Week 15 (Module 11)

CS 5254

Maps in apps

- The two major providers are **Google Maps** and **OpenStreetMap** (via the osmdroid project)
    - The interfaces are nearly identical, from a developer's point of view
        - Google Maps requires a developer account, API key, and emulator support of Google Play
            - More sophisticated features are available, with a familiar look-and-feel for users
        - OpenStreetMap requires only a user agent string with the name of the app
            - Sufficient advanced features are available to support most typical usages
- Mapping implementations generally support the concept of a **MapView** component
    - Just include this component in a layout to display a map (similar to a **WebView** for web pages)
        - Usually a few configuration steps are needed, either in the layout or code
    - This component retains the state of the map, including the zoom level and center
        - However, the developer is responsible for storing/restoring this state as necessary
    - All the tiles necessary to depict the map are downloaded automatically from specified servers

Sharing data among fragments

- In Project 3, the map resides in its own fragment, but it must share data with other fragments
  - The Map fragment must access the list of `GalleryItem` objects fetched by the Gallery fragment
  - This data is (initially) stored in the Gallery fragment's ViewModel
- A fragment's ViewModel can only be accessed from that fragment, not from any other fragments
  - However, all fragments hosted by the same activity can access that activity's ViewModel
- We need to promote the Gallery fragment's ViewModel to become owned by the common activity
  - Instead of delegation `by viewModels()` use `by activityViewModels()` from each fragment
  - Note that the Map fragment still also needs its own ViewModel to store/restore map state
    - This state could be held in the shared ViewModel, but this isn't a good practice
      - Data used only by a single fragment should reside in that fragment's ViewModel
- Just to be clear, sharing data among fragments doesn't *only* apply to maps!
  - This is a common situation in any app with multiple fragments

Map markers

- **Markers** are one of many overlays that can be displayed on any map
  - As with the maps, marker interfaces are nearly identical, from a developer's point of view
- In the osmdroid library we're using for Project 3:
  - The marker minimally requires only a location (latitude and longitude)
    - We can also specify several options, including anchor, title, and a related object
  - Markers are drawn in the order in which they reside within the overlay list
    - If markers overlap, and one should appear above the others, move it to the end of the list
  - By default, clicking a marker displays its info window (closing any other info windows)
    - Clicking within an info window closes it
    - We are going to extend this behavior for Project 3
      - Clicking a marker while its info window is shown will navigate to the PhotoPageFragment
        - Note that the bottom navigation will still reflect the Map screen
        - Navigating Back will return to the Map screen, centered and zoomed as before

Hints and Tips: Project 3

- Nothing new; please just be sure to make backups along the way!