Chapter 1

Rapport TP2 Mobile capteur

1.1 Introduction

Le projet à été fait sous Intelij en utilisant mon téléphone pour exécuter l'application. Mon téléphone a une résolution d'écran de $2310 \times 1080 pixels$.

1.2 Exercice 1

Le premier exercice cherche tous les capteurs du téléphone et les affiches sur une ListView avec leur

- Nom du capteur
- Le type de capteur
- La version du capteur
- La résolution du capteur
- La puissance du capteur
- La valeur maximum que le capteur peut envoyer
- Le délai d'actualisation du capteur

1.3 Exercice 2

Le deuxième exercice affiche sur une ListView une liste de capteurs que ne contient pas le téléphone parmi cette liste :

```
capteursensort.putIfAbsent(TYPE_ACCELEROMETER, "TYPE_ACCELEROMETER");
capteursensort.putIfAbsent(TYPE_AMBIENT_TEMPERATURE, "
    TYPE_AMBIENT_TEMPERATURE");
capteursensort.putIfAbsent(TYPE_GRAVITY, "TYPE_GRAVITY");
capteursensort.putIfAbsent(TYPE_GYROSCOPE, "YPE_GYROSCOPE");
capteursensort.putIfAbsent(TYPE_LIGHT, "TYPE_LIGHT");
capteursensort.putIfAbsent(TYPE_LINEAR_ACCELERATION, "
    TYPE_LINEAR_ACCELERATION");
capteursensort.putIfAbsent(TYPE_MAGNETIC_FIELD, "TYPE_MAGNETIC_FIELD");
capteursensort.putIfAbsent(TYPE_PRESSURE, "TYPE_PRESSURE");
capteursensort.putIfAbsent(TYPE_PROXIMITY, "TYPE_PROXIMITY");
capteursensort.putIfAbsent(TYPE_RELATIVE_HUMIDITY, "TYPE_RELATIVE_HUMIDITY")
    ;
capteursensort.putIfAbsent(TYPE_ROTATION_VECTOR, "TYPE_ROTATION_VECTOR");
```

1.4 Exercice 3

Le but de l'exercice est de changer la couleur de fond de l'application en fonction du mouvement du téléphone.
Ma classe principale est implémentée par l'interface SensorEventListener qui va fournir les méthodes :

```
public void onSensorChanged(SensorEvent sensorEvent)
 protected void onPause()
 protected void onResume()
```

Pour l'exercice je peux utiliser 2 capteurs soit le TYPE_ACCELEROMETER qui est l'accéléromètre basique ou le TYPE_LINEAR_ACCELERATION qui est un accéléromètre qui prend en compte la gravité ce qui rend les résultats plus précis.
Dans la fonction OnResume() j'initialise les capteurs.
Je vérifie d'abord si le téléphone a l'accéléromètre linéaire (Je le prends en priorité car c'est le plus précis) et s'il ne le trouve pas, l'application va utiliser l'accéléromètre basique (et ne fonctionnera pas en affichant un message d'erreur si aucun des deux n'est présent).

```
protected void onResume() {
    super.onResume();
    mSensorManager = (SensorManager) getSystemService(Context.
    SENSOR_SERVICE);
    if (mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION) !=
        null){
        linearAcc = mSensorManager.getDefaultSensor(Sensor
        .TYPE_LINEAR_ACCELERATION);
        mSensorManager.registerListener(this, linearAcc, SensorManager
        .SENSOR_DELAY_NORMAL);
    }
    else {
        Toast.makeText(this,"Il vous manque le linéaire accelerometre"
        , Toast.LENGTH_LONG).show();
        if (mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER) !=
            null){
                accelerometre = mSensorManager.getDefaultSensor(Sensor
                .TYPE_ACCELEROMETER);
                mSensorManager.registerListener(this, accelerometre,
                SensorManager.SENSOR_DELAY_NORMAL);
            }
        else
            Toast.makeText(this,"Il vous manque l'accelerometre"
            , Toast.LENGTH_LONG).show();
    }
}
```

La fonction onSensorChanged(SensorEvent sensorEvent) va être appeler à chaque mise à jour du capteur(Avec un délai initialisé dans la fonction protected void onResume()).

Dans la fonction onSensorChanged je fais la moyenne des 10 dernières valeurs enregistrées par le capteur ce qui permet d'enlever le bruit produit par le capteur et rendre le résultat final plus précis.(J'initialise les valeurs du tableau a 1000 ce qui sera concidéré comme des valeurs nulles donc non prises en compte).

```
tab[m%10] = Math.abs(x)+Math.abs(y) +Math.abs(z);//met la valeur absolue de
    x y z dans le tableau
++m;//va permettre de changer la case du tableau pour la prochaine valeur à
    enregistrer
float somme = 0;
for(int i = 0 ; i < 10 ;++i)
```

```
5       if(tab[i] != 10000)//ne prend pas en compte les valeurs à 1000 car
        valeur nulle
6           somme = tab[i] + somme;
7   float moyenne = somme/10;
```

Puis je change la couleur de fond de l'application en fonction de la valeur du capteur.

```
1   if(moyenne < vert )
2       color.setBackgroundColor(Color.GREEN);
3   else if(moyenne < noir)
4       color.setBackgroundColor(Color.BLACK);
5   else if(moyenne < rouge)
6       color.setBackgroundColor(Color.RED);
7   txt.setText(Float.toString(moyenne));
```

les couleurs sont initialisées avec ces valeurs :

- vert = 20f;
- noir = 40f;
- rouge = 60f;

Et enfin j'affiche au centre de l'écran la valeur de la moyenne calculée de l'accéléromètre.

Quand l'utilisateur quitte l'application la fonction :

```
1   protected void onPause() {
2       mSensorManager.unregisterListener(this, accelerometre);
3       mSensorManager.unregisterListener(this, linearAcc);
4       super.onPause();
5   }
```

Qui va désactiver les capteurs d'accéléromètre et d'accéléromètre linéaire.

1.5 Exercice 4

Pour l'exercice 4 il fallait faire une application qui affiche une flèche qui pointe la direction du mouvement du téléphone. Le code ressemble beaucoup à l'exercice 3 (je choisis le linéaire accéléromètre ou l'accèleromètre basique en fonction de la présence de chacun des capteurs).

Dans le onSensorChanged(SensorEvent sensorEvent) je fais pivoter l'image de la flèche en utilisant les règles de la trigonomètrie en fonction des résultats du capteur à chaque actualisation.

```
if(Math.abs(x) + Math.abs(y) > 10) {
    if (y > 0 && x >= 0)
        image.setRotation((float) Math.toDegrees(Math.atan(x / y) - Math.PI
    ));
    else if (y > 0 && x < 0)
        image.setRotation((float) Math.toDegrees(Math.atan(x / y) + Math.PI
    ));
    else if (y < 0)
        image.setRotation((float) Math.toDegrees(Math.atan(x / y)));
    else if (y == 0 && x > 0)
        image.setRotation((float) Math.toDegrees(Math.PI / 2));
    else if (y == 0 && x < 0)
        image.setRotation((float) Math.toDegrees((3 * Math.PI) / 2));
}
```

Dans cet exercice je n'utilise pas le concept de moyenne car les données sont suffisament précises.
J'ai mis le délai d'actualisation minimal parce que l'application ne nécéssite pas de connaître tout le temps la valeur de l'accéléromètre :

```
mSensorManager.registerListener(this, linearAcc, SensorManager.
    SENSOR_STATUS_ACCURACY_LOW);
```

1.6 Exercice 5

Le but de cet exercice est d'allumer le flash en secouant le téléphone.
Pour accéder au flash du téléphone j'ai mis dans la fonction protected void onResume() le code suivant :

```
camManager = (CameraManager) getSystemService(Context.CAMERA_SERVICE);
try {
    cameraId = camManager.getCameraIdList()[0];
} catch (CameraAccessException e) {
```

```
5       Toast.makeText(this,"Problème d'accès au flash", Toast.LENGTH_LONG).
        show();
6       e.printStackTrace();
7   }
```

Pour augmenter la précision j'ai comme l'exercice 3 fait la moyenne des 10 dernières valeurs enregistrées par le capteur.
Et comme pour l'exercice 3 et 4 je sélectionne l'accéléromètre linéaire ou l'accéléromètre basique en fonction de sa présence.

Dans cette partie de code je vérifie que la moyenne calculée des valeurs fournies par le capteur soit supérieur à 20 avant de faire des modifications.

Le boolean tampon permet de savoir si la valeur du capteur est passée en dessous de 20 (Avant de changer l'état de la lampe le téléphone doit être passé dans un statique pour éviter de faire clignoter la lampe continuellement en secouant le téléphone).
On change l'image de fond et on inverse le boolean "lampe" qui symbolise l'état de la lampe pour allumer ou éteindre la lampe.

```
1   if(moyenne > 20){//vérifie que la moyenne est bien supérieure à 20
2       if(tampon) {//vérifie qu'il y a bien eu un instant où le téléphone é
        tait statique avant de changer l'état de la lampe
3           lampe = !lampe;//change l'état de la lampe
4           tampon = false;
5           /*change l'image en fonction de l'état de la lampe*/
6           if(lampe)
7               image.setImageResource(R.drawable.allumer);
8           else
9               image.setImageResource(R.drawable.eteindre);
10          }
11      }else
12          tampon = true;//remet le tampon a true car il y eu un moment où le
        téléphone était statique
13      /*Met a jour l'état de la lampe*/
14      try {
15          camManager.setTorchMode(cameraId, lampe);
16      } catch (CameraAccessException e) {
17          e.printStackTrace();
18  }
```

1.7 Exercice 6

le but de l'exercice était d'utiliser le capteur de proximité du téléphone. Mon application à chaque fois que le capteur de proximité detecte un object, le téléphone déclenche une petite musique.
Pour ça dans la fonction protected void onResume() je vérifie la présence du capteur de proximité et j'initialise le audioManager pour déclencher la musique :

```
protected void onResume() {
    super.onResume();
    mSensorManager = (SensorManager) getSystemService(Context.
    SENSOR_SERVICE);
    if (mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY) !=null){
        proximiter = mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY)
    ;
        mSensorManager.registerListener(this, proximiter, SensorManager.
    SENSOR_STATUS_ACCURACY_LOW);
    } else
        Toast.makeText(this,"Il manque le capteur de proximité", Toast.
    LENGTH_LONG).show();
    audioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
    this.setVolumeControlStream(streamType);
    if (Build.VERSION.SDK_INT >= 21 ) {
        AudioAttributes audioAttrib = new AudioAttributes.Builder()
                .setUsage(AudioAttributes.USAGE_GAME)
                .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
                .build();
        SoundPool.Builder builder= new SoundPool.Builder();
        builder.setAudioAttributes(audioAttrib).setMaxStreams(MAX_STREAMS);
        this.soundPool = builder.build();
    }
    else
        this.soundPool = new SoundPool(MAX_STREAMS, AudioManager.
    STREAM_MUSIC, 0);
    this.soundIdnotif = this.soundPool.load(this, R.raw.sond,1);//je charge
     la musique à déclencher
}
```

Le délai d'actualisation du capteur de proximité sera lente car l'application ne nécessite pas un délai rapide pour utiliser le capteur.

Dans le onSensorChanged je vérifie la valeur du capteur de proximité (si le capteur rend 0 ça veut dire qu'il y a un objet proche du capteur et à 5 c'est qu'il n'y a pas d'objet proche du capteur).
La variable booleanne musique permet de faire en sorte que la musique ne se déclenche qu'une seule fois quand le capteur est à 0 et ne se redéclenchera qu'une fois que l'objet se sera éloigné du capteur.

```
1  public void onSensorChanged(SensorEvent sensorEvent) {
2      if (sensorEvent.values[0] ==0) {
3          if(musique) {
4              image.setImageResource(R.drawable.sond);//change l'image affich
       é
5              float currentVolumeIndex = (float) audioManager
6                                              .getStreamVolume(streamType);
7              float maxVolumeIndex  = (float) audioManager
8                                              .getStreamMaxVolume(streamType);
9              this.volume = currentVolumeIndex / maxVolumeIndex;//gère le
       volume de la musique
10             this.soundPool.play(this.soundIdnotif, this.volume
11                                     , this.volume, 1, 0, 1f);//permet de dé
       lclencher la musique
12             musique = false;
13         }
14     } else {
15         musique = true;
16         image.setImageResource(R.drawable.mute);//change l'image affichée
17     }
18 }
```

1.8 Exercice 7

Le but de l'exercice était d'afficher la longitude et la latitude du téléphone.J'ai décidé d'ajouter une map avec un point qui montre où se situe l'utilisateur sur la carte.
Pour accéder à la localisation il faut d'abord demander la permission de l'utilisateur à fournir ses donnés. Pour cela je mets dans la fonction protected void onCreate(Bundle savedInstanceState) de l'activité principale :

```
String permission1 = Manifest.permission.ACCESS_FINE_LOCATION;
String permission2 = Manifest.permission.ACCESS_COARSE_LOCATION;
if (!EasyPermissions.hasPermissions(this, permission1)) {
    EasyPermissions.requestPermissions(this, "Our App Requires a permission
    to access your storage", READ_STORAGE_PERMISSION_REQUEST, permission1);
}
if (!EasyPermissions.hasPermissions(this, permission2)) {
    EasyPermissions.requestPermissions(this, "Our App Requires a permission
    to access your storage", READ_STORAGE_PERMISSION_REQUEST, permission2);
}
```

Puis dans la fonction public void onResume() j'initilise le locationManager pour pouvoir ensuite accéder au capteur de geolocalisation :

```
public void onResume() {
    super.onResume();
    locationManager = (LocationManager) this.getSystemService(
    LOCATION_SERVICE);
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.
    ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
    ActivityCompat.checkSelfPermission(this, Manifest.permission.
    ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED)
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER
    , 5000, 10, this);
}
```

Puis dans la fonction public void onLocationChanged(@NonNull Location location) je récupère les valeurs de latitude et de longitude je les affiche sur les TextView et je les donne à la variable mMap qui va pointer la position sur l'utilisateur sur une map.

```
public void onLocationChanged(@NonNull Location location) {
/*change les TextView avec les valeurs de latitude et de longitude*/
    la.setText(location.getLatitude() + "");
    lo.setText(location.getLongitude() + "");
    myposition = new LatLng(location.getLatitude(), location.getLongitude()
    );
    /*supprime le dernier pointeur*/
    markerName.remove();
    /*le recréer avec la nouvelle localisation*/
     markerName = mMap.addMarker(new MarkerOptions()
```

```
10                .position(myposition)
11                .title("Ma position"));
12    }
```

Pour afficher la map j'utilise une API de google.
Malheureusement pour utiliser cette API il faut une clé payante, pour l'exemple fourni j'ai utilisé une clé gratuite pendant 30 jours mais qui sera sans doute périmée quand vous utiliserez l'application.
Pour pouvoir utiliser l'application suivez ce tutoriel :
https://devstory.net/10499/enregistrer-la-google-map-api-key
Puis aller dans le fichier AndroidManifest.xml et remplacer la clé dans la variable android :value de meta-data.

```
1  <meta-data
2       android:name="com.google.android.geo.API_KEY"
3       android:value="MY_KEY"/>//ICI
```

Pour initialiser la map j'implémente l'interface OnMapReadyCallback qui va fournir la méthode public void onMapReady(@NonNull GoogleMap googleMap) qui va me permettre d'initialiser la carte.

```
1  public void onMapReady(@NonNull GoogleMap googleMap) {
2      mMap = googleMap;;//initialise la map
3      /*Permet d'initialiser une première fois la location du téléphone*/
4      Location loc = null;
5      if (ActivityCompat.checkSelfPermission(this, Manifest.permission.
       ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
       ActivityCompat.checkSelfPermission(this, Manifest.permission.
       ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
6          loc = locationManager.getLastKnownLocation(LocationManager.
       GPS_PROVIDER);
7      }
8      la.setText(loc.getLatitude() + "");
9      lo.setText(loc.getLongitude() + "");
10
11
12     /*Plus joli visuellement mais n'a pas besoin de loc.getLatitude() et
       loc.getLongitude()*/
13     //mMap.setMyLocationEnabled(true);
```

```
14
15
16      /*Permet de zoomer sur la carte à l'endroit où se trouve le téléphone*/
17      mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(loc.
        getLatitude(), loc.getLongitude()), 10.0f));
18
19      /*Permet de mettre le pointeur à l'endroit où se trouve le téléphone*/
20      myposition = new LatLng(loc.getLatitude(), loc.getLongitude());
21      markerName = mMap.addMarker(new MarkerOptions()
22              .position(myposition)
23              .title("Ma position"));
24  }
```