Aplicación Android de Asistencia al Caminante

Enrique R. Delgado Garrido

Sensores.

Documento: Teórico-Práctico

Índice de contenido

A3.- Sensores.

Un sensor es un dispositivo que detecta una determinada acción externa, temperatura, presión, etc., y la transmite adecuadamente (RAE).

Desde el punto de vista de Android es una clase publica extensión de la clase Object y que se encuentra en el paquete andoid.harsware.Sensor. En

http://developer.android.com/reference/android/hardware/Sensor.html

tenemos en inglés la descripción.

Si queremos saber los sensores que tiene disponibles nuestro móvil debemos obtener los elementos de la lista que nos proporciona getSensorList(int), pequeño ejemplo que sigue nos los proporciona:

```
SensorManager sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
List<Sensor> list = mSensorManager.getSensorList(Sensor.TYPE_ALL);
for(Sensor sensor:list)
{
      Log.i("Sensor", sensor.getName());
}
```

La siguiente tabla muestra las constantes que podemos utilizar con getSensorList y que ya nos dan una idea de los sensores que podríamos tener a nuestra disposición.

int	TYPE_ACCELEROMETER	Constante para describir un sensor tipo acelerómetro
int	TYPE_ALL	Constante que describe todos los tipos de sensores
int	TYPE_GYROSCOPE	Constante para describir un sensor tipo giróscopo
int	TYPE_LIGHT	Constante para describir un sensor tipo luminoso
int	TYPE_MAGNETIC_FIELD	Constante para describir un sensor de campo magnético
int	TYPE_ORIENTATION	Constante descatalogada se debe utilizar la instancia SensorManager.getOrientation
int	TYPE_PRESSURE	Constante para describir un sensor de presión
int	TYPE_PROXIMITY	Constante para describir un sensor de proximidad
int	TYPE_TEMPERATURE	Constante para describir un sensor de temperatura

¿Qué debemos hacer para poder utilizar los sensores en un programa veamos el ejemplo para Caminante_DATOS:

Definir variables:

```
private      SensorManager                  sensorManager;
```

```
        private     Sensor                      accelerometer;
        private     Sensor                      orientation;
```

En el ejemplo anterior ya vimos que aparecía la clase SensorManager esta extiende también a Object y utiliza android.harware.SensorManager. Esta clase es la que nos permite acceder a los sensores y Context.getSyestemService() con el argumento SERSOR_SERVICE nos permite obtener una instancia de la clase. Ver

http://developer.android.com/reference/android/hardware/SensorManager.html

```
public void onCreate(Bundle savedInstanceState) {
      ...
      // módulo para inicializar lo concerniente a los sensores
      setSensors();
      ...
}

protected void setSensors() {
      sensorManager = (SensorManager)getSystemService(Context.SENSOR_SERVICE);
      List<Sensor> list;
      //Obtención del Sensor Acelerómetro
      list=sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);
      if (list.size()>0) accelerometer=list.get(0);
      //Obtención del Sensor Orientación
      list=sensorManager.getSensorList(Sensor.TYPE_ORIENTATION);
      if (list.size()>0) orientation=list.get(0);
      ...
}
```

Ya tenemos la instanciados los sensores ahora tenemos que ponerlos en escucha:

```
public void onClick(View vista) {
      ...
      if(vista.getId()==R.id.buttonStart) {
            ...
            onResumeSensors();
            ...
      }
      // Botón de terminación el proceso de captura de datos
      if(vista.getId()==R.id.buttonFinish) {
            ...
            stopProcess();
            ...
      }
}
/** módulo para el comienzo real del uso de los sensores por parte del programa
 */
//@Override
protected void onResumeSensors() {
```

```
      super.onResume();

      //Registro de los escuchadores
      if (accelerometer!=null) {
            sensorManager.registerListener(sensorEventListener,
                        //acelerometro,SensorManager.SENSOR_DELAY_NORMAL);
                        accelerometer,SensorManager.SENSOR_DELAY_FASTEST);
      }

      if (orientation!=null) {
            sensorManager.registerListener(sensorEventListener,
                        //orientacion,SensorManager.SENSOR_DELAY_NORMAL);
                        orientation,SensorManager.SENSOR_DELAY_FASTEST);
      }
}
/** módulo para la recepción de los valores de los sensores cuando se detecta
 *  un evento de cambio en dichos valores
 */
private SensorEventListener sensorEventListener = new SensorEventListener() {
      // variables de coordenadas
      public      float[] coordinatesAccelerometer    = new float[3];
      public      float[] coordinatesOrientation            = new float[3];
      // modulo para capturar los eventos de cambio de precisión
      // no se usa
      //@Override
      public void onAccuracyChanged(Sensor sensor, int accuracy) {
            // TODO Auto-generated method stub
      }
      // modulo para capturar los eventos de cambio de valor
      //@Override
      public void onSensorChanged(SensorEvent event) {

            //implantación de los escuchadores que sincronizamos
            synchronized (this) {
                  //damos formato de número float a los valores de los eventos
                  for (int i=0;i<3;i++) {
                        int w=(int)(10*event.values[i]);
                        event.values[i]=(float)(w/10.0f);
                  }
                  //actualización de los valores dependiendo del tipo de sensor
                  if (event.sensor==accelerometer) {
                        this.setAccelerometer(event.values);
                        //escritura los dato en el fichero del acelerómetro
                        // por medio del objeto de la clase FicherosCAMINANTE
                        dataFiles.addDatoACEL(coordinatesAccelerometer[0],
                                                coordinatesAccelerometer[1],
                                                coordinatesAccelerometer[2]);
                  } else {
                        this.setAccelerometer(coordinatesZero);
```

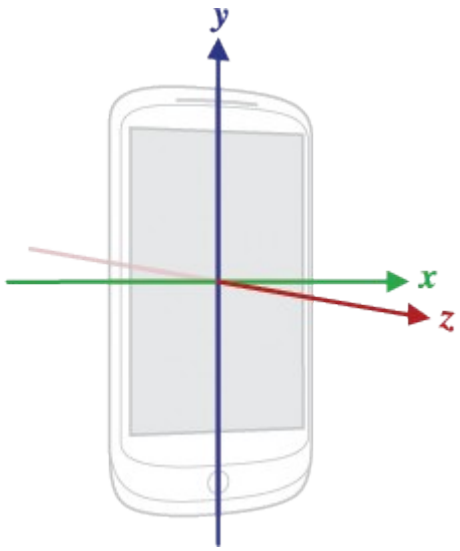
```
                  }
                  if (event.sensor==orientation) {
                        this.setOrientacion(event.values);
                        //escritura los dato en el fichero de orientación
                        // por medio del objeto de la clase FicherosCAMINANTE
                        dataFiles.addDatoORIE(coordinatesOrientation[0],
                                              coordinatesOrientation[1],
                                              coordinatesOrientation[2]);
                        } else {
                              this.setOrientacion(coordinatesZero);
                        }
                  }
            }
            public void setAccelerometer (float[] coordinates) {
                  this.coordinatesAccelerometer=coordinates;
            }
            public void setOrientacion(float[] coordinates) {
                  this.coordinatesOrientation=coordinates;
            }
      };

      protected void stopProcess() {
            // elimina de escucha los sensores
            sensorManager.unregisterListener(sensorEventListener);
            ...
      }

/** módulo para las paradas fuera del botón terminar
 */
      protected void onDestroy() {
            super.onDestroy();
            if (sensorrunning) {
                  stopProcess();
            }
      }
}
```

Aparece en el código anterior SensorEventListener que nos la proporciona android.hardware.SensorEventListener y que es una clase importante se utiliza para recibir la notificaciones de los sensores cuando estos han cambiado, tiene dos métodos void abstractos:

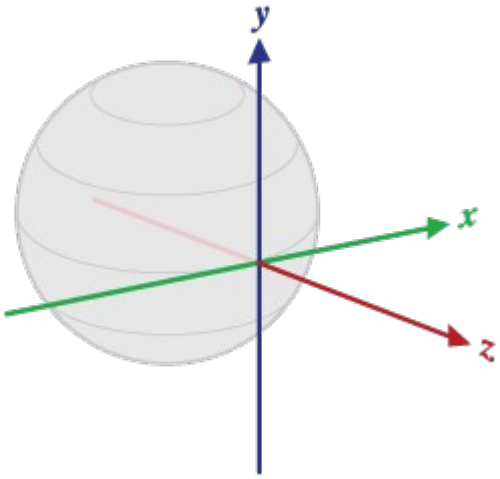
onAccuracyChanged (Sensor sensor, int accuracy) que se llama cuando la exactitud de un sensor ha cambiado.

onSensorChanged (SensorEvent event) que se le llama cuando valores de los sensores han cambiado.

A.3.1.- Coordenadas Acelerómetro. Este sensor nos proporciona tres valores que corresponden, a los tres ejes cartesianos, cada uno tiene su situación fija con respecto al móvil. La siguiente figura proveniente del fichero del SDK de Android nos los muestra.



Puede ser conveniente también tener la referencia de como consideramos los ejes terrestres y que coinciden con los del teléfono.



La correspondencia con lo obtenidos del programa es:

- eje X se corresponde con la coordenada 0
- eje Y se corresponde con la coordenada 1
- eje Z se corresponde con la coordenada 2

Los valores están en el SI, es decir en m/s^2.

Hay que hacer constar la aceleración de la gravedad también es detectada, así por ejemplo si ponemos horizontalmente le teléfono en una superficie con el eje Z perpendicular a la superficie terrestre el sensor deberá marcar el valor de la gravedad en ese eje y cero en los otros

A.3.2.- Coordenadas orientación. Los ejes son los anteriores pero los valores en este caso son grados y su significado es el siguiente:

– coordenada 0 Azimut corresponde en grados al giro al rededor del eje Z (0-259). Valor 0 cuando el eje Y apunta al norte. (0 = Norte, 90 = Este, 180 = Sur, 270 = Oeste)

– coordenada 1 Pitch (cabeceo) corresponde en grados a la rotación alrededor del eje X. Valor positivo cuando el eje Z se mueve hacia el eje Y. (-180 a 180)

– coordenada 2 Roll (balanceo) corresponde en grados a la rotación alrededor del eje Y. Valor positivo cuando el eje X se mueve hacia el eje Z. (-90 a 90).