

TrashFinder

Progetto di Sviluppo di Applicazioni Mobili

Leonardo Pantani

7 Giugno 2022

1 Note

Il progetto è stato sviluppato su *Android Studio* ed è disponibile su **GitHub** [cliccando qui](#).

Per l'ottenimento dei dati iniziali sui cestini si è fatto uso della mappa open source *OpenStreetMap* (abbreviata **OSM**). I crediti e il link al sito sono visibili nelle impostazioni dell'app.

Per il sistema di clustering viene utilizzato il modulo *Marker clustering* della libreria **Android Maps Utils**, visibile su **GitHub** [qui](#).

Tutta la parte dell'app che prevede un collegamento alla rete (eccetto che per il download dei dati iniziali) non è stata implementata, pertanto alcune sezioni potrebbero non essere complete.

Durante la fase di test si è notato che l'*AVD* (**Android Virtual Device**) presenta dei problemi durante l'apertura della fotocamera, pertanto le sezioni dell'app che fanno uso del suddetto sistema dovranno (probabilmente) essere eseguite su un dispositivo fisico.

2 Importazione progetto

2.1 Da File

Tenere a portata di mano la cartella scompattata con il progetto all'interno. In seguito, seguire i passaggi: **New...** → **Import Project...** ed infine selezionare il progetto dalla finestra che si sarà aperta.

2.2 Da GitHub

Seguire i seguenti passaggi: **New...** → **Project from Version Control...** ed inserire l'URL: <https://github.com/LeonardoPantani/UNIPi-TrashFinder>

3 Spiegazione generale

TrashFinder è un'app che ha lo scopo di fornire all'utente un sistema di localizzazione di cestini. Questa app può essere utile, per esempio: in paesi in cui il numero di cestini per la spazzatura è basso (come il Giappone) o alle persone con difficoltà di movimento. Inoltre, permette di differenziare i tipi di cestini che vengono visualizzati, come quelli per le batterie esauste, medicinali scaduti e altro ancora.

4 Argomenti del corso implementati dall'app

- **Sistema di localizzazione** → traccia la posizione dell'utente in modo da mostrare i cestini più vicini intorno alla sua zona, l'utente potrà inoltre passare dalla visualizzazione mappa a bussola, la quale punterà al cestino selezionato in precedenza dall'utente;

- **Sensoristica** → vengono sfruttati i sensori all'interno del dispositivo mobile per simulare una bussola che punta il cestino selezionato;
- **Database** → l'utente può registrarsi opzionalmente per aggiungere i cestini ad un'area che non ne ha, o cambiarne alcune proprietà;
- **Fotocamera** → l'utente, per aggiungere un cestino, dovrà provarne l'esistenza aggiungendo una fotografia di quest'ultimo. In seguito, la proposta sarà valutata dalla comunità prima che venga aggiunto il segnalino sulla mappa.

5 Aspetti principali dell'app

- **Mappa principale** → costruita grazie alle librerie di Google Maps, questo elemento permette all'utente di vedere a colpo d'occhio i cestini intorno alla propria posizione. Visto l'elevato numero di punti di interesse, è stato implementato un sistema di clustering grazie al quale si riduce il carico sul thread UI e si rende la visione della mappa più piacevole. E' presente inoltre un tasto sulla ActionBar di filtraggio, per poter nascondere (o mostrare) le categorie che ci interessano.
- **Bussola** → frammento che si occupa di mostrare la direzione, rispetto all'orientamento del dispositivo in uso, del cestino precedentemente selezionato nella mappa. Per funzionare necessita della posizione attuale dell'utente e dei seguenti sensori: **Accelerometro**, **Campo Magnetico**.
- **Contributi** → uno degli aspetti principali di questa app è quello collaborativo, dove gli utenti possono aggiungere, modificare o rimuovere cestini dalla mappa qualora fosse necessario. Questa schermata, in particolare, mostra:
 - **Le tue richieste** → l'utente che ha effettuato l'accesso (dalla barra laterale) può contribuire aggiungendo, modificando o anche rimuovendo cestini sulla mappa. Le modifiche effettuate vengono tracciate e memorizzate in modo che gli altri utenti possano valutarle e approvarle. Premendo sul pulsante "Le tue richieste" si avrà una lista delle ultime inviate dall'utente attualmente collegato. E' possibile toccare su una qualsiasi richiesta per visualizzarne informazioni extra, oltre che a poter aprire il navigatore.
 - **Contributi dell'utente e globali** → queste schede mostrano all'utilizzatore quante richieste ha proposto e quante modifiche sono state proposte dall'intera comunità. E' presente anche un contatore dei cestini visibili sulla mappa dell'utente.
- **Editor Mappa** → l'utente che ha effettuato l'accesso può accedere a questo frammento, composto da una mappa e uno spazio dove compariranno dei campi relativi alla modifica di quest'ultima.
 - **Crea** → per entrare in questa modalità basta toccare l'area sottostante alla mappa (come consigliato dall'app stessa). L'area, inizialmente vuota, diventa un modulo da compilare a tutti gli effetti. Per selezionare la posizione del cestino bisogna spostare la mappa in modo da sovrapporre il mirino (che sarà comparso) con la posizione del cestino nel mondo reale. Per poter creare il cestino bisognerà compilare tutti i campi richiesti. E' necessario scattare una foto del cestino per provare alla community l'esistenza dello stesso.
 - **Modifica** → per entrare in questa modalità basta toccare un cestino qualsiasi sulla mappa. Entrati in modalità modifica, si può cambiare alcune proprietà del cestino. Fatto ciò che è necessario, toccare su "Salva" per terminare.

6 Struttura del progetto

Il codice sorgente del progetto è stato suddiviso in vari *packages* per permettere una migliore comprensione della sua struttura.

- **community** → contiene la parte relativa alla community (chiamata in seguito "Contributi"). Comprende la visualizzazione dei contributi dell'utente attuale e dell'intera comunità (non essendoci un database reale online, alcuni dati si possono considerare semplici *placeholders*). Include anche la parte della visualizzazione delle richieste inviate dall'utente.
- **compass** → gestisce la sezione della bussola. Notare che una parte del metodo (commentata) **onSensorChanged** nella classe *Compass.java* è stata ottenuta da dei progetti Open Source online.
- **data** → comprende la definizione degli oggetti **POIMarker** e **POIRequest**.
 - **marker** → rappresenta un marcatore di *TrashFinder*. Comprende, oltre che a latitudine e longitudine, dati come: **tipo di cestino** e **note**.
 - **requests** → rappresenta una richiesta / proposta di un utente. Include: l'elemento **POIMarker** aggiornato, la **data di invio**, il **link all'immagine** e l'**email dell'utente** che ha proposto la modifica.
- **feedback** → semplice attività che contiene un modulo per inviare una segnalazione allo sviluppatore dell'app. Non essendoci un sistema esterno a cui inviare le informazioni, si è preferito un semplice invio di e-mail.
- **intro** → adapter che gestisce l'introduzione da mostrare all'utente al primo avvio dell'app. E' importante, in particolare, perché fa capire a chi ha scaricato l'app il perché della richiesta del permesso alla posizione.
- **maps** → gestisce tutta la sezione relativa alla visualizzazione della mappa. Notare che la medesima mappa è presente in due frammenti: *Mappa* e *Editor Mappa*. Il codice di questi ultimi è ridondante in molte parti, ma durante la fase di sviluppo si è preferita questa scelta per velocizzare la scrittura del codice.
- **services** → l'unico servizio implementato è il **DownloadService**, che si occupa del download dalla rete dei dati iniziali sui cestini. Normalmente è lanciato al primo avvio dell'app, se non nel caso in cui si stia usando la rete mobile; in tal caso, sarà richiesto all'utente se desidera connettersi per ottenere i dati aggiornati. In caso di problemi è possibile comunque invalidare i dati sui cestini attuali dalla schermata delle *impostazioni*.
- **settings** → contiene la logica della schermata *impostazioni*, come la disattivazione / attivazione di alcune opzioni e della gestione degli **onClick** listeners dei crediti.