CSCI3310 Mobile Computing & Application Development

Assignment 3 (Fragments and Google Maps)
Flowery Campus

Trumpet Tree at Science Centre
captured by CPRO CUHK

Due : Apr 11, 2022 (Mon) 11:59pm

In different seasons, blossom among the CUHK campus has decorated the hill and the suburbs area with colorful charms and created many must-visit spots: in spring with the pink cherry blossoms in NA, in summer with the blue Jacaranda, and in autumn with the purple Orchid tree. This assignment needs you to write an **Android** App which can display building images and maps with **Java** and **Android Studio Arctic Fox**. The default testing simulator is **Pixel 4 Portrait** mode on **Android 10.0 (API 29)**. Name your package as `edu.cuhk.csci3310.flowerycampus`.

In this assignment, you need to **display a Google map view** showing **9 flowery locations** in the campus as markers when the program starts up. As a practice of various programming skills in this assignment, you need to use the `Fragment` to **display the corresponding blossom photo by clicking the marker** below the map view. The **photo will be removed on clicking any empty map area** other than the markers (or any default POIs on the map).

The **marker title** and **GPS** locations of the flowery locations, in latitude/longitude pair, can be obtained from an additional CSV file. Your task is to show users the locations as markers in a `MapFragment` through the use of Google Maps Android API. You also need to add/remove an additional photo, contained as another `Fragment`, according to different clicks on the map.

In this assignment, you are **NOT required** to handle the layout variation and state retain due to configuration change, such as screen rotation. You can assume the testing is always under Portrait mode. On the other hand, you can optionally implement this.

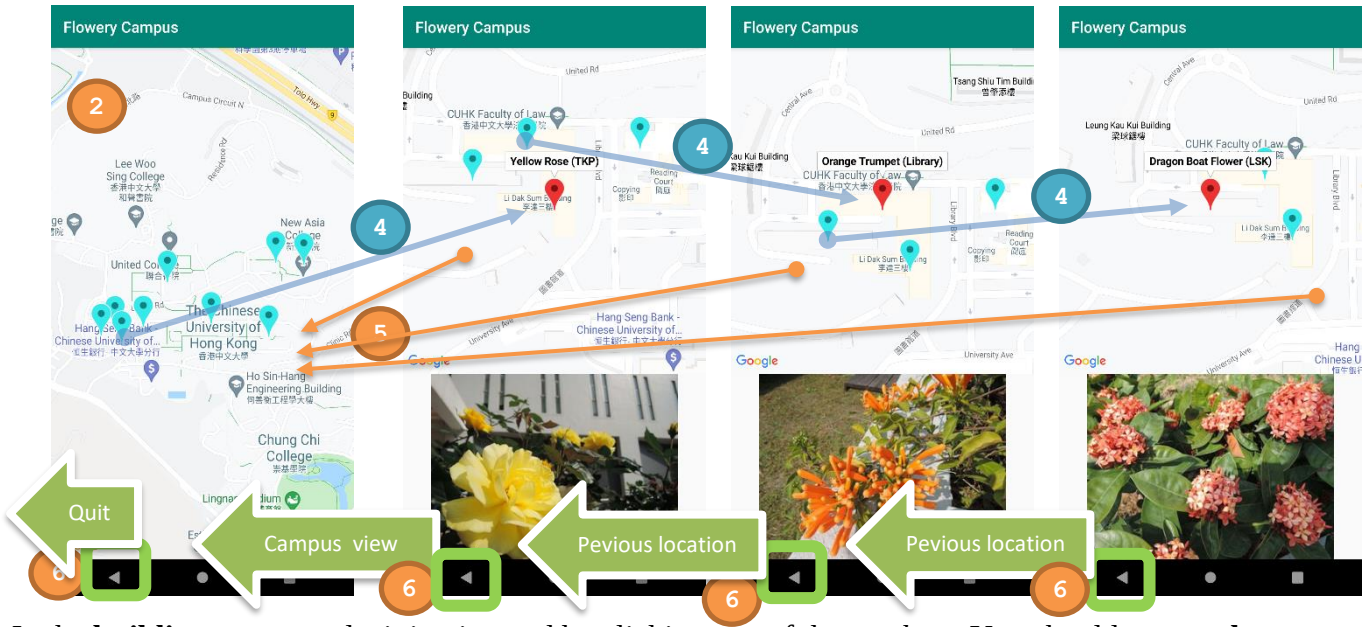
It is recommended to open a new project to try the Maps SDK for Android Quickstart tutorial to get familiar with it: https://developers.google.com/maps/documentation/android-sdk/start
Once you have the confidence in working on the map, you may proceed to do the following.

Requirements

1. Download a Starter Project "*Flowery Campus-Starter.zip*" to start your work where resources are included. In `/raw/`, text-based details of each flower are given in a comma-separated value (CSV) file, **`cu_flowers.csv`**. The location, i.e. the GPS latitude/longitude pair, with each flower's info and photo filename, can also be obtained from the CSV file.

2. You need to embed a map into your **`flowerycampus`** app. The map type should be normal, while the **initial stop** of your map view should be a campus overview showing all 9 flowery locations. For each flowery location, your program should **add a marker** with *HUE_CYAN in colour* and a *title of the flower's name and location* (which will be shown as a balloon text on top of the marker on selection by default).

3. Your map should support **TWO** zoom modes: the **campus zoom** mode and the **building zoom** mode, where they can be switched between by clicking on the marker or non-marker region on the map view.



4. In the **building zoom** mode, it is triggered by clicking one of the *markers*. You should **center the camera at the building location** with suitable zoom to **display the selected building** and some nearby buildings as well, say at a zoom level of 19. For *ALL* other locations, your program should show markers of default colour, i.e. *HUE_CYAN*; the selected location's marker should be shown in *HUE_RED*. In addition, the **flowery photo for the selected location should be shown below the map view**. It is recommended to contain the photo in an additional **`Fragment`** for easier navigation management.

5. In the **campus zoom** mode, this is where your mapView should enter initially, also triggered by clicking any "*non-marker region*" on the map while **resetting the selection**. Your program should set the zoom such that all locations can be visible at the greatest possible zoom level, say 16. You should compute the **bounds** based on all GPS locations and find its center.

6. On pressing the BACK button, it should **go back to the previously selected location with the corresponding photo shown** until going back to the campus zoom. Any click on the map other than the marker will bring the view back to campus zoom as well, such that all previous location navigation can be ignored. [NOTE: BackStack cannot handle all navigation states, such as those for the markers, you may need to build an extra stack to save/load navigation states.]

Submission

Pack your app folders and related files into an archive named "**3310_asg3.zip**". Do packaging via Android Studio's main menu: **File > Export > Export to Zip File…** (this would include necessary all /java, /res and /Gradle files), and include a readme if you have included additional dependency in your project.

Submit it into our assignment box in the Blackboard system before the deadline, **Apr 11, 2022, at 11:59 pm**. Late submissions will risk a score deduction of range between *10%* to *50%* if they are being done within 48 hours after the deadline. Submission later than **Apr 13, 2020,** won't be considered.

Grading Remarks:

1. Follow the project/package naming stated in the specification.
2. Put down personal information (Name and SID) in all Java (.java) code.
3. The code should be easy to read and contain comments to indicate computational logic.
4. The submitted code should be free of any typing mistakes, compilation errors/warnings.
5. The submitted app should be runnable at least on the virtual device stated in spec.
6. The program has to pass the corresponding test steps stated in the specification. In general, if you've followed the instructions above, you'll earn the vast majority of the points below.

    - **(30%) Basics** – does the app run correctly and properly styled in code?
    - **(20%) Fragment Display** – does all Views correctly contained in Fragments and rightfully transacted?
    - **(20%) Map Display** – are the map, markers and info displayed correctly?
    - **(30%) Navigation of location** – are pages flowed correctly with correct info via clicking or Back-pressing?

Extensions

There exist quite several extensions you may work within this assignment if you find it interesting:

1. Your program can restrict the user's panning to a given area. For example, in the *campus zoom* mode, the scroll can be limited to approximately 500 metres horizontally and vertically away from the bounds; in the *building zoom* mode, the scroll can be limited to approximately 50 metres horizontally and vertically away from the building's location.
2. Google Direction API provides a means to search for a path between two endpoints. You may enable your current location and find the route to reach a different building. Pay attention: Google Direction API is no longer free since Jun-2017.

References

1. Setting the bounds of the map – Android Developers
https://developers.google.com/maps/documentation/android-sdk/views#setting_boundaries

2. Restricting the user's panning to a given area – Android Developers
https://developers.google.com/maps/documentation/android-sdk/views#restricting_the_users_panning_to_a_given_area

3. Get lat/long given current point, distance and bearing – Stack Overflow
https://stackoverflow.com/questions/7222382/get-lat-long-given-current-point-distance-and-bearing