

Projet ISN 2013



Evan Guélard



AutoScol'Info

Sommaire:

I- Introduction et présentation du projet

II- Cheminement pour arriver au résultat final

III- Résultat final

IV- Sources

I- Introduction et présentation du projet

- Une application et un site internet liés
- Utile pour la conduite accompagnée
- Un projet concret

II- Cheminement pour arriver au résultat final

Algorithme de base – Projet ISN

Début

Afficher le bouton Départ

Si le bouton Départ est enclenché,

 // Prendre les données GPS au point de départ 1

 Données LatDep du GPS : Latitude

 Données LonDep du GPS : Longitude

 Effacer le bouton Départ et Afficher le bouton Arrivée

FinSi

Si le bouton Arrivée est enclenché,

 // Prendre les données GPS au point d'arrivée 2

 Données LatArr du GPS : Latitude

 Données LongArr du GPS : Longitude

 Effacer le bouton Arrivée

FinSi

// Convertir les données GPS en Radians

Calculer LatDep = LatDep * $\pi/180$

Calculer LonDep = LonDep * $\pi/180$

Calculer LatArr = LatArr * $\pi/180$

Calculer LonArr = LonArr * $\pi/180$

// Calculer la distance d entre les 2 points

Calculer $d = 6371 * \cos^{-1}[\sin(\text{LatDep}) * \sin(\text{LatArr}) + \cos(\text{LatDep}) * \cos(\text{LatArr}) * \cos(\text{LonDep} - \text{LonArr})]$

Dire d // d est donné en km

Fin

II- Cheminement pour arriver au résultat final


Logiciel Eclipse

Application avec GPS

Problème: mon ordinateur n'as pas de GPS

Donc l'émulateur ne fonctionne pas complètement

Bienvenue dans l'application Auto Scol Info



Départ Arrivée

II- Cheminement pour arriver au résultat final

Légère démotivation due au GPS et l'impossibilité de vraiment me rendre compte de mon travail

Création d'un nouveau projet ->

```
package com.example.auto_scol_info;

import android.os.Bundle;

public class MainActivity extends Activity {

    Button btCalcul;
    EditText kmDepart;
    EditText kmArrive;
    EditText kmDistance;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        kmDepart = (EditText) findViewById(R.id.kmDepart);
        kmArrive = (EditText) findViewById(R.id.kmArrive);
        kmDistance = (EditText) findViewById(R.id.kmDistance);
        btCalcul = (Button) findViewById(R.id.btCalcul);

        //on applique un écouteur d'événement au clique sur le bouton
        btCalcul.setOnClickListener(
            new OnClickListener() {
                @Override
                public void onClick(View v) {
                    //on récupère le texte écrit dans l'EditText
                    Integer dep = Integer.parseInt(kmDepart.getText().toString());
                    Integer arr = Integer.parseInt(kmArrive.getText().toString());
                    Integer result = arr-dep;

                    ((TextView)findViewById(R.id.kmDistance)).setText("Distance parcourue: " + result.toString() + " km.");
                }
            }
        );
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

III- Résultat final et améliorations possibles

L'application fonctionne, cependant ils y a des imprécisions:

- Imprécision due au GPS
- Imprécision due à la programmation

En montagne, l'altitude compte cependant mon application de la prend pas en compte

III- Résultat final et améliorations possibles

```
package com.example.autoscolinfo;

import com.example.autoscolinfo.Constantes;

public class MainActivity extends Activity {
    //<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Déclaration des boutons
        Button btnDepart = (Button)findViewById(R.id.btnDepart);
        Button btnArrive = (Button)findViewById(R.id.btnArrive);

        // Déclaration des textes
        EditText distance = (EditText) findViewById(R.id.distance);
        EditText fournisseur = (EditText) findViewById(R.id.fournisseur);
        EditText coordonnees = (EditText) findViewById(R.id.coordonnees);

        // sur clique du bouton départ
        btnDepart.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v) {

                float[] results = null;
                LocationListener listener;
                double distance = 0;
                double distancetotale = 0;

                // Déclaration du service système pour le GPS
                LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

                // Déclaration des critères pour sélection le meilleur fournisseur de services de localisation
                Criteria critere = new Criteria();

                // Pour indiquer la précision voulue
                // On peut mettre ACCURACY_FINE pour une haute précision ou ACCURACY_COARSE pour une moins bonne précision
                critere.setAccuracy(Criteria.ACCURACY_FINE);

                // Est-ce que le fournisseur doit être capable de donner une altitude ?
                critere.setAltitudeRequired(true);

                // Est-ce que le fournisseur doit être capable de donner une direction ?
                critere.setBearingRequired(false);

                // Est-ce que le fournisseur peut être payant ?
                critere.setCostAllowed(false);

                // Pour indiquer la consommation d'énergie demandée
                // Criteria.POWER_HIGH pour une haute consommation, Criteria.POWER_MEDIUM pour une consommation moyenne et Criteria.POWER_LOW pour une basse consommation
                critere.setPowerRequirement(Criteria.POWER_LOW);

                // Est-ce que le fournisseur doit être capable de donner une vitesse ?
                critere.setSpeedRequired(true);

                // Récupération du meilleur fournisseur suivant les critères ci-dessus
                String fournisseur = locationManager.getBestProvider(critere, true);
            }
        });
    }
}
```


III- Résultat final et améliorations possibles

```
// Si le fournisseur n'est pas trouvé, alors on prend le fournisseur GPS par défaut
if (fournisseur == null) {
    fournisseur=LocationManager.GPS_PROVIDER;
    ((TextView)findViewById(R.id.fournisseur)).setText("L'application AutoScolInfo est connectée sur le fournisseur: " + locationManager.GPS_PROVIDER + ".");
}

// Demande de localisation toutes les 10000ms
locationManager.requestLocationUpdates(fournisseur, 10000, 0, new LocationListener() {

    // Déclaration des variable de coordonnées
    double startLatitude=0;
    double startLongitude=0;
    double endLatitude=0;
    double endLongitude=0;
    double distance=0;

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
        String newStatus = "";
        switch (status) {
            case LocationProvider.OUT_OF_SERVICE:
                newStatus = "hors service";
                break;
            case LocationProvider.TEMPORARILY_UNAVAILABLE:
                newStatus = "temporairement indisponible";
                break;
            case LocationProvider.AVAILABLE:
                newStatus = "disponible";
                break;
        }
        ((TextView)findViewById(R.id.fournisseur)).setText("Le fournisseur " + provider + " est " + newStatus + ".");
    }

    @Override
    public void onProviderEnabled(String provider) {
        ((TextView)findViewById(R.id.fournisseur)).setText("L'application AutoScolInfo est connectée sur le fournisseur: " + provider + ".");
    }

    @Override
    public void onProviderDisabled(String provider) {
        ((TextView)findViewById(R.id.fournisseur)).setText("L'application AutoScolInfo est déconnectée du fournisseur: " + provider + ".");
        //MainActivity.distanceBetween (startLatitude, startLongitude, endLatitude, endLongitude, results);
    }

    @Override
    public void onLocationChanged(Location location) {
        ((TextView)findViewById(R.id.coordonnees)).setText("Latitude " + location.getLatitude() + " et longitude " + location.getLongitude());
        if (startLatitude==0 && startLongitude==0) {
            // Initialisation des premières coordonnées
            startLatitude=location.getLatitude();
            startLongitude=location.getLongitude();
            ((TextView)findViewById(R.id.distance)).setText("Début de l'appli");
        } else {
            // Récupération des dernières coordonnées et mise à jour de la distance parcourue
            endLatitude=location.getLatitude();
            endLongitude=location.getLongitude();

            distance = distance + calculDistance(startLatitude,startLongitude,endLatitude,endLongitude);
            startLatitude=location.getLatitude();
        }
    }
});
```

III- Résultat final et améliorations possibles

```
        startLongitude=location.getLongitude();
        ((TextView)findViewById(R.id.distance)).setText("Distance parcourue: "+distance+" mètres.");
    }
}
});

}
});
}

protected static double calculDistance(double startLatitude,
    double startLongitude, double endLatitude, double endLongitude) {
    // TODO Auto-generated method stub
    double distance=0;
    double x1,x2;
    double y1,y2;

    // Conversion et simplification des coordonnées en un plan 2D
    // La faible différence entre les deux points justifie de ne pas tenir compte de la rondeur de la terre
    x1=startLatitude*Constantes.perimetreTerrestreMoyen/360;
    x2=endLatitude*Constantes.perimetreTerrestreMoyen/360;
    y1=startLongitude*Constantes.perimetreTerrestreMoyen/360;
    y2=endLongitude*Constantes.perimetreTerrestreMoyen/360;
    distance=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));

    return distance;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

III- Résultat final et améliorations possibles

Décorer l'application

Augmenter la précision

Finaliser la liaison entre le site et l'application

Utiliser le bouton « Arrivée »

Et surement d'autres finitions

IV- Sources

<http://www.siteduzero.com/informatique/tutoriels/votre-premiere-application-android>

<http://android-pour-les-nuls.fr/tutoriaux/developpement/tuto-developpement-application-android-debutant>

<http://android.developpez.com/cours/>

<https://maps.google.fr/>

<http://www.siteduzero.com/forum/sujet/calcul-d-une-distance-95555>

<http://japan-party.net/infos-pratiques/>