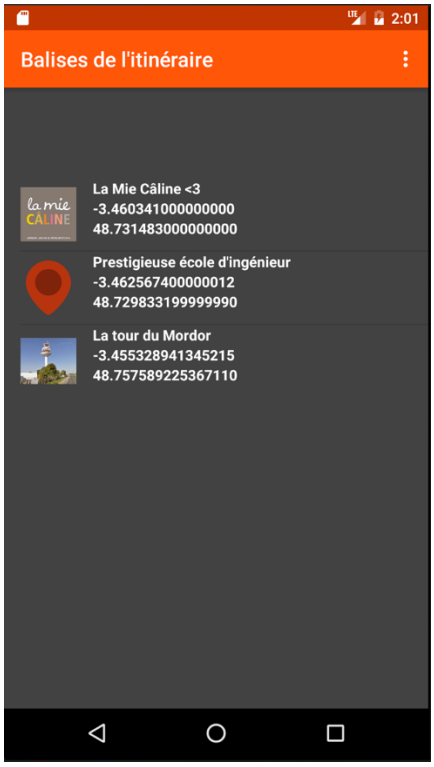
Pour ses itinéraires, l'utilisateur peut ajo
nouvelles balises. Il n'est pas possible d'ajc
balise à l'itinéraire par défaut « Example »
donc obligatoirement renseigner un indice
ainsi que les coordonnées GPS (longit
latitude).

Les champs longitude et latitude
automatiquement renseignées au fur et à
que les coordonnées de l'utilisateur évolue

L'utilisateur a la possibilité, s'il le s
d'ajouter une image à la balise. La pris
photo est réalisée à l'aide de l'appareil pl
terminal.

Le bouton d'enregistrement n'est disponi
si les champs obligatoires (indice, longi
latitude) sont non vides.





Pour chaque itinéraire, l'utilisateur peut visualiser la liste des balises. Il visualise ainsi les différents éléments de chaque balise (photo si présente, indice, coordonnées). Si une balise n'a pas d'image, un affichage par défaut est proposé.

Il peut accéder à l'ajout d'une nouvelle balise s'il ne s'agit pas de l'itinéraire par défaut.

L'ordre des indices dans l'itinéraire n'est pas configurable. Il est défini en fonction de l'ordre d'ajout dans l'éditeur (ou de définition dans le JSON de l'itinéraire par défaut).