Rapport

14 janvier 2013

Table des matières

Introduction

Afin de valider les comptences acquises concernant la cration d'application sur terminaux mobiles, les tudiants par groupe de quatre ont chacun reu un projet Android.

Notre sujet consiste crer un **GPS piton**. Ce GPS offrira plusieurs possibilits, deux principales seront de retrouver des points d'intrts, que nous appelleront par la suite **POI**, sur une carte ou l'aide de la ralit augmente. Nous avons dcid d'utiliser Google API pour notre application. Ce choix est d principalement au fait que certains membres du groupe ont dj une premiere exprience sur ces API.

Trois grandes ides de travails sont ressorties la rcuprations de POI via le service Google, la ralit augmente en utilisant la camra, et le travail de calcul d'itinraire sur la Map.

La premire partie consiste crr une classe permettant la communication entre notre application et la base de donne Google. La seconde utilise la camra afin d'afficher les POI sur l'cran, et la dernire partie utilise les informations reues pour les afficher de manire classique sur une carte.

Chapitre 1

Rpartition du travail et travail ralis

Nous avions rpartit les taches ds le debut du projet : Kevin souhaitait implmenter la ralit augmente, Christophe et Benoit se sont chargs de comprendre comment nous pouvions interroger un service web pour rcuprer des informations Golocalises. Nicolas avait choisis de se charger de la partie d'affichage de la carte, et de la gestion de l'itinraire tant donn qu'il connaissait dj ces problmes de part un sujet de stage similaire qu'il a ralis par le pass.

Nous aurions voulu prsenter une application plus complte, qui permette a la fois de consulter un rsultat via la ralit augmente ainsi que via la carte. Mais la carte n'a pas t implmente de manire fonctionnelle.

Chapitre 2

Rcupration des informations de lieu

Un des points important du sujet est la rcuprations de points d'intrts. Pour ce faire il nous fallait utiliser des services en ligne, mais quel service ?

Quelle API choisir ?

Dans un premier temps, nous avions opter pour une solution se dissociant de Google, OpenStreetMap. Aprs quelques recherches infructueuses de librairies ou d'explications claires, l'quipe dcida de migrer vers une API dj connu de certains membres, **Google API**.

OpenStreetMap est assez peu document sur le web : la page Wikipedia ne permet pas de savoir comment peuvent etre interroges les donnes d'une carte. Nous avons compris que les donnes sont enregistres en XML, avec un format spcifique, mais cette structure n'est pas dfinie clairement. Cela signifie que si nous voulions charger des rsultats rcuprs depuis OpenStreetMap, il nous aurait fallu parser nous mme le Xml et implmenter des classes contenant les rsultats. Par ailleurs, il n'existe que trs peu de tutoriaux dtailles permettant d'implmenter des fonctionnalits grce a OpenStreetMap. Etant donn que cela reprsentait une quantit de travail importante sans pour autant nous offrir plus de possibilits pour l'application, nous avons choisis

aprs quelques semaines d'utiliser les librairies Google. Notre envie de dcouvrir OpenStreetMap nous a au final fais perdre du temps que nous aurions pu passer implmenter l'application. En l'occurence, l'API Google Place dfinit des dizaines de types de batiments, et il existe des services web que l'on peut interroger via des requetes Web, avec les paramtres de recherche dans l'URL.

Google Place

Parmi les nombreuses fonctionnalits que propose le gant de l'internet, une en particulire a retenu notre attention, une librairie permettant de rcuprer les informations des lieux, Google Place API. Cette dernier nous permet de consulter, comme sur les applications maps du mme groupe, les caractristiques des lieux enregistrs dans leur base de donnes.

Comment cela fonctionne-t-il ?

Le principe de l'API de d'envoyer des requtes http au serveur afin de recevoir les points d'intrts demands.

les diffrents requtes

Nous utilisons quatres types de requtes pour ce service.

Les recherches de proximit

La recherche de proximit permet la rcupration de tous points d'intrts (cinma,banque ...) dans un rayon maximal de 50km autour de la position de l'utilisateur.

Les recherches par texte

Cette partie permet a l'utilisateur de recherche un lieu a partir du texte entre. Par exemple, s'il saisit **Pizza Orlans**, il retournera les lieux en rapport avec pizza et Orlans.

La demande de dtails

Cette requte est trs utile car par dfaut, nous rcuprons un lieu vraiment basique (nom, adresse, etc). Elle nous permet de complter le lieu avec des informations complmentaires qui vont du site internet jusqu'aux commentaires ajout par les utilisateurs de Google en passant par les horaires d'ouverture. Dans notre cas nous utilisons que des informations simple comme le site web.

La rcupration de photo

Cette derniere permet de rcuprer via une rfrence, une url poitant vers la photo d'un lieu.

Les rsultats obtenus

L'usage de cette librairie nous renvoi les donnes au format JSON. Afin de les rendre utilisable, le rsultat est directement transformer en classe par le biais d'un parseur. Celui-ci est crer lors de la gnration du transporteur HTTP dans la classe FouilleDonnee.

Le code

Comme tous service de Google l'exige, il a fallu enregistrer le projet afin d'obtenir une cl nous autorisant se connecter aux serveurs. La norme Android spcifie d'utiliser cette classe dans une classe asynchrone.

Les requtes

Voici une des requtes concernant la rcupration de point d'intrts

```
public ListeLieu getLieuProximiteParType(double lat,double lng,
    ArrayList<String> types,int distance) {
  try {

    HttpRequestFactory httpRequestFactory = createRequestFactory(
        HTTP_TRANSPORT);
    HttpRequest request = httpRequestFactory
        .buildGetRequest(new GenericUrl(PLACES_SEARCH_URL));


    request.getUrl().put("location", lat + "," + lng);
    request.getUrl().put("radius", distance); // in meters

    if(types != null && types.size()>0) {
      types=FrancaisToApi(types);
      request.getUrl().put("types", typesFormatUrl(types));
    }

    completePlaceQuery(request.getUrl());
    Log.d("url",request.getUrl().toString());
    ListeLieu list = request.execute().parseAs(ListeLieu.class);
    return list;

  } catch (HttpResponseException e) {
          Log.e("Error:", e.getMessage());
      } catch (IOException e) {
    e.printStackTrace();
  //} catch (InterruptedException e) {
    //e.printStackTrace();
      }
  return null;
}
```

Lieu

Sur chaque membre de la classe nous ajoutons l'annotation @Key afin que le parseur remplisse chaque champs de la classe avec le code JSON. Afin de filtrer les informations dont nous n'avons pas besoin, comme les commentaires d'utilisateurs de Google, il suffit de ne pas mettre de champs correspondant.

```java
//Il faut que cette classe soit serializable pour appliquer le
    writeObject() dessus
public class Lieu implements Serializable{

  private static final long serialVersionUID = 1L;

  public Lieu(){
  }
  /**
   * L utilisateur veut enregistrer sa position actuelle comme
       favoris
   * Il saisis certains champs seulement, et l'id n'est pas connu
   * @param name
   * @param reference
   * @param icon
   * @param vicinity
   * @param geometry
   * @param formatted_address
   */
  public Lieu(String name, List<String> types , String reference,
      String icon, String vicinity, Geometry geometry, String
      formatted_address,String phoneNumer,String website){
    id="";
    this.types = types;
    this.name=name;
    this.reference=reference;
    this.icon=icon;
    this.vicinity=vicinity;
    this.geometry=geometry;
    this.formatted_address=formatted_address;
    this.formatted_phone_number = phoneNumer;
    this.website = website;
  }

  @Key
  public String id;
  @Key
  public String name;
  @Key
  public String reference;
  @Key
  public String icon;
  @Key
  public String vicinity;
  @Key
  public Geometry geometry;
  @Key
  public String formatted_address;
```

```
45    @Key
      public List<Photo> photos;
      @Key
      public List<String> types;
      /** DETAILS */
50    @Key
      public String formatted_phone_number;
      @Key
      public String website;
      @Override
55    public String toString() {
        return id+" "+name+" "+icon;
      }

      public static class Geometry implements Serializable
60    {
        private static final long serialVersionUID = -1846546423355113268
            L;
        @Key
        public MyLocation location;

65      public Geometry(){
        }

        public Geometry(MyLocation location){
          this.location=location;
70      }
      }

      public static class MyLocation implements Serializable
      {
75      private static final long serialVersionUID = -745398283024148157L
            ;

        @Key
        public double lat;
        @Key
80      public double lng;

        public MyLocation(){
        }

85      public MyLocation(double lat, double lng){
          this.lat=lat;
          this.lng=lng;
        }
      }
90
      public double getLatitude() {
```

```
        return geometry.location.lat;
    }

    public double getLongitude() {
        return geometry.location.lng;
    }

    @Override
    public boolean equals(Object o) {
        Lieu l = (Lieu)o;
        return this.name==l.getNom();
    }
}
```

ListeLieu

Cette classe correspondant la racine du JSON, le status nous permet de savoir comment la requte s'est droule, **next_page_token** permet d'avoir acccs au 20 rsultats suivants dans une limite de 60 par requtes et la liste de Lieu contient tous les lieux que la requte a demande.

```
public class ListeLieu implements Serializable {

    private static final long serialVersionUID = -1467727864221797449
        L;

    @Key
    public String status;

    @Key
    public String next_page_token;

    @Key
    public List<Lieu> results;

}
```

PlaceDetails

Cette classe, comme ListeLieu correspondant a la racine du retour de notre requte concernant la demande de dtails sur un lieu, elle contient un lieu et le status de la requte.

Ces classes forment la liaison entre notre application et les serveurs de Google. Mais existe-t-il d'autres hbergeurs pour ce type de services ?

Et pourquoi pas quitter Google ?

Les plateformes Google tant gratuites et mondialement connues, ils doivent donc limiter les usages de leurs services, alors autres que rcuprer toutes vos informations, ce dernier bride le nombre de requtes pour chaque services, dans notre cas nous atteignons la barre de 1000 requtes par jour, ridicule.

Il nous est possible de dpasser cette limitation mais tout en restant gratuit par l'enregistrement d'une carte bancaire, impensable pour nous. De ce faite, vous pensions nouveau migrer vers un autre hbergeur pour ce service.

En effet, il existe *FourSquare* ou *Yahoo GeoPlanet*, qui propose les mmes fonctionnalits. Si l'envie en prenait aux dveloppeurs de migrer vers ces plateformes, il ne suffirait que de modifier les fonctions associs aux requtes afin de se dsenchaner de ce gant.

Chapitre 3

Gestion de la carte

3.1 La carte

On utilise la carte de l'api google maps. L'ativit qui gre la carte devra utiliser la classe MapActivity au lieu d'Activity.

L'utilisateur a besoin de pouvoir zoommer sur la carte, pour cela on utilise la fonction mapView.setBuiltInZoomControls(true) qui nous permet le zoom et aussi cette fonction permet au tlphone qui n'on pas de multitouch de pour zoomer grce au boutonnqui s'affiche suite au clic sur la carte.

Ensuite nous avons besoin d'un controleur qui va nous permettre de faon interne de pourvoir effectuer un zoom ou de placer la carte sur un point prcis. Par exemple au premier affichage de la carte, on va centrer la carte sur notre possition.

L'objet MyLocationOverlay va nous permetre de connaitre la position de l'utilisateur et aussi d'ajouter un overlay la carte qui sera la position de l'utilisateur.

3.2 Gestion des POI

Un point d'intret sera caractris par un OverlayItem qui sera afficher sur la carte. pour stocker nos points d'intrets, on a cr la classe ListItemizedOverlay qui contient une liste de tout les POI. Cette classe contient un Drawable qui sera l'icone associ notre point.

L'utilisateur pour cliquer sur chacun de ces points. ainsi il accedera aux informations de ce point, donc l'adresse, numro de tlphone, ... Ensuite depuis cette fentre il pourra dcider de crer un itinraire vers ce point en mode statique, il y aura juste le trac de l'itinraire sur la carte. Sinon l'utilisateur pourra choisir d'utiliser la navigation.

Chapitre 4

Ralit Augmente

Vue Personnalise

Pour pouvoir afficher les icnes sur la camra, il fallait dans un premier temps pourvoir grer l'ouverture et la fermeture de celle-i. Pour cela nous avons d avoir recours une vue personnalise. La vue personnalise rcupre les paramtres de la camra du mobile, elle cre une surface que l'on va pourvoir utiliser dans nos layouts en mettant le chemin de notre classe.

```
<com.example.eyeway.realiteAugmente.CustomCameraView
        android:id="@+id/camera"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
```

Positionner les points d'intrt

Le second travail est l'un des plus important de la partie ralit augmente, il s'agit du positionement des points d'intrt sur la camra. Nous devons pour cela rcuprer la position actuelle de l'utilisateur et la position du point d'intrt. Nous faisons ensuite un calcul d'angle en rcuprant l'angle entre nous et le nord ainsi que celui entre nous et le point d'intrt. Ceci va nous permettre de savoir en fonction de l'orientation du tlphone , les points d'intrt se trouvant dans la direction regarde par l'utilisateur.

Pour rcuprer notre position actuelle nous avons accs deux ressources, le GPS et le rseau du tlphone. Chacune de ces ressources comportent ses inconvnients et ses avantages. Le GPS a l'avantage d'tre prcis mais il a pour inconvnient d'tre lent se mettre en route dans certains cas d'utilisations, par exemple dans un btiment. Le rseau lui est pltot rapide mais il comporte une marge d'erreur vraiment importante.

Nous avertissons donc l'utilisateur lorsque le GPS n'est pas activ, en lui indiquant que sans l'utilisation de celui-i l'application sera moins prcise. Nous avons donc absolument besoin de l'un ou de l'autre, c'est pourquoi nous bloquons l'utilisation de la gocalisation si l'itinrance de donnes n'est pas active ou au moins une des deux ressources n'est pas active.

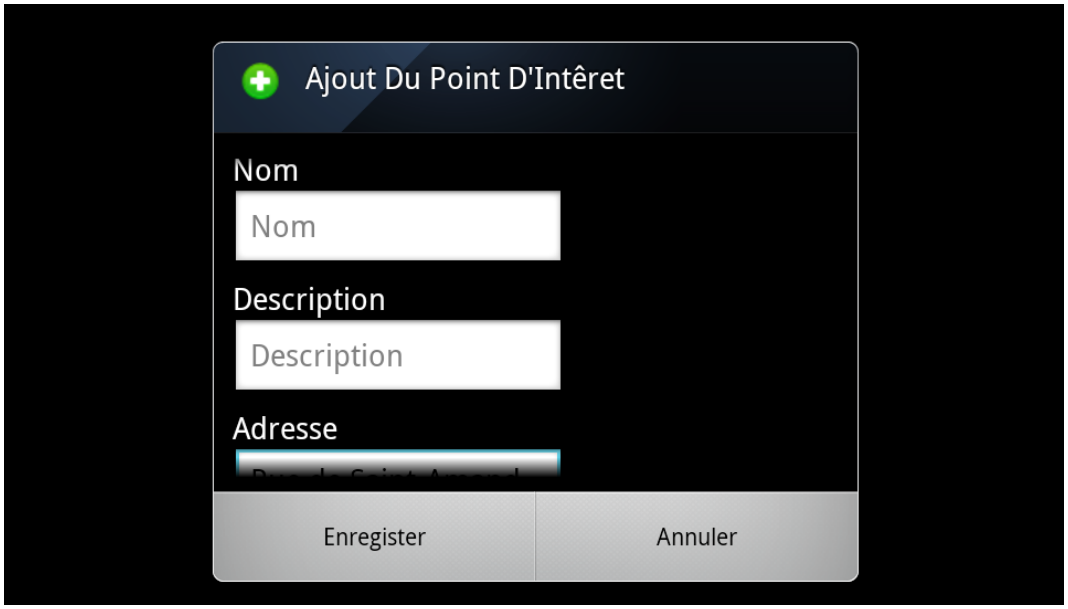
L'tape suivante pour le positionement d'un point d'intrt tait de savoir en fonction de l'angle o le postionner sur l'cran. Pour cela nous avons utilis l'attribut margin pour placer l'imageView sur l'cran, nous rcuprons ainsi la hauteur, la largeur de l'cran, les valeurs renvoyes par l'acclrometre du tlphone et nous utilisons un calcul trouv sur internet qui permet partir de ces valeurs de connatre la position du point d'intrt sur l'cran.

Chaque point d'intrt aura une icne en relation avec son type de batments et un label qui correspondra la distance entre la position actuelle de l'utilisateur. Nous devons donc pour chaque point d'intrt recalculer le label lorsque la position de l'utilisateur change. Nous avons mis une icne par dfaut si le type de batment n'est pas connu.

Cration d'un nouveau point d'intrt

L'utilisateur peut tout moment vouloir crer un nouveau point d'intrt s'il le souhaite. Pour cela il devra appuy quelques secondes sur l'cran, nous offrons la possibilit l'utilisateur de renseigner certains champs : le nom, une description, l'adresse, le numro de tlphone et le site web de son nouveau point.

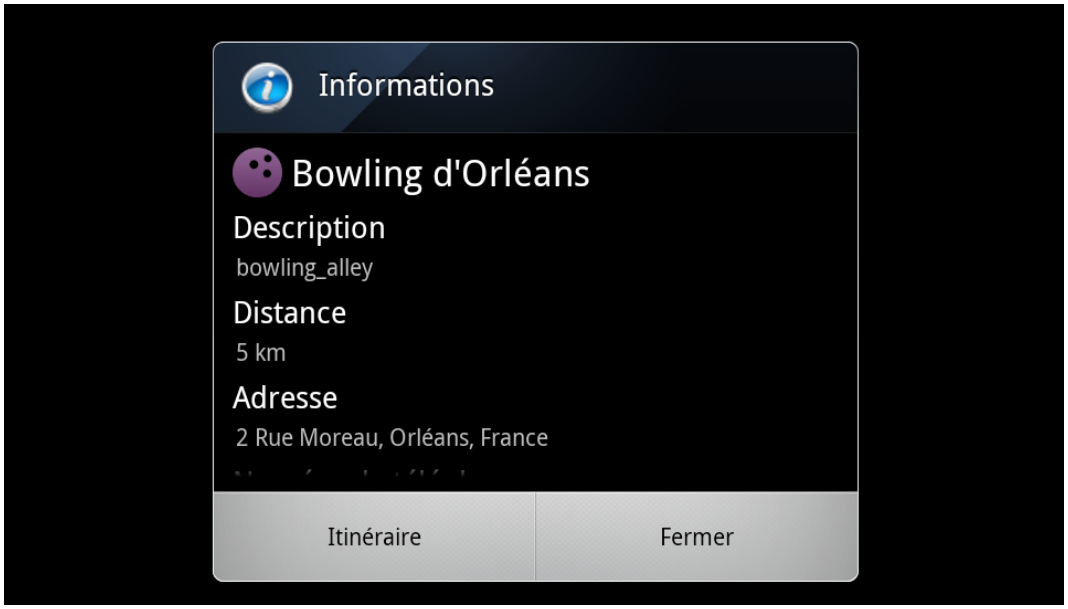
Nous pr-remplissons le champ adresse l'aide de la classe GeoCoder qui permet de retourner en fonction d'une longitude et d'une lattitude l'adresse correspondante. L'utilisateur doit ensuite appuy sur le bouton sauvegarder, ceci aura comme action de sauvegarder le nouveau point comme favoris et de l'ajouter directement l'cran. L'utilisateur pourra ensuite retrouver son nouveau point chaque utilisation de l'application dans le menu de gestion de favoris.



Dtails d'un point d'intret

Pour avoir accs au dtails d'un point d'intrt, nous avons fait en sorte que les images soit cliquables. Donc lorsque l'utilisateur voudra le dtails d'un certain point il devra juste cliquer sur l'icne en question. Lorsque le clic est effectu nous utilisons la requte de dtails qui va rcuprer le lieu complt avec les informations manquantes.

Une fois la requte execute nous affichons une boite de dialogue comportant tout les champs renseigns dans lieu. Et nous offrons aussi la possibilit l'utilisateur de sauvegarder le lieu en question et de basculer vers la vue map pour calculer un itinraire. La partie Map n'tant pas fonctionnelle nous avons pas eu la possibilit de l'utiliser.



Amlioration possible

Pour la ralit augmente nous aurions aim pouvoir implmenter un systme de guidage vers un point d'intrt slectionn par l'utilisateur. Pour cela nous avions penser utiliser une flche qui dirigerais l'utilisateur vers sa destination.

Chapitre 5

Interface

Implmentation des layouts

Les layouts sont la mise en forme des diffrentes pages de l'application. Android propose des attributs prdfinis pour paramtrer l'affichage des composants (chaque composant Android est une View). Nous avons t plus loin que les attributs prdfinis en Android : Christophe souhaitait par exemple avoir des lments avec des bords arrondis, des background en couleur dgrades, etc. Pour cela, il a fallu utiliser les drawable, a dfinir dans une fichier Xml, puis a appliquer sur les View concernes :

```
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <gradient
    android:startColor="@color/border1_start"
    android:endColor="@color/border1_end"
    android:gradientRadius="3"
    android:type="sweep"/>
</shape>
```

Drawable dfinissant les bords arrondis

Les diffrentes listes visibles dans l'application (par exemple la liste des fonctionnalits sur le menu principal) sont gres par un fichier Xml qui dfini la liste en elle mme, et un fichier Xml qui dfinit un lment de cette liste. La encore il s'agit d'tendre ce qui est offert de base par android : les items d'une listview par dfaut ne peuvent

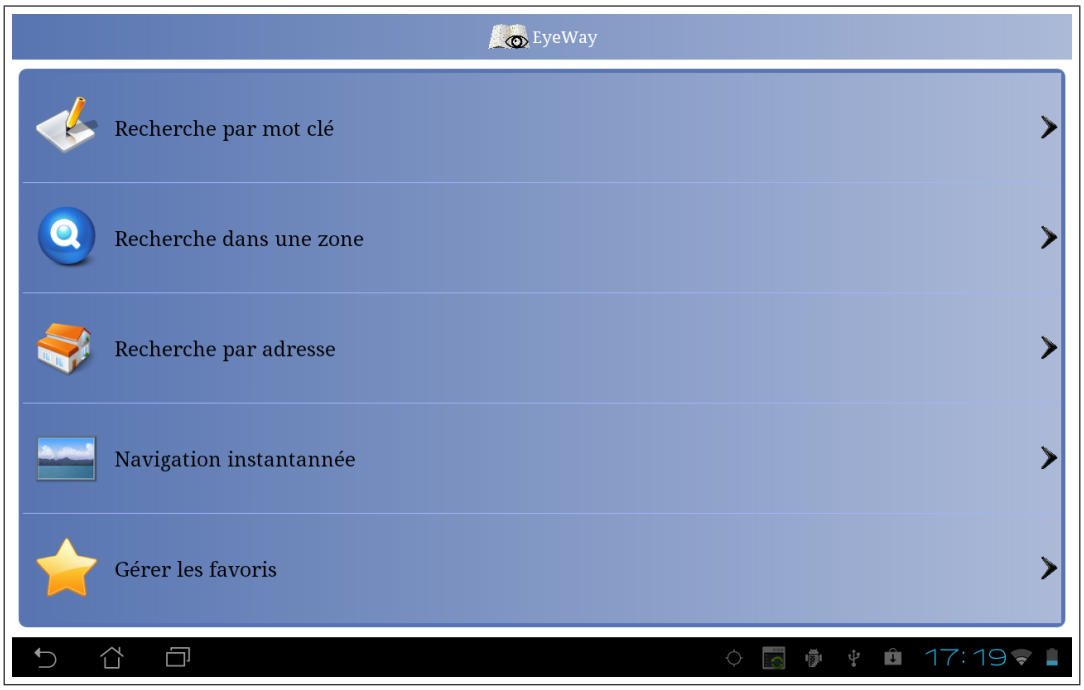
contenir qu'une String, alors que dans notre cas nous voulons afficher une icone et une String. Cela est rendu possible par l'heritage de la classe Adapter. Dans le cas d'une listview que l'on veut mettre a jour au cours de la vie de l'application, il faut implmenter la mthode notifyDataSetChanged() et l'appeler explicitement afin de rafraichir la ListView.

```
public class ListAdapter extends ArrayAdapter<Fonctionnalite> {

    Context context;
    int layoutResourceId;
    ArrayList<Fonctionnalite> data= null;

    public ListAdapter(Context context, int layoutResourceId, ArrayList<Fonctionnalite> data) {
        super(context, layoutResourceId, data);
        this.layoutResourceId = layoutResourceId;
        this.context = context;
        this.data = data;
    }
    public void add(Fonctionnalite f){
        if(!data.contains(f)){
            data.add(f);
            notifyDataSetChanged();
        }
    }
    public void remove(int index){
        data.remove(index);
        notifyDataSetChanged();
    }
```

Adapter permettant la cration d'un listview personnalise

Lorsque nous avons test l'application sur une tablette, nous nous sommes apercus que l'affichage du menu tait trop petit sur une tablette, et lorsqu'il tait correct sur tablette, les lments taient trop grand sur telephone. Nous avons grer ces deux cas en dtectant le priphrique sur lequel est lanc l'application et en appliquant un layout diffrent.


Orange F
13:52
EyeWay
Recherche par mot clé
Recherche dans une zone
Recherche par adresse
Navigation instantannée
Gérer les favoris


EyeWay
Recherche par mot clé
Recherche dans une zone
Recherche par adresse
Navigation instantannée
Gérer les favoris
17:19

Nous avons crer un bandeau contenant le nom et l'icne de l'application visible sur chaque page. Ce bandeau consiste en un fichier xml, charg par les diffrents layout grce la directive include.

Gestion du swipe

Nous avons implment le swipe sur diffrents formulaires afin de revenir sur le menu principal.

Gestion de l'orientation

Lorsque l'orientation du terminal change, android fait un appel au onCreate() de l'application, ce qui a pour effet de perdre les saisies de l'utilisateur. C'est le cas dans la recherche par perimtre avec la liste des types extensible. Nous avons gr le changement d'orientation en restaurant les types saisis par le pass.

AlertDialog customise

De base, Android ne permet de crer que des AlertDialog contenant un texte, et pour afficher un favoris ou bien un rsultat de la recherche, nous avions besoin d'afficher le nom du lieu, l'adresse, la ville, le site web... Nous avons utilis des AlertDialog customiss, qui contiennent un layout dfinit la encore par un fichier Xml.

```java
public void afficherAlertDialogDetailsPoi(final Lieu l,final int pos){
    LayoutInflater factory = LayoutInflater.from(this);
    final View alertDialogView = factory.inflate(R.layout.alertdialog_details_poi,null);
    AlertDialog.Builder adb = new AlertDialog.Builder(this);
    adb.setView(alertDialogView);
    TextView t = (TextView) alertDialogView.findViewById(R.id.nom_poi);
    t.setText(l.getNom());
    t = (TextView) alertDialogView.findViewById(R.id.description_poi);
    t.setText(l.getVicinity());
    t = (TextView) alertDialogView.findViewById(R.id.adresse_poi);
    /* du code ... */
    adb.setPositiveButton("Afficher ", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            showBoiteChoix(l);
        }
    });
    adb.setNegativeButton("Supprimer", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            afficherAlertDialogSuppression(l,pos);
            return ;
        }
    });
```

AlertDialog permettant de voir un favoris enregistr

On a aussi construit une classe AlertDialogManager qui permet de contenir la dfinition des AlertDialog personnaliss de l'application, de manire a factoriser le code et a le rendre simple a utiliser.

Choix de l'affichage

Pour chaque formulaire il est possible de consulter les rsultats via la map ou la ralit augmente par le biais d'une boite de dialogue.

Affichage et sauvegarde des favoris

Nous avons gr la sauvegarde d'un point d'intret, de manire a ce que l'utilisateur puisse conserver les lieux qu'il a consult. Pour cela, l'utilisateur peut rester appuy sur l'ecran de ralit augmente ou bien lorsqu'il consulte le dtail d'un rsultat.

L'affichage est confi a une listview extensible et rafraichie automatiquement (comme expliqu prcdemment pour la listview des types pour la recherche par perimtre).

La sauvegarde des favoris est faite avec des enregistrements dans l'internal storage Android. Pour enregistrer un Lieu, il faut que cette classe soit serializable, puis il faut convertir l'objet concern en tableau de bytes, et appeler la mthode write() :

```
String nomLieu=lieu.getNom();
FileOutputStream fos = fileContext.openFileOutput(nomLieu, Context.MODE_PRIVATE);
byte[] b=null;
b=getBytes(l);
fos.write(b);
fos.close();
```

Nous conservons dans l'application les favoris enregistrs, et il est possible de supprimer un favoris, ce qui va supprimer le fichier concern dans l'internal storage.