Partie MAP

QIU Mingming

Partie MAP....1
Utilisation de GoogleApiClient pour current location....2
utilisation de JSON format et URL format....4
Obtention d'un route entre deux positions....4
Change de mode de sortir....5
Fonction de localisation(l'origine)....7
Réussite de bouton des mode de sortie....10
Combinaison le projet de TestForButtonDeSortir et le projet d'obtention mon position actuelle....11
Réalisation de la navigation avec la destination écrite....12
Combinaison le mode de sortir avec la fonction de navigation....13
La destination....14
Méthode de cliquer sur la map....14
PlaceAutoComplete....15
La version finale pour la combinaison des fonction....17
La version d'obtenir la valeur de la position actuel....17
Affichage le temps nécessaire à partir de l'origine à la destination....17

Utilisation de GoogleApiClient pour current location

Pour decider la current position, on peut utiliser la fonction de “GoogleMap.setMyLocationEnable(true)” pour afficher l’icon qui a représenté la position de celle en haut à droite, mais le problème est que on peut pas obtenir la position de “latitude” et “Longitude”, selon ce document, on peut savoir d’utiliser la fonction de location API, alors accordez ce document et celui, on peut obtenir les codes sur obtenir la position précisée de current position.[code]

Interprétation des codes:

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        .....
        if (mGoogleApiClient == null) {
            mGoogleApiClient = new GoogleApiClient.Builder(this)
                    .addConnectionCallbacks(this)
                    .addOnConnectionFailedListener(this)
                    .addApi(LocationServices.API)
                    .build();

            mGoogleApiClient.connect();
        }
        }
```

Définir une “instance”, et mGoogleApiClient.connect(); pour commencer la fonction de GoogleApiClient.

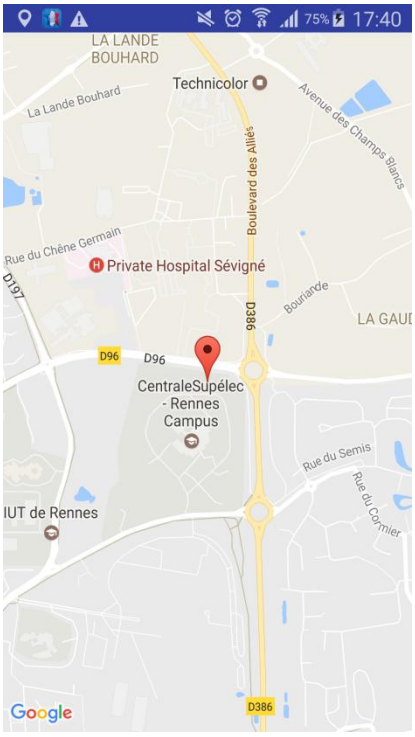
```
  public void onConnected(Bundle connectionHint) {

        if(ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATION)!=
PackageManager.PERMISSION_GRANTED&&ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //       ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //     public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                            int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);

        if (mLastLocation != null) {      /*this justification should be placed here, but not in the onMapReady()*/
            mLatitudeText=mLastLocation.getLatitude();
            mLongitudeText=mLastLocation.getLongitude();
```

```
                LatLng mLatlng=new LatLng(mLatitudeText,mLongitudeText);
                Marker mMarker = mMap.addMarker(new MarkerOptions()
                        .position(mLatlng)
                        .draggable(true));
/*
                CameraUpdate update = CameraUpdateFactory.newLatLngZoom(mLatlng, 15);

                mMap.moveCamera(update);*/

               gotoLocationZoom(mLatitudeText,mLongitudeText,15);
            }
        }
```

Définition pour les paramètres:

```
private final static int CONNECTION_FAILURE_RESOLUTION_REQUEST = 9000;
private GoogleMap mMap;
private GoogleApiClient mGoogleApiClient;
private Location mLastLocation;
private Marker mMarker;
private double mLatitudeText;
private double mLongitudeText;
```

La photo de cette fonction:



Si on utilise seulement “mMap.setMyLocationEnabled(true);” on peut pas obtenir la position avec les données précisément.

utilisation de JSON format et URL format

pour obtenir les informations des routes entre deux destination, alors on va utiliser les connaissances de ce document, donc on a cette fonction de “private String getDirectionsUrl(LatLng origin, LatLng dest) {...}”

Obtention d’un route entre deux positions

Selon notre diagramme, on va dessiner un toute entre deux positions(de mon current position à celle que j’ai importé dans la boîte d’entrée), pour assurer chaque step est correct et éviter la situation où il sera difficile de trouver les problèmes si on écrit les codes de réaliser toutes les fonctions d’abord, alors j’ai choisir deux positions, un est celle de mon position

```
LatLng origin = new LatLng(mLatitudeText, mLongitudeText);
```

, une autre est quelconque lieu que j’ai choisi dans le map comme la destination:

```
“LatLng dest = new LatLng(48.144996, -1.596103);”
```

Et j’ai ajouté un marker sur cette position:

```
“mMap.addMarker(new
MarkerOptions().position(dest).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLU
E)).draggable(true));”
```

Et alors en utilisant la fonction de

```
String url = getDirectionsUrl(origin, dest);
```

Je peux obtenir la route en forme de URL, et après il est nécessaire de la transformer en JSON

```
“DownloadTask downloadTask = new DownloadTask();
// Start downloading json data from Google Directions API
downloadTask.execute(url);”
```

Alors j’ai obtenu ce diagramme:

Et je peux changer le mode de sortir en changant “mode”, dans mon plan, je vais utiliser seulement “mode=walking”,”mode=driving”,”mode=bicycling”,la phrase complet est

```
private String getDirectionsUrl(LatLng origin, LatLng dest) {
......
String parameters = str_origin + "&" + str_dest + "&" + str_mode;

// Output format
String output = "json";

String url = "https://maps.googleapis.com/maps/api/directions/" + output + "?" + parameters;
......
}
```

Change de mode de sortir

Dans la dernière partie, je peux changer que dans la phrase, pour changer directement, je vais ajouter trois boutons pour changer dans le map directement
J’ai ajouté trois boutons dans activity_XML, avec les codes suivants:

```
<Button
      android:id="@+id/bicycling"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="bicycling"
      android:layout_margin="10dp"

      />

  <Button
      android:id="@+id/walking"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignTop="@+id/bicycling"
      android:layout_marginLeft="17dp"
      android:layout_marginStart="17dp"
      android:layout_toEndOf="@+id/bicycling"
      android:layout_toRightOf="@+id/bicycling"
      android:text="walking"
    />

  <Button
      android:id="@+id/driving"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignTop="@+id/walking"
      android:layout_marginLeft="34dp"
```

```
      android:layout_marginStart="34dp"
      android:layout_toEndOf="@+id/walking"
      android:layout_toRightOf="@+id/walking"
      android:text="driving"
      />
```

Mais on doit faire attention ici, parce que dans cette XML, il y le fragment de map, et les buttons de mode, alors on va regarder fragment de map comme une partie de cette activity_maps.XML, et ce fragment sérail rangé avec les trois buttons en forme horizontal en utilisant

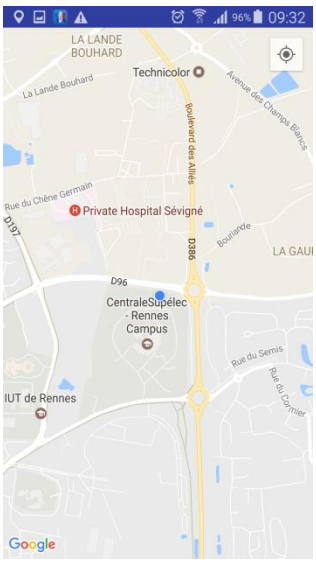
```
<RelativeLayout
......
android:orientation="horizontal"
......>
<fragment
.......(pour map)
/>
<Button
android:id="@+id/bicycling"
......
/>
<Button
android:id="@+id/walking"
android:layout_marginLeft="17dp"
android:layout_marginStart="17dp"
android:layout_toEndOf="@+id/bicycling"
android:layout_toRightOf="@+id/driving"
......
/>
<Button
android:id="@+id/driving"
android:layout_marginLeft="34dp"
android:layout_marginStart="34dp"
android:layout_toEndOf="@+id/walking"
android:layout_toRightOf="@+id/walking"
......
/>
</RelativeLayout>
```

Fonction de localisation(l'origine)

J'ai réalisé une autre fonction dans autre programme, ce que j'essaie de réaliser est de combiner les deux programmes
Dans ce programme, j'ai utilise

" `mGoogleMap.setMyLocationEnabled(true);`
"

Alors comme j'ai dit dans la partie premièrement, je peux pas obtenir latitude ni longitude, ce que j'ai obtenir est la photo suivante



Alors dans cette programme, j'ai ajouté un EditText pour que utilisateur importe l'endroit qu'il veut aller, les codes dans XML sont suivants:

```
<LinearLayout
.....
>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:id="@+id/editText"
    android:layout_gravity="center_horizontal"
    android:text="where do you go"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Go"
    android:id="@+id/button"
    android:onClick="geoLocate"
    android:layout_gravity="center_horizontal"/>
<fragment
……..
/>
</LinearLayout>
```

Et le bouton est de pour exciter un événement de click, quand on click le button “Go”, l’adresse qu’on écrit vsérail obtenu en utilisant:

```
String location = et.getText().toString();  //get the writing from input
```

Mais il est nécessaire de transformer le type de “String” en autre forme pour obtenir autre le type de “double” de latitude et longitude:

```
Geocoder gc = new Geocoder(this);
List<Address> list = gc.getFromLocationName(location, 1);
Address address = list.get(0);
String locality = address.getLocality();
```

Utilisant les codes suivants pour obtenir latitude et longitude, et “gotoLocationZoom()” pour bouger la caméra à l’endroit et aussi grandir le map pour préciser l’endroit.

```
double lat = address.getLatitude();
double lng = address.getLongitude();
gotoLocationZoom(lat, lng, 15);
```

Et on dois ajouter le marker sur l’endroit qu’on importe:

```
setMarker(locality, lat, lng);
```

Les deux fonction (obtenir l’endroit importée et changer la forme de ce que on obtenir dans le text) sont réalisées dans la fonction de :

```
public void geoLocate(View view) throws IOException {
……
}
```

Parpe qu’on définir le button dans l’activité de XML, on a utiliser

```
android:onClick="geoLocate"
```

Ce que permet est d’écrire la fonction de click directement, n’est pas nécessaire d’écrire la fonction de “OnClickListener
”

Alors on importe “Rennes”, on peut obtenir la photo suivante:

Et après ce que je vais réaliser est de combiner les deux programmes, l'endroit original est ma current position, la destination est l'endroit que j'ai importé, pour la fonction de partager la position entre les amis, je vais la réaliser en envoyant mon current position, ça va utiliser la fonction de bavarder.

Réussite de bouton des mode de sortie

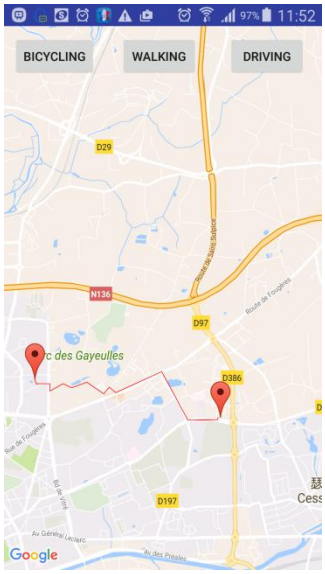
F:\AS_esace\04.05\TestForButtonDeSortir.rar

Dans cette partie, j'ai réalisé à faire les bouton de mode de sortie fonctionner, la chose que j'ai fait plus importante est que j'ai mis tous les codes sur comment dessiner le polyline entre deux points dans la méthode de

```
public void geoLocate(View view)
{
.....
}
```

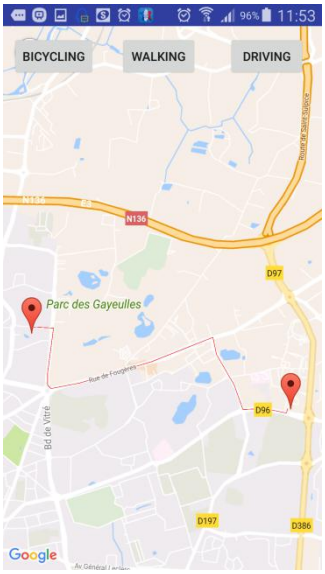
Mais la choses moins bien est que chaque mois quand je change le mode de sortie, les markers seront rechargés, mais le map sera pas stable quand je change le mode de sortie. Les photos suivant sont les résultats du changement de mode

La photo de "bicycling"



La photo de "walking"

La photo de "driving"



Maintenant j'ajouterai la fonction de position, le problème est que j'ai utilisé la fonction de

```
mMap.clear();
```

Chaque fois quand je change le mode de sortir, c'est-à-dire le marker de position sérail disparaître quand je change le mode de sortir, seulement si je l'ajoute chaque fois dans le même temps de changer le mode de sortir.

Combinaison le projet de TestForButtonDeSortir et le projet d'obtention mon position actuelle

F:\AS_esace\04.05\MyApplication.rar

Dans cette projet, j'ai combiné le projet de TestForButtonDeSortir avec le projet d'obtenir la position actuelle, et régler la position actuelle comme la position originaire, et dans la fonction de

```
public void geoLocate(View view) {
```

.....

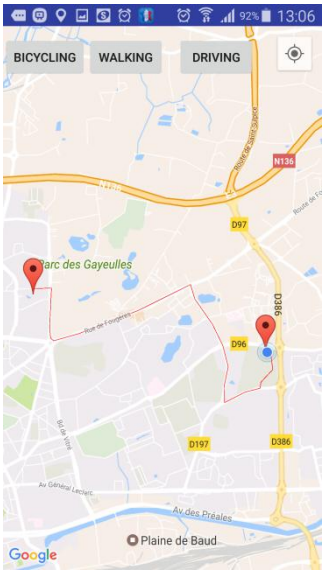
}

Je peux obtenir la fonction finale, c'est-à-dire je change le mode de sortir différent en tapant les buttons différents.

La photo de "bicycling"

La photo de "walking"

La photo de "driving"



Réalisation de la navigation avec la destination écrite

F:\AS_esace\04.05\MyApplication1.rar

Dans cette partie, j'ai réalisé d'obtenir la destination en tapant sur la ligne en haute, par utilisant la phrase

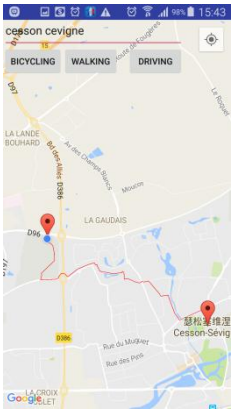
```
EditText et= (EditText) findViewById(R.id.editText);
String location=et.getText().toString();
Geocoder gc=new Geocoder(this);
List<Address>list=gc.getFromLocationName(location,1);
Address address=list.get(0);
String locality=address.getLocality();
Toast.makeText(this, locality, Toast.LENGTH_SHORT).show();
LatLng origin = new LatLng(mLatitudeText, mLongitudeText);
LatLng dest = new LatLng(address.getLatitude(),address.getLongitude());
```

On peut obtenir la destination écrit sur la ligne et la régler comme la position de la destination, les codes procédés sont écrits dans la méthode de

```
public void geoLocate(View view) throws IOException {......}
```

La photo de "bicycling"

La photo de "walking"



La photo de "driving"

Combinaison le mode de sortir avec la fonction de navigation

F:\AS_esace\04.05\MyApplication2.rar
Dans cette partie, j'ai ajouté l'événement de click dans les codes précédents,

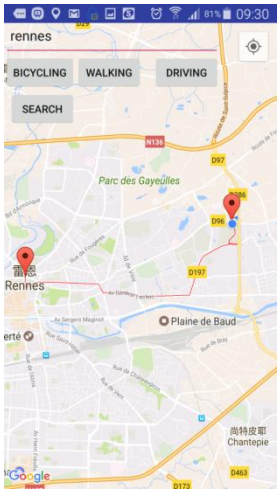
```
public void Search(View view) throws IOException {
    if (mMap!=null){
        mMap.clear();
    }
    Address address = getAddress();

    gotoLocationZoom(address.getLatitude(), address.getLongitude(), 15);

    LatLng dest=new LatLng(address.getLatitude(),address.getLongitude());
    mMap.addMarker(new MarkerOptions().position(dest));
}
```

Avec la fonction:

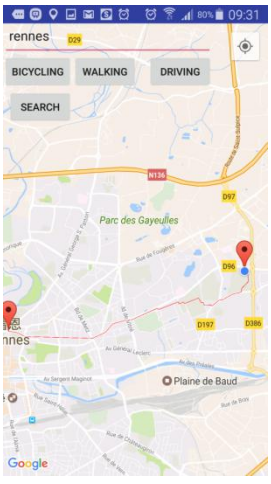
```
@NonNull
private Address getAddress() throws IOException {
    EditText et = (EditText) findViewById(R.id.editText);
    String location = et.getText().toString();

    Geocoder gc = new Geocoder(this);
    List<Address> list = gc.getFromLocationName(location, 1);
    Address address = list.get(0);
    String locality = address.getLocality();

    Toast.makeText(this, locality, Toast.LENGTH_SHORT).show();
    return address;
}
```

Et le résultat:

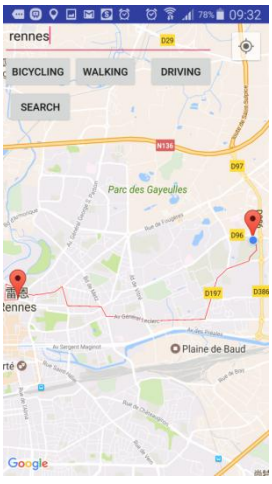
La photo de "bicycling"



La photo de "walking"



La photo de "driving"



La destination

Méthode de cliquer sur la map

Pour obtenir un endroit où on veut aller, on peut juste cliquer sur la map, et les codes sont suivants:

```
mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {

    @Override
    public void onMapClick(LatLng point) {

        if (mMap!=null){
            mMap.clear();
        }

        // clearing map and generating new marker points if user clicks on map more than two times
        if (MarkerPoints.size() > 0) {
            mMap.clear();
            MarkerPoints.clear();
            MarkerPoints = new ArrayList<>();
            ShowDistanceDuration.setText("");
        }

        // Adding new item to the ArrayList
        MarkerPoints.add(point);

        // Creating MarkerOptions
        MarkerOptions options = new MarkerOptions();

              options.position(point);
        if (MarkerPoints.size() == 0) {
            options.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
        } /*else if (MarkerPoints.size() == 2) {
            options.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
        }*/

        // Add new marker to the Google Map Android API V2
        origin = new LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());
        mMap.addMarker(options);
       /* mMap.addMarker(new MarkerOptions().position(origin));*/

        // Checks, whether start and end locations are captured
        if (MarkerPoints.size() >= 1) {

            dest = MarkerPoints.get(0);
        }
```

```
    }
});
```

PlaceAutoComplete

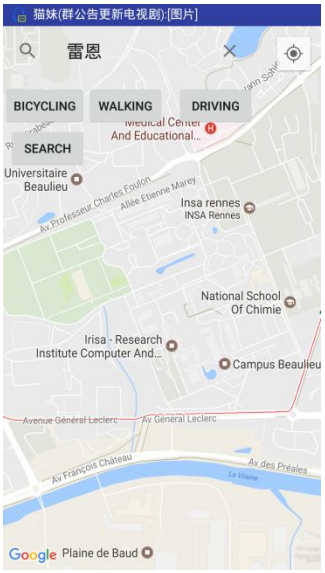
F:\AS_esace\04.05\MyApplication3.rar,
et le test pour cette fonction:F:\AS_esace\10.05\TestForAutoComplete.rar

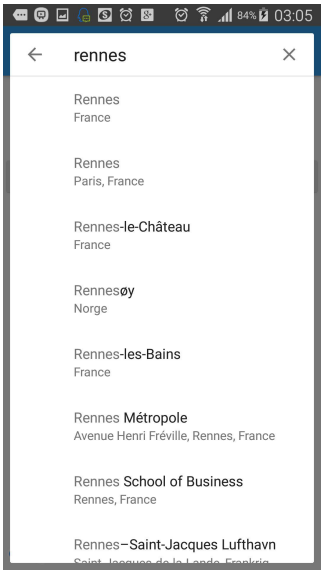
Dans cette partie, j'ai utilisé la fonction de compléter la place automatiquement remplacer la fonction d'emporter la place sans autocomplete, et la fonction de "PlaceAutoComplete" est représenté dans cette article, en ajoutant le fragment dans le code de XML et modifiant la fonction de "onMapReady" comme suivant:

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    mMap.setMyLocationEnabled(true);

    PlaceAutocompleteFragment autocompleteFragment= (PlaceAutocompleteFragment)
getFragmentManager().findFragmentById(R.id.place_autocomplete_fragment);
    autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {
        @Override
        public void onPlaceSelected(Place place) {
            autoPlace=place;
        }

        @Override
        public void onError(Status status) {


        }
    });
}
```

Et dans l'événement de click de "search", j'ai modifié les codes comme suivants:

```
if (mMap!=null){
    mMap.clear();
}

mMap.addMarker(new MarkerOptions().position(autoPlace.getLatLng()));
gotoLocationZoom(autoPlace.getLatLng().latitude,autoPlace.getLatLng().longitude,15);
}
```





Les codes pour placeautocomplete dans le fichier de XML:

```
<fragment
    android:id="@+id/place_autocomplete_fragment"
    android:name="com.google.android.gms.location.places.ui.PlaceAutocompleteFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

La version finale pour la combinaison des fonction

La version d'obtenir la valeur de la position actuel

Pour la raison que j'ai parlé avant, si on veut obtenir la valeur de la position actuele précisée, on doit utiliser GoogleApiClient, et pour la décision de valeur, je le faire dans le méthode de:

`public void onLocationChanged(Location location){...}, cette fonction est effectué quand la position du utilisateur a changé, et on peut obtenir location comme la position de l'utilisateur, et la fréquence de l'effectuer est décidée par la fonction de:`

```
@Override
public void onConnected(@Nullable Bundle bundle) {

    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(10000);
    mLocationRequest.setFastestInterval(10000);
.......
}
```

`On peut changer les deux chiffres pour changer la fréquence de vérifier si la position a changé ou pas.`

Affichage le temps nécessaire à partir de l'origine à la destination

Dans cette partie, j'ai utilisé la méthode de:

```
private void build_retrofit_and_get_response(String type) {.....

String distance = response.body().getRoutes().get(i).getLegs().get(i).getDistance().getText();
String time = response.body().getRoutes().get(i).getLegs().get(i).getDuration().getText();
ShowDistanceDuration.setText("Distance:" + distance + ", Duration:" + time);
.......
.......
}
```

`Pour obtenir la valeur de la distance et le temps nécessaire, et après, on va les afficher sur map,mais pour obtenir le distance et le temps en temps réel, c'est-à-dire quand ;aposition de l'utilisateur a changé, les deux valeurs vont changer aussi, du coup, ce que on va faire est de changer la fréquence de vérifier si la position a changé ou pas(la méthode qu'on a parlé tout à l'heure), et fais attention, sauf ça, on va redéfinir les différent "mode de sortir " dans la méthode de :`

```
public void onLocationChanged(Location location) {
......
```

```
mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
mMap.animateCamera(CameraUpdateFactory.zoomTo(15));


if (Type=="driving"){
    build_retrofit_and_get_response("driving");
}else if (Type=="walking") {
    build_retrofit_and_get_response("walking");
}else if(Type=="bicycling"){
    build_retrofit_and_get_response("bicycling");
}
.....
}
```

Et dans cette méthode, les codes suivants:

```
MarkerOptions markerOptions = new MarkerOptions();
markerOptions.position(latLng);
markerOptions.title("Current Position");
markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
mCurrLocationMarker = mMap.addMarker(markerOptions);
```

Ont assuré quand on change la positon actuel, la ligne rouge qui a connecté l'origine et la destination va changer aussi.