

Cellular Connectivity and Noise Map

Relazione

Xia · Tian Cheng

Matricola: 0000975129

Email: tiancheng.xia@studio.unibo.it

Anno accademico

2022 — 2023

Corso di Laboratorio di applicazioni mobili
Alma Mater Studiorum · Università di Bologna

Indice

1	Introduzione	1
1.1	Feature implementate	1
2	Scelte progettuali	1
2.1	Informazioni generali	1
2.2	Mappa	1
2.2.1	Generazione cella	1
2.2.2	Generazione griglia	1
2.3	Raccolta dei dati	2
2.3.1	Struttura e memorizzazione delle misurazioni	2
2.3.2	Sampler	2
2.4	Servizi in background	3
2.5	Condivisione dati	3
3	Criticità e problemi noti	3

1 Introduzione

1.1 Feature implementate

2 Scelte progettuali

2.1 Informazioni generali

2.2 Mappa

Per la mappa è stato utilizzato *Google Maps* e l'implementazione è contenuta nel fragment `WaveHeatMapFragment`.

2.2.1 Generazione cella

Una cella della mappa rappresenta la misurazione di un'area quadrata¹ e la dimensione di quest'ultima scala automaticamente in base al livello dello zoom.

Una cella è descritta dalle coordinate del vertice superiore sinistro (nord-ovest) e a partire da questa vengono calcolate le coordinate degli altri convertendo la dimensione della cella (in metri) in un offset da applicare a latitudine e longitudine.

Per questioni estetiche, gli offset applicati alle coordinate sono approssimati in modo tale che tutte le righe siano allineate verticalmente (vedi figura 1).



Figura 1: Offset calcolati in maniera precisa (sinistra) e approssimato (destra)

2.2.2 Generazione griglia

La griglia è composta da celle generate relativamente ad una posizione di riferimento. In particolare, in fase di inizializzazione viene designata come cella di riferimento quella che pone la posizione dell'utente al centro e in base a questa è possibile determinare la posizione di tutte le altre celle della mappa.

Nello specifico, date delle coordinate (pos_{lat}, pos_{lon}), per determinare la cella che la contiene si calcola il numero di celle da saltare rispetto a quella di riferimento:

$$to_skip_tiles_{lat} = \lceil \frac{pos_{lat} - center_top_left_{lat}}{latitudeOffset(tile_length_meters)} \rceil$$

¹Esclusa la zona equatoriale, le celle appariranno rettangolari

$$\text{to_skip_tiles}_{\text{lon}} = \lfloor \frac{\text{pos}_{\text{lon}} - \text{center_top_left}_{\text{lon}}}{\text{longitudeOffset}(\text{tile_length_meters})} \rfloor$$

Le coordinate del vertice superiore sinistro sono quindi:

`tilelat = center_top_leftlat + (to_skip_tileslat · latitudeOffset(tile_length_meters))`

`tilelon = center_top_leftlon + (to_skip_tileslon · longitudeOffset(tile_length_meters))`

Con questo approccio, ogni volta che viene spostata la visuale della mappa, la griglia viene generata iterando a partire dalle coordinate dell'angolo nord-ovest visibile dello schermo, fino a raggiungere l'angolo sud-est.

In aggiunta, per maggiore efficienza, si tiene traccia delle celle generate in modo da evitare di ridisegnare una cella già presente. Questo meccanismo viene resettato quando viene cambiato il livello di zoom, in quanto tutte le celle già presenti diventano obsolete e vengono cancellate.

2.3 Raccolta dei dati

2.3.1 Struttura e memorizzazione delle misurazioni

Una misurazione è descritta dall'interfaccia `WaveMeasure` e contiene il valore della misurazione, un timestamp, la posizione e un flag per indicare se si tratta di una misurazione propria o ottenuta tramite condivisione. In aggiunta, è presente un campo per informazioni aggiuntive utile per distinguere alcune tipologie di misurazioni (es. per Wi-Fi e Bluetooth viene salvato il BSSID).

L'interfaccia `WaveMeasure` viene quindi utilizzata per implementare la classe `MeasureTable` che descrive la tabella del database dedicata per memorizzare le misurazioni. Tutte le misurazioni sono salvate nella stessa tabella e sono differenziate da un campo `type`.

2.3.2 Sampler

Per la raccolta dei dati è stato introdotto il concetto di *sampler* per gestire in maniera modulare le misurazioni. Nello specifico, un *sampler* è descritto dalla classe astratta `WaveSampler` e richiede l'implementazione dei seguenti metodi:

- `sample` per prendere una nuova misurazione
- `store` per il salvataggio dei dati nel database
- `retrieve` per la ricerca dei dati note le coordinate dei vertici di una cella della mappa

Inoltre, sono esposte le seguenti funzioni ausiliarie:

- `average` richiama `retrieve` e restituisce la media dei valori
- `sampleAndStore` richiama in sequenza `sample` e `store`

Per maggiore flessibilità, le misure vengono sempre intese come liste di `WaveMeasure`. Ciò permette di gestire misurazioni che per loro natura non generano un'unica misurazione (es. Wi-Fi e Bluetooth).

A partire da `WaveSampler` sono quindi implementati i *sampler* per Wi-Fi, Bluetooth, LTE e suono.

2.4 Servizi in background

2.5 Condivisione dati

3 Criticità e problemi noti