



Laboratorio

Android: accelerometro

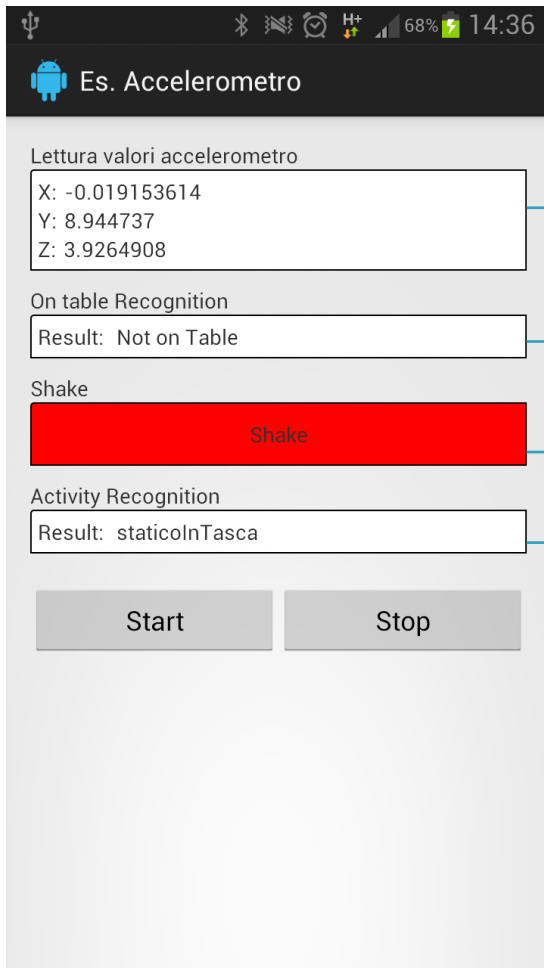
Corso di Smart City e Tecnologie Mobili

Università di Bologna

Dipartimento di Informatica — Scienza e Ingegneria

Luca Calderoni, Andrea Cirri, Dario Maio

Obiettivi



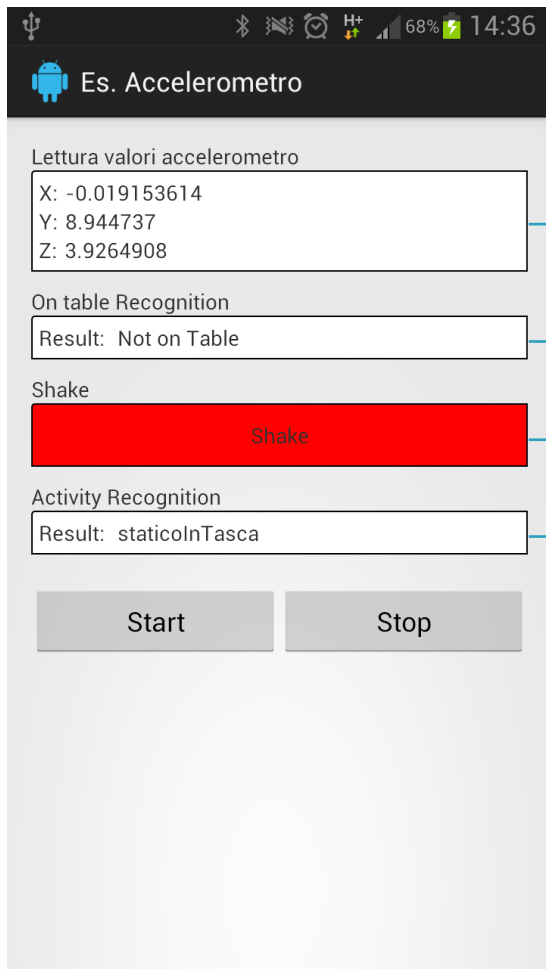
Visualizzare le coordinate rilevate dall'accelerometro (accelerazioni sugli assi x , y , z).

Visualizzare un indicatore che permetta di capire se il dispositivo si trovi fermo su un tavolo o no.

La casella di testo *Shake* deve «vibrare» quando il dispositivo viene scosso bruscamente.

Mostrare il risultato derivante da un classificatore che distingua tra le seguenti casistiche: *dispositivo in tasca e utente fermo*, *dispositivo sul tavolo*, *l'utente sta camminando*, *l'utente sta correndo*.

Aggiornamento dell'interfaccia



```
public void setAccelerometerValues(String x, String  
y, String z);
```

```
public void setOnTableRecognitionResult(boolean  
result);
```

```
public void shake();
```

```
public void setActivityRecognitionResult(String  
result);
```

Per visualizzare a video i valori inferiti dal sensore si possono utilizzare i metodi sopra citati, già implementati nel progetto scheletro.

Sensor Event Listener

Per rilevare le variazioni di stato di un sensore, Android si avvale dell'interfaccia Sensor Event Listener. L'oggetto che la implementa sarà notificato a ogni campionamento operato dal sensore.

```
public interface SensorEventListener() {  
  
    //Nuovi valori rilevati  
    //Nota: l'oggetto event fa parte di un pool di risorse e può essere riutilizzato  
    //dal sistema senza preavviso, quindi è consigliabile salvare localmente i  
    //valori in esso contenuti ed elaborarli successivamente.  
    public void onSensorChanged(SensorEvent event);  
  
    //Richiamato quando l'accuratezza del sensore varia.  
    //Accuracy può assumere i valor «HIGH», «MEDIUM», «LOW»  
    public void onAccuracyChanged(Sensor sensor, int accuracy);  
  
}
```

Sensor Event

L'oggetto Sensor Event che perviene al metodo onSensorChanged contiene 4 parametri:

- **Int** *accuracy*: Grado di accuratezza col quale sono stati campionati i valori («LOW», «MEDIUM», «HIGH»).
- **Sensor** *sensor*: Il sensore che ha generato l'evento. Per conoscerne il tipo, *sensor.getType()*;
- **Long** *timestamp*: Timestamp relativo all'evento.
- **Final float[]** *values*: Array dei valori effettivi rilevati, dipendenti dal tipo di sensore.

Nel caso dell'accelerometro il tipo di sensore è «TYPE_ACCELEROMETER» e i valori si ricavano nel seguente modo:

- **float** *x* = *event.values*[0];
- **float** *y* = *event.values*[1];
- **float** *z* = *event.values*[2];

Accelerazione: alcune considerazioni

In realtà, parlare di accelerazione quando ci riferiamo ai valori restituiti dal sistema Android è **formalmente scorretto** in quanto tali valori rappresentano la proper acceleration (g-force).

Per ricavare l'accelerazione lineare effettiva su ciascun asse occorre quindi sottrarre l'accelerazione di gravità, che deve essere scomposta nelle tre componenti relative a ciascun asse.

```
//calcolo l'accelerazione effettiva su ciascun asse
//values[0]: Acceleration minus G_x on the x-axis
//values[1]: Acceleration minus G_y on the y-axis
//values[2]: Acceleration minus G_z on the z-axis
```

A tal fine, Android mette a disposizione dello sviluppatore una diversa costante con la quale istanziare il *SensorManager*, che comporta la restituzione dei reali valori di accelerazione sui tre assi: «TYPE_LINEAR_ACCELERATION»

Registrazione al sensore

L'oggetto che implementa l'interfaccia *SensorEventListener* deve registrarsi al sensore voluto per poter ricevere gli aggiornamenti di stato.

A tal fine si usa la classe *SensorManager*, grazie alla quale è possibile associare un *Listener* a uno o più sensori.

```
//Nota: getSystemService è un metodo che può essere invocato solo su un Context (ad esempio all'interno
//di una Activity).

SensorManager sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

Sensor sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
//In alternativa:
//Sensor sensor = sensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

//Registrazione di un Listener
//SensorEventListener myListener
//SENSOR_DELAY_FASTEST: Rileva il massimo numero di campionamenti possibili.
//Altri valori possono essere: SENSOR_DELAY_GAME, SENSOR_DELAY_NORMAL, SENSOR_DELAY_UI.
sensorManager.registerListener(myListener, sensor, SensorManager.SENSOR_DELAY_FASTEST);

//Rilascio di un Listener
sensorManager.unregisterListener(myListener);
```

Consigli per l'implementazione

Per praticità, l'interfaccia *SensorEventListener* si può implementare direttamente nell'*Activity* principale dell'applicazione.

La registrazione e il rilascio del *Listener* devono essere collegati ai pulsanti «Start» e «Stop» presenti nell'applicazione scheletro.

```
//Nota al ciclo di vita delle Activity:  
//è opportuno interrompere la ricezione degli aggiornamenti dai sensori quando l'applicazione passa in  
//secondo piano o viene sospesa (ad esempio in caso di disattivazione del display).  
  
onPause();  
onResume();  
  
//Nota ai valori di accelerazione: i valori rilevati sugli assi x, y e z comprendono il valore indotto  
dalla forza di gravità (9,8 m/s2).
```


Implementazione: valori

Estrapolo i valori dall'array dell'oggetto *SensorEvent* e li mostro a video con il metodo opportuno.

```
float x = event.values[0];  
float y = event.values[1];  
float z = event.values[2];  
  
//visualizzazione nell'interfaccia grafica su TextView  
setAccelerometerValues(x, y, z);
```

Implementazione: onTable

Controllo che il valore restituito sull'asse z sia circa $9,8 \text{ m/s}^2$ e che quelli relativi agli assi x e y siano pressoché nulli.

```
//desumo se il telefono si trova su un piano parallelo alla superficie terrestre
if((z >= 9.6f && z <= 10.4f) && (Math.abs(x) + Math.abs(y)) <= 0.5f)
{
    setOnTableRecognitionResult(true);
}
else
{
    setOnTableRecognitionResult(false);
}
```

Implementazione: Shake

```
currentUpdate = System.currentTimeMillis();

//Massimo un update ogni 100ms.
if ((currentUpdate - lastUpdate) > 100)
{
    long diffTime = (currentUpdate - lastUpdate);
    lastUpdate = currentUpdate;
    float speed = Math.abs(x + y + z - last_x - last_y - last_z) / diffTime * 10000;

    //SHAKE_THRESHOLD è inizialmente settata a 2000
    if (speed > SHAKE_THRESHOLD) shake();
    last_x = x;
    last_y = y;
    last_z = z;
}
```

Implementazione: classificatore

Nell'applicazione scheletro è incluso un classificatore rudimentale che può essere usato per analizzare i diversi stati nei quali si trova il dispositivo in base ai valori ricavati dall'accelerometro.

```
classifier = new Classifier();  
  
...  
String res = classifier.classify(x, y, z);  
  
if(!res.isEmpty())  
{  
    setActivityRecognitionResult(res);  
}
```