

# Corso di Laboratorio di Programmazione per Sistemi Mobile e Tablet

Docente: Dr. Mauro Dragoni

Dipartimento di Ingegneria e Scienze dell'Informazione  
Anno Accademico 2021/2022

# | in questa lezione...

- Sensori
- Location

# Sensori

- Motions sensors (misurano il movimento del device)
  - Accelerometro
  - Giroscopio
  - Sensore di gravita'
  - Sensore di rotazione
- Environmental sensors (misurano condizioni ambientali)
  - Barometro
  - Fotometro
  - Termometro
- Position sensors (misurano la posizione fisica del device)
  - Magnetometro
  - Sensore di prossimita'

# Implementazione

- SensorManager
  - permette di gestire l'accesso e l'ascolto dei sensori
  - registra e de-registra i listener degli eventi
  - acquisisce informazioni riguardo all'orientamento del dispositivo
  - fornisce le costanti relative all'accuratezza, frequenza di acquisizione e calibrazione dei sensori
- Sensor: determina le capacità di specifici sensori
- SensorEvent: fornisce le informazioni relative agli eventi, inclusi i dati grezzi
- SensorEventListener: riceve notifiche circa gli eventi riguardanti i sensori (es. quando i sensori hanno nuovi dati o quando l'accuratezza di un sensore cambia).

# Roadmap per l'utilizzo dei sensori

- Determinare quali sensori sono disponibili sul dispositivo
- Determinare le capacità di ciascun sensore (es. intervallo di valori, risoluzione, requisiti di energia)
- Registrare i listener relativi agli eventi
- Acquisire i dati grezzi
- De-registrare i listener relativi agli eventi

# Roadmap per l'utilizzo dei sensori

- Sfruttare Google Play per filtrare i dispositivi in base alla disponibilità dei sensori
- Usare il tag `<uses-feature>` nel manifesto di Android
- Esempio:

```
<uses-feature  
    android:name="android.hardware.sensor.accelerometer"  
    android:required="true" />
```

# Registrazione dei listener

- Quando registrarli?
- Se si (de-)registrano all'interno dei metodi `onStart()` e `onStop()` il sensore utilizzerà la batteria anche quando l'applicazione non è in foreground.
- In alternativa, implementare i listeners nei metodi `onResume()` e `onPause()`

# Localizzazione

## ■ Cosa serve?

- Controllare di avere tutti i permessi
- Chiedere quelli necessari

## ■ Cosa possiamo fare?

- Chiedere la location piu' recente
- Chiedere gli aggiornamenti

## ■ Best practice: utilizzare la classe

FusedLocationProviderClient

- combina GPS, Wi-Fi, e celle telefoniche per stabilire la posizione
- bilancia l'accuratezza e la velocita' del posizionamento con un occhio d'attenzione alla batteria
- restituisce un oggetto di tipo `Location` contenente latitudine e longitudine
- utilizzare la classe Geocoder per la conversion lat/long → indirizzo e viceversa

# Localizzazione - Autorizzazioni

- Da Marshmallow in avanti:
  - gli utenti possono dare e togliere l'accesso alla localizzazione per ciascuna app
  - gli utenti possono modificare i permessi in qualsiasi momento
  - la vostra applicazione puo' chiedere all'utente di fornire l'accesso alla localizzazione
  
- Che autorizzazioni ci servono?
  - ACCESS\_COARSE\_LOCATION (per accuratezza grezza)
  - ACCESS\_FINE\_LOCATION (per accuratezza precisa)
  - ACCESS\_INTERNET (per sfruttare il Wi-Fi)
  - ACCESS\_NETWORK\_STATE (per sfruttare i ripetitori telefonici)
  
- **Controllare sempre la presenza o meno delle autorizzazioni!!!**

# Approfondimenti

Sul sito ufficiale:

- [Risorsa ufficiale \(Sensori\)](#)
- [Risorsa ufficiale \(Localizzazione\)](#)