Sensore

Molto spesso ci è capitato di notare che il dispositivo "si accorge" di una serie di fattori ed eventi fisici: se lo giriamo, se lo scuotiamo e via dicendo. Altre volte si può essere rimasti stupiti notando che alcune app forniscono informazioni sull'ambiente in cui ci troviamo. Percepiscono magari temperatura, umidità, luminosità. I dispositivi Android grazie ai sensori di cui sono forniti riescono a percepire movimenti, condizioni ambientali e lo faranno sempre più e sempre con maggiore precisione grazie all'ampliamento costante di queste tecnologie.

Innanzitutto, i sensori possono essere suddivisi in tre grandi gruppi:

- sensori di **movimento**: percepiscono le forze fisiche che agiscono sul dispositivo. Ad esempio, l'accelerometro, il giroscopio, sensore di gravità;
- sensori **ambientali**: rilevano particolari dell'ambiente in cui ci si trova: temperatura, pressione, umidità;
- sensori di **posizione**: raccolgono dati sulla posizione del dispositivo, ad esempio il sensore di orientamento.

I sensori software sono chiamati anche virtuali in quanto possono essere consultati con lo stesso interfacciamento di quelli hardware dissimulando quindi la loro natura software.

Il SensorManager

Come in molte altre situazioni, il sottosistema Android che vogliamo sfruttare ci viene dischiuso da un *system service*, accessibile mediante una classe "manager". In questo caso, si tratta del SensorManager. La prima cosa che può essere utile fare con il SensorManager è chiedergli un inventario dei sensori disponibili nel nostro dispositivo. Usando una serie di costanti intere (tutte ben spiegate nella documentazione ufficiale) si può chiedere una lista dei sensori. Un po' tutti i dispositivi avranno a disposizione almeno tre o quattro sensori essenziali per la vita di uno smartphone tra cui accelerometro, orientamento e rotazione.

Leggere dati da un sensore

La prassi comune per ricevere dati periodici da un sensore è **registrare un listener** nella nostra applicazione. Ciò, da un punto di vista sintattico, obbligherà all'implementazione di un metodo di callback all'interno del quale si potrà fare un qualche uso delle misurazioni rilevate.

Gli aspetti da notare maggiormente sono:

- nel metodo onCreate è stato prelevato un riferimento al SensorManager. Opzionalmente questo punto sarà buono per recuperare un riferimento anche al sensore specifico con cui si vuole interagire;
- nei metodi onPause e onResume che come sappiamo regolano l'inizio e la fine dell'interazione tra Activity e utente avviene, rispettivamente, la registrazione e la cancellazione del listener;
- l'activity implementa l'interfaccia SensorEventListener che forza all'override di due metodi onAccuracyChanged e onSensorChanged.

Il metodo **onSensorChanged** costituisce il cuore dell'interazione con il sensore. È qui che arrivano le chiamate del listener ogni volta che sono disponibili nuove misurazioni. L'evento notificato verrà formalizzato con un oggetto di classe SensorEvent.

SensorEvent permette di leggere i valori recuperati come un array numerico. Il tutto visto in questo modo potrebbe sembrare semplice. La difficoltà sta proprio nell'interpretare e sfruttare i valori dell'evento. Essendo i sensori dei misuratori di grandezza fisiche, i dati letti con essi dovrebbero essere sottoposti ad opportune valutazioni nel rispetto, eventualmente, di leggi scientifiche.