

Elise DECOURVAL

Roman KRYVCHENKO

“Fitness Trail Assistant” Project

Jacques Klein, Tegawendé F. Bissyandé, and Médéric Hurier



Table of contents

Introduction	3
1. Environment	3
1. Achievement of the application	4
1.1. Analysis	4
1.1.1. Mode Predefined Route	5
1.1.2. Mode Free Trail	7
1.2. Application design	7
1.2.1. Geolocation.....	7
1.2.2. Way.....	8
1.2.3. Exercise	8
1.2.4. Distance	9
2. Results and Perspectives	9

Table of figures

FIGURE 1 : SDK MANAGER

FIGURE 2: STEP CONCEPTION APPLICATION

FIGURE 3 : DIAGRAM WELCOME ACTIVITY

FIGURE 4 : SCREENSHOT WELCOME ACTIVITY

FIGURE 5 : DIAGRAM MAPS ACTIVITY

FIGURE 6 : SCREENSHOT MAPS ACTIVITY

FIGURE 7 : SCREENSHOT MAPSWAY ACTIVITY

FIGURE 8 : SCREENSHOT TRAINING EXO ACTIVITY

FIGURE 9 : SCREENSHOT FREE TRAIL ACTIVITY

FIGURE 10 : SCREENSHOT FREE TRAIL ACTIVITY 2

Introduction

The main objective of the “Fitness Trail Assistant” project is to develop an Android application, by reusing as much as possible the concepts, methods and tools presented in the Principles of Software Development course. The goal of this trail assistant is to help trailer to track their activities, but also to give training instructions.

The trail assistant proposes two modes:

1. Predefined Route: the trailer runs following a route which is already defined in the app. In other words, the app knows what the itinerary is and indicates this itinerary to the trailer.

2. Free Trail: The trailer decides by him (her) self the itinerary.

This project allows us to explore the development of android application. For that we had a lot of help during our teacher course but also a lot of tutorials on the internet.

Our challenge was to achieve an innovative project, useful, related to current events and meets the school's project criteria.

1. Environment

For this project we worked with Android Studio provides an API that panel. Thanks to our SDK Manager we have download a lot of tool which allow us to facilitate the conception of our application (figure 1). The most important tool is Google Play Service, indeed, we will see that this API offer a lot of functionality really interesting.

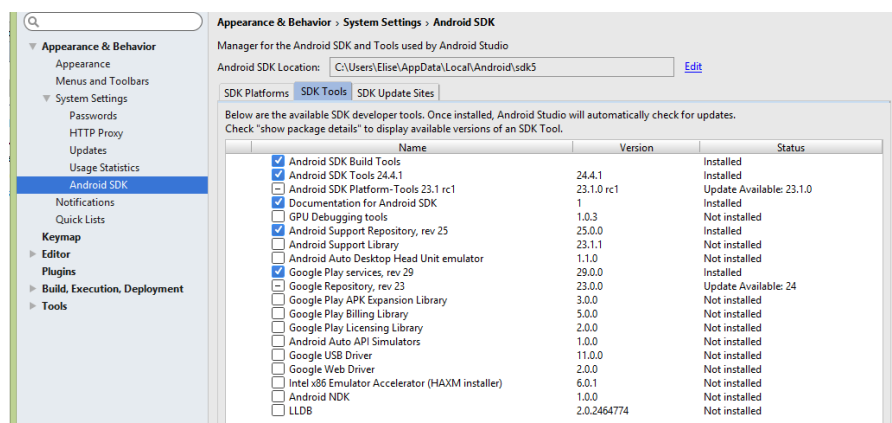


FIGURE 1 : SDK MANAGER

We did not use emulator, namely virtual machine because we have Android device which permit to directly test the evolution of our application.

Our project was also directly update in our git repository:

https://github.com/elisedcvl/appDecourval_Kryvchenko.git

1. Achievement of the application

The conception of our application was made in several step, which you can see on the figure below:

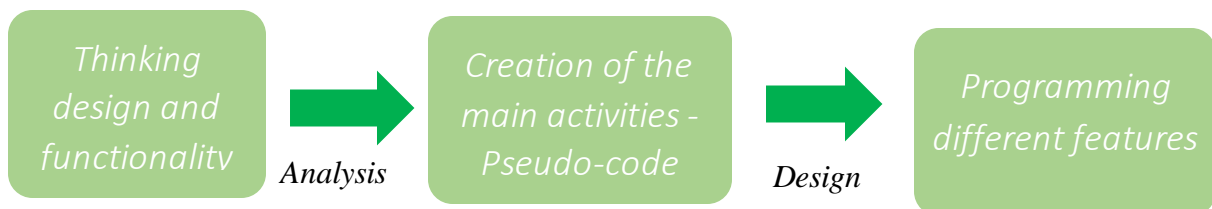


FIGURE 2: STEP CONCEPTION APPLICATION

1.1. Analysis

Concerning the analysis. Our application has 4 main step.

The first Step of our application is the **connection** of the user thanks to its Google Play account, *mainActivity*. It's a very important step because in the application we use several Google Services which can work only with a Google account.

The second step was configuration and programming for obtaining a **Google map**. This step was quite complicated because we have never used such tools. In addition many configuration are also done on the google site to obtain authorization.

The next two steps involve the two main ways that our proposed application. Namely, **Predefined route** and **Free Trail**.

Once the user is connected to this Google account, he is led to the *welcomeActivity*, where is proposed the two principal modes.

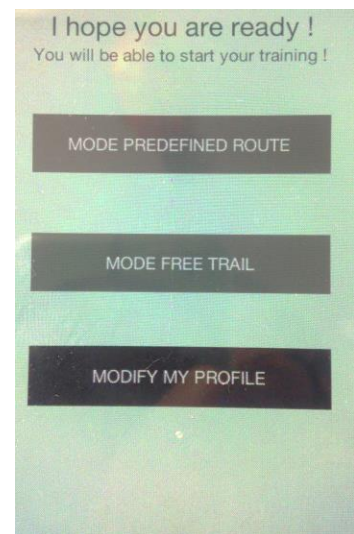
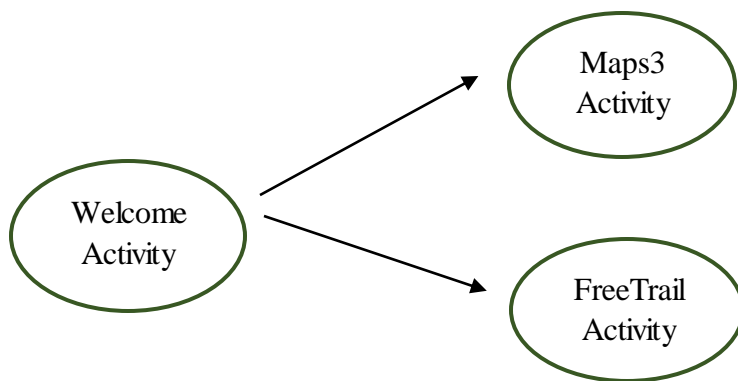


FIGURE 3 : DIAGRAM WELCOME ACTIVITY - FIGURE 4 : SCREENSHOT WELCOME ACTIVITY

1.1.1. Mode Predefined Route

If the user selects the mode Predefined Route he is led on the *maps3activity* who he can see his geolocation. He has to write the place where he wants to go and he has to do a choice between, leave from his current location or from another address. For do this choice he can tick or note the checkbox "Leave from this place".

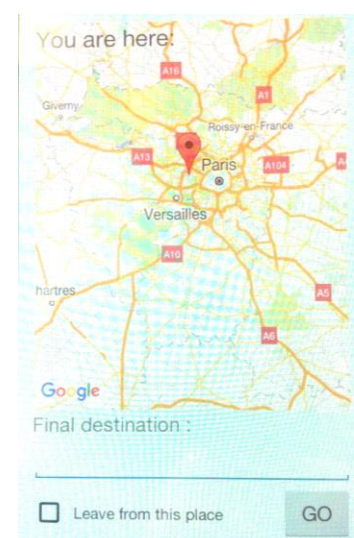
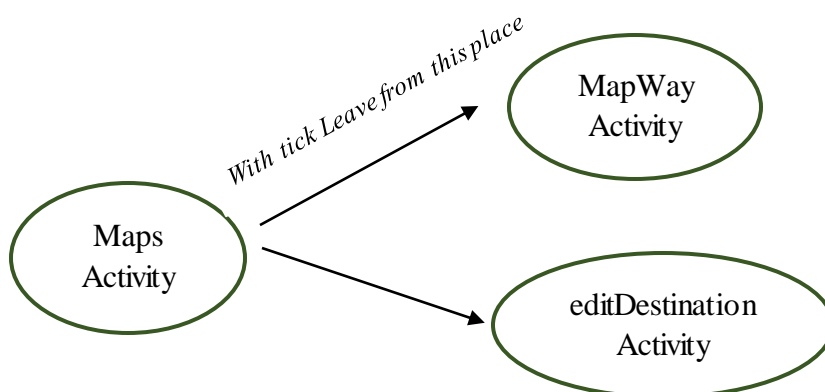


FIGURE 5 : DIAGRAM MAPS ACTIVITY - FIGURE 6 : SCREENSHOT MAPS ACTIVITY

If the user do the choice to leave from another place he is led on the Edit destination Acitvity where he has to put his departure address. After this step the application shows the way from its starting address or current location to its destination address.

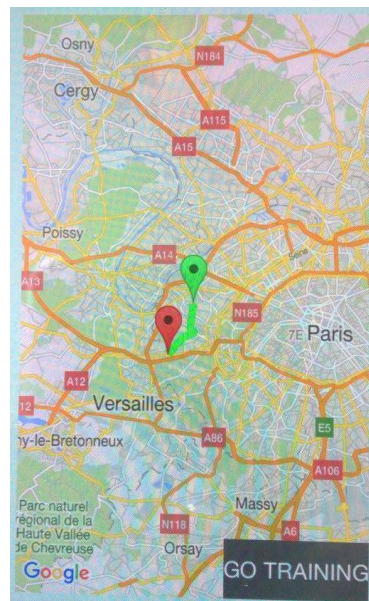


FIGURE 7 : SCREENSHOT MAPSWAY ACTIVITY

Once the user has seen the way he has to click on the button “Go training” and he can see his sports exercises with the distance. The total distance is calculated its starting address or current location to its destination address. It also has available a stopwatch for timing performance.

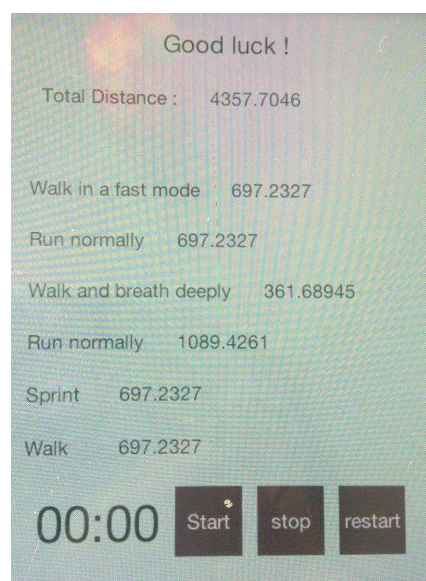


FIGURE 8 : SCREENSHOT TRAINING EXO ACTIVITY

1.1.2. Mode Free Trail

If the user do the choice Free Trail, first he can see his location. Secondly with the activity FreeTrail 2 he can see his time and the distance he has already travelled.

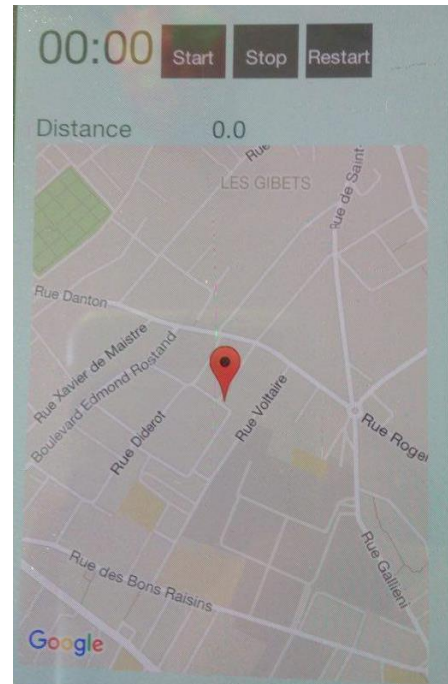
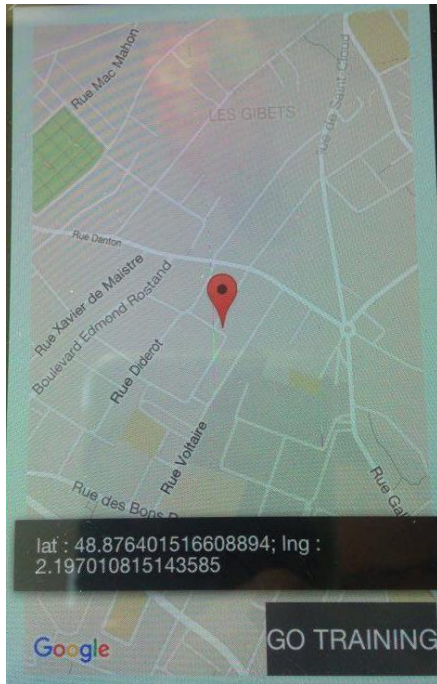


FIGURE 9 : SCREENSHOT FREE TRAIL ACTIVITY - FIGURE 10 : SCREENSHOT FREE TRAIL ACTIVITY 2

1.2. Application design

In this part I will clarify certain points of the design.

1.2.1. Geolocation

Concerning the user's geolocation used in several activity, our activity extends from *locationListener*. This class is used for receiving notifications from the *LocationManager* when the location has changed. "These methods are called if the *LocationListener* has been registered with the location manager service using the request *LocationUpdates(String, long, float,*

LocationListener) method.” Therefore, we have several function which permit to update the location when the user change his place.

In our layout we built a fragment that uses the following service :

“*com.google.android.gms.maps.MapFragment*”.

Of course we have added the following permission in our *AndroidManifest* to allow the application to use internet, wifi and geolocation

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="fr.rolandl.blog_gps.permission.MAPS_RECEIVE"
/>
```

1.2.2. Way

In the mode predefined route we show the way to the user. For calculate the itinerary from the starting point to the final point we use an **Asynchrone** Task. An asynchrone task allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.

There are several step in this java class. First we create an URL by concatenating the different addresses for the desired URL. Then we call the google service which allow to find a way between two points, thanks to our previous URL. Afterwards, we have the data processing.

Also We have two functions (*decodePolylines* and *onPostExecute*) wich allow to draw and show the itinerary.

1.2.3. Exercise

The process to dictate the user exercises is very simple. We calculate the total distance of just his route and carve the distance with the proportions we wanted for each type of exercise.

1.2.4. Distance

For calculate the distance between two point, first we have created two functions to retrieve the latitude and longitude from a given address.

```
public double getLatFromAddress(String strAddress){
    Geocoder coder = new Geocoder(this);
    List<Address> address;
    double lat =0;
    try {
        address = coder.getFromLocationName(strAddress, 5);
        Address location=address.get(0);
        lat = location.getLatitude();
        location.getLongitude();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return lat;
}
```

Once the latitude and longitude are recovered, it is very easy to get the distance through function :

```
Location.distanceBetween(latD, lngD, latA, lngA, dist)
```

This function give the distance with a float.

2. Results and Perspectives

Regarding the outcome of the application, I know that there would be a lot of improvement to do to have the application I had in my imagination. However, I'm still happy with the result of this application as it was very difficult for me. I spent a lot of time, and I know that the record could have been better but as a beginner I lose a lot of time on small things.

The main improvement that I have desired to do was to save the profile to submit exercises adapted to his profile. I started, but recording or recovering my data has never worked. I spent a lot of time but had to give up after a lot of testing. I left the traces of the various tests (TestBDUser, TestDataBaseUserActivity, User, UserBDD, UserDataSource, UserDBAdaptateur)

Another thing was obviously to make test. But I did not fully understand the course given by our professor, even by rereading several times. I regret that there is very little documentation on the internet because it could help me.