TESINA

per il corso di Laurea

*PROGRAMMAZIONE DI SISTEMI MOBILE*

***SENSORI DI MOVIMENTO***

***e SISTEMA DI GEOLOCALIZZAZIONE***

***DEI SISTEMI MOBILE ANDROID***

Marica Pasquali ( matricola : 0000802302 )

**Indice**

1. Sensori di movimento
   1. Sensori di progetto open source Android
   2. Sensore di gravità
   3. Accelerazione lineare
   4. Vettore di rotazione
   5. Sensore di movimento significativo
   6. Contapassi
   7. Rilevatore di passi
   8. Lavorazione con dati non elaborati ( RAW DATA )
      1. Accelerometro
      2. Giroscopio
      3. Giroscopio non calibrato
2. GPS
   1. Il GPS su dispositivi mobili
   2. Location Data
      1. Autorizzazioni di posizione
      2. My Location Layer
      3. L'API di localizzazione dei servizi di Google Play
      4. LocationSource
3. **SENSORI DI MOVIMENTO**

La piattaforma Android offre diversi sensori che ti consentono di monitorare il movimento di un dispositivo.

Le possibili architetture dei sensori variano in base al tipo di sensore:

1. I sensori di gravità, accelerazione lineare, vettore di rotazione, movimento significativo, contapassi e rilevatore di passi sono basati su hardware o software.
2. I sensori dell'accelerometro e del giroscopio sono sempre basati su hardware.

La maggior parte dei dispositivi Android ha un accelerometro e molti ora includono un giroscopio. La disponibilità dei sensori basati su software è più variabile perché spesso fanno affidamento su uno o più sensori hardware per ricavarne i dati. A seconda del dispositivo, questi sensori basati su software possono derivare i loro dati dall'accelerometro e dal magnetometro o dal giroscopio.

I sensori di movimento sono utili per monitorare il movimento del dispositivo, come inclinazione, vibrazione, rotazione o oscillazione. Il movimento di solito è un riflesso dell'input dell'utente diretto (ad esempio, un utente che guida un'auto in un gioco o un utente che controlla una palla in un gioco), ma può anche essere un riflesso dell'ambiente fisico in cui si trova il dispositivo (ad esempio, spostarti con te mentre guidi la tua auto). Nel primo caso, si sta monitorando il movimento relativo al quadro di riferimento del dispositivo o al quadro di riferimento dell'applicazione; nel secondo caso stai monitorando il movimento relativo al quadro di riferimento del mondo.

I sensori di movimento da soli non vengono in genere utilizzati per monitorare la posizione del dispositivo, ma possono essere utilizzati con altri sensori, come il sensore di campo geomagnetico, per determinare la posizione di un dispositivo rispetto al quadro di riferimento del mondo (vedere Sensori di posizione per ulteriori informazioni).

Tutti i sensori di movimento restituiscono array multidimensionali di valori dei sensori per ciascun SensorEvent.

La tabellaseguente riassume i sensori di movimento disponibili sulla piattaforma Android.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Sensor*** | ***Sensor event data*** | ***Description*** | ***Units of measure*** |
| [TYPE\_ACCELEROMETER](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER) | SensorEvent.values[0] | Acceleration force along the x axis (including gravity). | m/s2 |
| SensorEvent.values[1] | Acceleration force along the y axis (including gravity). |
| SensorEvent.values[2] | Acceleration force along the z axis (including gravity). |
| [TYPE\_ACCELEROMETER\_UNCALIBRATED](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER_UNCALIBRATED) | SensorEvent.values[0] | Measured acceleration along the X axis without any bias compensation. | m/s2 |
| SensorEvent.values[1] | Measured acceleration along the Y axis without any bias compensation. |
| SensorEvent.values[2] | Measured acceleration along the Z axis without any bias compensation. |
| SensorEvent.values[3] | Measured acceleration along the X axis with estimated bias compensation. |
| SensorEvent.values[4] | Measured acceleration along the Y axis with estimated bias compensation. |
| SensorEvent.values[5] | Measured acceleration along the Z axis with estimated bias compensation. |
| [TYPE\_GRAVITY](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GRAVITY) | SensorEvent.values[0] | Force of gravity along the x axis. | m/s2 |
| SensorEvent.values[1] | Force of gravity along the y axis. |
| SensorEvent.values[2] | Force of gravity along the z axis. |
| [TYPE\_GYROSCOPE](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GYROSCOPE) | SensorEvent.values[0] | Rate of rotation around the x axis. | rad/s |
| SensorEvent.values[1] | Rate of rotation around the y axis. |
| SensorEvent.values[2] | Rate of rotation around the z axis. |
| [TYPE\_GYROSCOPE\_UNCALIBRATED](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GYROSCOPE_UNCALIBRATED) | SensorEvent.values[0] | Rate of rotation (without drift compensation) around the x axis. | rad/s |
| SensorEvent.values[1] | Rate of rotation (without drift compensation) around the y axis. |
| SensorEvent.values[2] | Rate of rotation (without drift compensation) around the z axis. |
| SensorEvent.values[3] | Estimated drift around the x axis. |
| SensorEvent.values[4] | Estimated drift around the y axis. |
| SensorEvent.values[5] | Estimated drift around the z axis. |
| [TYPE\_LINEAR\_ACCELERATION](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_LINEAR_ACCELERATION) | SensorEvent.values[0] | Acceleration force along the x axis (excluding gravity). | m/s2 |
| SensorEvent.values[1] | Acceleration force along the y axis (excluding gravity). |
| SensorEvent.values[2] | Acceleration force along the z axis (excluding gravity). |
| [TYPE\_ROTATION\_VECTOR](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ROTATION_VECTOR) | SensorEvent.values[0] | Rotation vector component along the x axis (x \* sin(θ/2)). | Unitless |
| SensorEvent.values[1] | Rotation vector component along the y axis (y \* sin(θ/2)). |
| SensorEvent.values[2] | Rotation vector component along the z axis (z \* sin(θ/2)). |
| SensorEvent.values[3] | Scalar component of the rotation vector ((cos(θ/2)).1 |
| [TYPE\_SIGNIFICANT\_MOTION](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_SIGNIFICANT_MOTION) | N/A | N/A | N/A |
| [TYPE\_STEP\_COUNTER](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_STEP_COUNTER) | SensorEvent.values[0] | Number of steps taken by the user since the last reboot while the sensor was activated. | Steps |
| [TYPE\_STEP\_DETECTOR](https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_STEP_DETECTOR) | N/A | N/A | N/A |

Immagine che contiene interni, tavolo

Descrizione generata automaticamenteIl sensore vettoriale di rotazione e il sensore di gravità sono i sensori più utilizzati per il rilevamento e il monitoraggio del movimento. Il sensore vettoriale rotazionale è particolarmente versatile e può essere utilizzato per una vasta gamma di attività legate al movimento, come il rilevamento di gesti, il monitoraggio dei cambiamenti angolari e il monitoraggio dei relativi cambiamenti di orientamento. Ad esempio, il sensore vettoriale rotazionale è ideale se si sta sviluppando un gioco, un'applicazione di realtà aumentata, una bussola bidimensionale o tridimensionale o un'app di stabilizzazione della videocamera. Nella maggior parte dei casi, l'utilizzo di questi sensori è una scelta migliore rispetto all'utilizzo dell'accelerometro e del sensore di campo geomagnetico o del sensore di orientamento.

***Sensori di progetto open source Android***

Il progetto Android Open Source (AOSP) fornisce tre sensori di movimento basati su software: un sensore di gravità, un sensore di accelerazione lineare e un sensore vettoriale di rotazione. Questi sensori sono stati aggiornati in Android 4.0 e ora utilizzano un giroscopio di un dispositivo (oltre ad altri sensori) per migliorare la stabilità e le prestazioni. Se si desidera provare questi sensori, è possibile identificarli utilizzando il metodo getVendor () e il metodo getVersion () (il fornitore è Google LLC; il numero di versione è 3). L'identificazione di questi sensori per fornitore e numero di versione è necessaria perché il sistema Android considera questi tre sensori come sensori secondari. Ad esempio, se un produttore di dispositivi fornisce il proprio sensore di gravità, il sensore di gravità AOSP viene visualizzato come sensore di gravità secondario. Tutti e tre questi sensori si basano su un giroscopio: se un dispositivo non ha un giroscopio, questi sensori non vengono visualizzati e non sono disponibili per l'uso.

**Sensore di gravità**

Il sensore di gravità fornisce un vettore di tre dimensioni ed indica la direzione e la magnitudine della gravità.

Immagine che contiene screenshot

Descrizione generata automaticamente

Le unità sono uguali a quelle utilizzate dal sensore di accelerazione (m / s2) e il sistema di coordinate è uguale a quello utilizzato dal sensore di accelerazione.

Quando un dispositivo è a riposo, l'output del sensore di gravità dovrebbe essere identica a quella dell'accelerometro.

**Accelerazione lineare**

Il sensore dell’accelerazione lineare fornisce un vettore tridimensionale rappresentante l’accelerazione su ogni asse del dispositivo (esclusa la gravità).

Immagine che contiene screenshot

Descrizione generata automaticamente

L’equivalenza corrisponde a:



In genere l’accelerazione lineare viene usata quando si vuole ottenere l’accelerazione senza l’influenza della gravità (ad es. velocità macchina).

Il sistema di coordinate del sensore è uguale a quello utilizzato dal sensore di accelerazione, così come le unità di misura (m / s2).

**Vettore di rotazione**

Il vettore di rotazione rappresenta l’orientamento del dispositivo come una combinazione di un angolo ed un asse, nel quale il dispositivo viene ruotato attraverso un angolo intorno ad un asse (x, y o z).

Immagine che contiene screenshot

Descrizione generata automaticamente

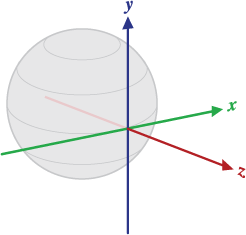
Si tratta di un sensore software, se un’applicazione richiede questo sensore verranno utilizzati l’accelerometro, la bussola e, se disponibile, il giroscopio.

Figura - Sistema di coordinate utilizzato dal

I tre elementi del vettore di rotazione sono espressi come segue:



Dove l'ampiezza del vettore di rotazione è uguale a sin (θ / 2) e la direzione del vettore di rotazione è uguale alla direzione dell'asse di rotazione.

I tre elementi del vettore di rotazione sono uguali agli ultimi tre componenti di un'unità quaternione

**( , , ).**

Gli elementi del vettore di rotazione sono senza unità. Gli assi *x, y* e *z* sono definiti allo stesso modo del sensore di accelerazione. Il sistema di coordinate di riferimento è definito come base ortonormale diretta (vedi figura 1). Questo sistema di coordinate ha le seguenti caratteristiche

* X è definito come il prodotto vettoriale Y x Z. È tangenziale al suolo nella posizione corrente del dispositivo e punta approssimativamente ad est.
* Y è tangenziale al suolo nella posizione corrente del dispositivo e punta verso il Polo Nord geomagnetico.
* Z punta verso l'alto ed è perpendicolare al piano terra.

**Sensore di movimento significativo**

Il sensore di movimento attiva un evento ogni volta che un movimento importante viene rilevato, dopo il quale si disabilita. Per movimento significativo si intende un movimento che potrebbe portare ad un cambiamento nella posizione dell’utente, per esempio camminare, andare in bicicletta, o stare seduti in un’automobile in movimento.

Il codice seguente mostra come ottenere un’istanza del sensore di movimento significativo di default e come registrare un listener di eventi:

Immagine che contiene screenshot

Descrizione generata automaticamente

**Contapassi**

Il contatore di passi, o Counter Step, fornisce il numero di passi fatti dall’utente dall’ultimo riavvio. Questo sensore ha una grande latenza (più di 10 secondi) ma è molto più accurato rispetto al Step Detector.

Immagine che contiene screenshot

Descrizione generata automaticamente

Per preservare la durata della batteria sui dispositivi che eseguono l'app, è necessario utilizzare la classe *JobScheduler* per recuperare il valore corrente dal sensore contapassi a un intervallo specifico.

Sebbene diversi tipi di app richiedano intervalli di lettura del sensore diversi, è necessario mantenere questo intervallo il più lungo possibile a meno che l'app non richieda dati in tempo reale dal sensore.

**Rilevatore di passi**

Registrandosi sul sensore per il rilevamento dei passi, o Step Detector, si riceva un evento ogni volta che l’utente effettua un passo (latenza sotto i 2 secondi).

Immagine che contiene screenshot

Descrizione generata automaticamente

La differenza tra contapassi e il rilevatore di passi è che il primo conta il numero di passi, il secondo invece ci segnala quando viene effettuato un nuovo passo.

Per questi ultimi due sensori, si deve dichiarare l'autorizzazione *ACTIVITY\_RECOGNITION* per consentire alla app di utilizzare questo sensore su dispositivi con Android 10 (livello API 29) o superiore.

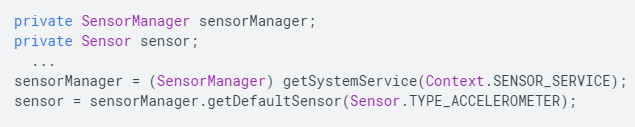
*<uses-permission android:name="android.permission.ACTIVITY\_RECOGNITION"/>*

**Lavorazione con dati non elaborati ( RAW DATA )**

I seguenti sensori forniscono alla tua app dati grezzi sulle forze lineari e rotazionali applicate al dispositivo. Per utilizzare efficacemente i valori di questi sensori, è necessario filtrare i fattori dall'ambiente, come la gravità. Potrebbe anche essere necessario applicare un algoritmo di livellamento all'andamento dei valori per ridurre il rumore

**Accelerometro**

Il sensore di accelerazione misura l’accelerazione applicata al dispositivo (inclusa la forza di gravità):



Concettualmente, un sensore di accelerazione determina l'accelerazione che viene applicata a un dispositivo (Ad) misurando le forze che vengono applicate al sensore stesso (Fs) usando la seguente relazione:

Tuttavia, la forza di gravità influenza sempre l'accelerazione misurata in base alla seguente relazione:

Per questo motivo, quando il dispositivo è seduto su un tavolo (e non sta accelerando), l'accelerometro legge una magnitudine di g = 9,81 m / s2. Allo stesso modo, quando il dispositivo è in caduta libera e quindi in rapida accelerazione verso terra a 9,81 m / s2, il suo accelerometro legge una magnitudine di g = 0 m / s2. Pertanto, per misurare la reale accelerazione del dispositivo, il contributo della forza di gravità deve essere rimosso dai dati dell'accelerometro. Ciò può essere ottenuto applicando un high-pass filter. Al contrario, è possibile utilizzare un low-pass filter per isolare la forza di gravità.

L'esempio seguente mostra come è possibile farlo:

Immagine che contiene testo

Descrizione generata automaticamente

Gli accelerometri utilizzano il sistema di coordinate standard del sensore. In pratica, ciò significa che le seguenti condizioni si applicano quando un dispositivo è disteso su un tavolo nel suo orientamento naturale:

* se si spinge il dispositivo sul lato sinistro (quindi si sposta verso destra), il valore dell'accelerazione x è positivo.
* se spingi il dispositivo in basso (quindi si allontana da te), il valore di accelerazione y è positivo.
* Se si spinge il dispositivo verso l'alto con un'accelerazione di *A* m / s2, il valore dell'accelerazione *z* è pari a *A + 9,81*, che corrisponde all'accelerazione del dispositivo (+ A m / s2) meno la forza di gravità (- 9,81 m / s2).
* Il dispositivo fermo avrà un valore di accelerazione di +9,81, che corrisponde all'accelerazione del dispositivo (0 m / s2 meno la forza di gravità, che è -9,81 m / s2).

In generale, l'accelerometro è un buon sensore da utilizzare se si sta monitorando il movimento del dispositivo. Quasi ogni telefono e tablet Android ha un accelerometro e consuma circa 10 volte meno energia rispetto agli altri sensori di movimento. Uno svantaggio è che potrebbe essere necessario implementare low-pass filter e high-pass filter per eliminare le forze gravitazionali e ridurre il rumore.

**Giroscopio**

Il giroscopio misura il tasso o rotazione in radianti/secondi intorno all’asse x, y, z di un dispositivo.

Immagine che contiene screenshot

Descrizione generata automaticamente

L’accelerometro utilizzato per misurare l’accelerazione di un corpo lungo una direzione, per poter misurare l’accelerazione in uno spazio 3D occorre un accelerometro per ogni asse. Invece il giroscopio misura la rotazione di un corpo su di un proprio asse, pertanto è possibile ricavare i valori di velocità angolare. In uno spazio a tre dimensioni saranno necessari anche in questo caso tre sensori.

Il sistema di coordinate del sensore è lo stesso utilizzato per il sensore di accelerazione. La rotazione è positiva in senso antiorario; vale a dire, un osservatore che osserva da una posizione positiva sull'asse x, y o z un dispositivo posizionato sull'origine segnalerebbe una rotazione positiva se il dispositivo sembrava ruotare in senso antiorario. Questa è la definizione matematica standard di rotazione positiva e non è la stessa definizione del rotolo utilizzata dal sensore di orientamento.

Di solito, l'output del giroscopio è integrato nel tempo per calcolare una rotazione che descrive il cambiamento degli angoli nel tempo. Per esempio:

Immagine che contiene screenshot, testo

Descrizione generata automaticamente

I giroscopi standard forniscono dati di rotazione grezzi senza alcun filtro o correzione per rumore ed inclinazioni. In pratica, il rumore e l'inclinazione del giroscopio introdurranno errori che devono essere compensati. Di solito si determinano l'inclinazione e il rumore monitorando altri sensori, come il sensore di gravità o l'accelerometro.

**Giroscopio non calibrato**

Il giroscopio non calibrato è simile al giroscopio, tranne per il fatto che alla velocità di rotazione non viene applicata alcuna compensazione giroscopica.

La calibrazione di fabbrica e la compensazione della temperatura vengono ancora applicate alla velocità di rotazione. Il giroscopio non calibrato è utile per i dati di orientamento post-elaborazione e fusione.

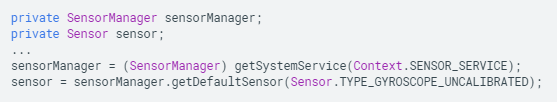
In generale, *gyroscope\_event.values [0]* sarà vicino a *uncalibrated\_gyroscope\_event.values [0] - uncalibrated\_gyroscope\_event.values [3].*

Questo è, calibrated\_x ~ = uncalibrated\_x - bias\_estimate\_x

I sensori non calibrati forniscono risultati più grezzi e possono includere alcuni errori, ma le loro misurazioni contengono meno salti dalle correzioni applicate attraverso la calibrazione. Alcune applicazioni potrebbero preferire questi risultati non calibrati come più fluidi e più affidabili. Ad esempio, se un'applicazione sta tentando di condurre la propria fusione del sensore, l'introduzione delle calibrazioni può effettivamente distorcere i risultati.

Oltre alle velocità di rotazione, il giroscopio non calibrato fornisce anche l'inclinazione stimata attorno a ciascun asse.

Il codice seguente mostra come ottenere un'istanza del giroscopio non calibrato predefinito:



1. **GPS**

Nelle telecomunicazioni il sistema di posizionamento GPS ( Global Positioning System ) è un sistema di posizionamento e navigazione satellitare militare statunitense. Attraverso una rete dedicata di satelliti artificiali in orbita, fornisce a un terminale mobile o ricevitore GPS informazioni sulle sue coordinate geografiche e sul suo orario in ogni condizione meteorologica, ovunque sulla Terra o nelle sue immediate vicinanze dove vi sia un contatto privo di ostacoli con almeno quattro satelliti del sistema. La localizzazione avviene tramite la trasmissione di un segnale radio da parte di ciascun satellite e l'elaborazione dei segnali ricevuti da parte del ricevitore.

Il sistema GPS è gestito dal governo degli Stati Uniti d'America ed è liberamente accessibile da chiunque sia dotato di un ricevitore GPS. Il suo grado attuale di accuratezza è dell'ordine di pochi metri, in dipendenza dalle condizioni meteorologiche, dalla disponibilità e dalla posizione dei satelliti rispetto al ricevitore, dalla qualità e dal tipo di ricevitore, dagli effetti di radiopropagazione del segnale radio in ionosfera e troposfera e dagli effetti della relatività.

***Il GPS su dispositivi mobili***

Con la diffusione dei sistemi GPS, ed il conseguente abbattimento dei costi dei ricevitori, molti produttori di telefoni cellulari/smartphone hanno cercato di inserire un modulo GPS all'interno dei loro prodotti, aprendosi quindi al nuovo mercato dei servizi (anche sul web) basati sul posizionamento (o LBS, location based services). Tuttavia, la relativa lentezza con cui un terminale GPS acquisisce la propria posizione al momento dell'accensione (in media, tra i 45 e i 90 secondi), dovuta alla necessità di ricercare i satelliti in vista, ed il conseguente notevole impegno di risorse hardware ed energetiche, ha frenato in un primo momento questo tipo di abbinamento.

Negli ultimi anni, però, è stato introdotto in questo tipo di telefoni il sistema GPS assistito, detto anche A-GPS dall'inglese Assisted GPS, con cui è possibile ovviare a tali problemi: si fanno pervenire al terminale GPS, attraverso la rete di telefonia mobile, le informazioni sui satelliti visibili dalla cella radio a cui l'utente è agganciato. In questo modo un telefono A-GPS può in pochi secondi ricavare la propria posizione iniziale, in quanto si assume che i satelliti in vista dalla cella siano gli stessi visibili dai terminali sotto la sua copertura radio.

***Location Data***

Una delle caratteristiche uniche delle applicazioni mobili è la consapevolezza della posizione.

Gli utenti di dispositivi mobili portano con sé i propri dispositivi ovunque e l'aggiunta della consapevolezza della posizione alla tua app offre agli utenti un'esperienza più contestuale.

I dati sulla posizione disponibili per un dispositivo Android includono *la posizione corrente del dispositivo* (individuata mediante una combinazione di tecnologie) , la *direzione* e il *metodo di movimento* e se il dispositivo si è spostato attraverso un confine geografico predefinito. A seconda delle esigenze dell'applicazione, è possibile scegliere tra diversi modi di lavorare con i dati sulla posizione:

* ***My Location Layer*** fornisce un modo semplice per visualizzare la posizione di un dispositivo sulla mappa. Non fornisce dati.
* ***Google Play services Location API*** è consigliata per tutte le richieste programmatiche per i dati sulla posizione.
* L'interfaccia ***LocationSource*** consente di fornire un provider di posizione personalizzato.

***Autorizzazioni di posizione***

Se l'applicazione deve accedere alla posizione dell'utente, si deve richiedere l'autorizzazione aggiungendo la relativa autorizzazione alla posizione Android all'app.

Android offre due autorizzazioni per la posizione: ACCESS\_COARSE\_LOCATION e ACCESS\_FINE\_LOCATION. L'autorizzazione scelta determina l'accuratezza della posizione restituita dall'API.

Immagine che contiene screenshot

Descrizione generata automaticamente

Si deve solo richiedere una delle autorizzazioni di localizzazione per Android, a seconda del livello di precisione di cui hai bisogno:

* ***android.permission.ACCESS\_COARSE\_LOCATION*** - Consente all'API di utilizzare WiFi o dati di celle mobili (o entrambi) per determinare la posizione del dispositivo. L'API restituisce la posizione con una precisione approssimativamente equivalente a un isolato.
* ***android.permission.ACCESS\_FINE\_LOCATION*** - Consente all'API di determinare una posizione quanto più precisa possibile dai provider di posizione disponibili, incluso il Global Positioning System (GPS), nonché i dati WiFi e delle celle mobili.

***My Location Layer***

Puoi utilizzare *My Location Layer* e il pulsante *My Location* per mostrare all' utente la sua posizione corrente sulla mappa, chiamando ***map.setMyLocationEnabled ()*** per abilitare *My Location Layer* sulla mappa.

Prima di abilitare *My Location Layer* , è necessario assicurarsi di disporre dell'autorizzazione di posizione di runtime richiesta.

L'esempio seguente mostra un semplice utilizzo di *My Location Layer* :

Immagine che contiene screenshot

Descrizione generata automaticamente

Immagine che contiene mappa, testo

Descrizione generata automaticamenteQuando *My Location Layer* è abilitato, il pulsante *My Location* appare nell'angolo in alto a destra della mappa. Quando un utente clicca sul pulsante, la telecamera centra la mappa sulla posizione corrente del dispositivo, se è nota. La posizione è indicata sulla mappa da un piccolo punto blu se il dispositivo è fermo o come un gallone se il dispositivo è in movimento.

È possibile impedire la visualizzazione del pulsante *My Location* chiamando ***UiSettings.setMyLocationButtonEnabled (false).***

L'applicazione può rispondere ai seguenti eventi:

* Se l'utente clicca sul pulsante *My Location* , l'app riceve una richiamata onMyLocationButtonClick () da GoogleMap.OnMyLocationButtonClickListener.
* Se l'utente clicca sul punto blu *My Location* , l'app riceve una richiamata onMyLocationClick () da GoogleMap.OnMyLocationClickListener.

***L'API di localizzazione dei servizi di Google Play***

L'API di localizzazione dei servizi di Google Play è il metodo preferito per aggiungere consapevolezza della posizione alla tua applicazione Android. Include funzionalità che ti consentono di:

* Determinare la posizione del dispositivo.
* Ascoltare i cambiamenti di posizione.
* Determinare la modalità di trasporto, se il dispositivo è in movimento.
* Creare e monitorare regioni geografiche predefinite, note come *geofence*.

Le API di localizzazione facilitano la creazione di applicazioni efficienti dal punto di vista energetico. Come l'SDK di Maps per Android, l'API di posizione è distribuita come parte dell'SDK dei servizi di Google Play.

***LocationSource***

Definisce un'interfaccia per fornire dati sulla posizione, in genere a un oggetto GoogleMap.

Un oggetto GoogleMap ha un provider di localizzazione incorporato per My Location Layer, ma può essere sostituito con un altro che implementa questa interfaccia.

Un oggetto GoogleMap attiva il proprio provider di posizione utilizzando ***activate(OnLocationChangedListener)*** .

Mentre è attivo (tra ***activate(OnLocationChangedListener)***  e ***deactivate()*** ), un fornitore di servizi di localizzazione deve inviare periodici aggiornamenti di posizione al listener registrato ***activate(OnLocationChangedListener)*** . È responsabilità del fornitore utilizzare saggiamente i servizi di localizzazione in base allo stato del ciclo di vita della mappa. Ad esempio, dovrebbe utilizzare solo servizi ad alta intensità di batteria (come il GPS) occasionalmente o solo mentre un'attività è in primo piano.

SITOGRAFIA

* <https://developer.android.com/guide/topics/sensors/sensors_motion#java>
* <https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale>
* <https://developers.google.com/maps/documentation/android-sdk/location>