Week 9
(Module 7)

CS 5254

Menus in the app bar

- The BNRG approach has been deprecated and will soon be unavailable
  - This approach involved two separate lifecycle states just to create the menu
- The current approach is much cleaner, but still relatively new (Sep 2021)
  - Many of the examples you'll find online are obsolete now
- Menus can be managed by any on-screen host, typically an activity or fragment
    - The creating and handling functions are provided upon menu creation in most lifecycles
      - Typically in `onCreate()` for activities and `onCreateView()` for fragments
    - The menus are also aware of lifecycle owners for convenience
      - Menus can automatically follow appear/disappear with the presence/absence of its host
- Each **Menu** requires a menu resource which defines one or more **Item** objects
  - The menu items are the individual actions that can be taken be the user
- The main point of entry is the `addMenuProvider()` of the activity, with two parameters:
  - A `MenuProvider` object, which has two abstract functions that require implementation:
    - The `onCreateMenu()` function inflates the menu resource
    - The `onMenuItemSelected()` function provides a handler for the selection of any item by ID
  - An optional `LifecycleOwner` object, which determines the lifespan of the menu
    - Default is the activity, so this is usually specified as `viewLifecycleOwner` for a fragment

Implicit intents

- Recall from Module 4 that an Intent can be used to launch a new activity within a single app
- An **Implicit Intent** can be used to request services from *other* apps on the same device
  - The OS will look for apps that have registered to service actions via their manifest
  - The `Intent` class has several constants, such as `ACTION_VIEW`, `ACTION_DIAL`, and `ACTION_SEND`
    - An intent may include **extra** data, exactly as with explicit intents
- Creating an intent generally requires the action, a MIME type, and the location of any data
  - An `ACTION_CHOOSER` intent will ask the user to decide which app to handle the request
- Apps should query the **PackageManager** to determine whether any appropriate app is available
  - If the implicit activity can't be resolved, then no app is willing/able to handle the action
- Apps must disclose intent queries via the manifest, to request user permission for access
- A few specific notes:
  - Taking photos with implicit intents will be covered in Module 8
  - Working with contacts is an especially complex topic, outside the scope of our curriculum

Hints and Tips for Project 2C

- You'll be adding 8 new features in order to complete the DreamCatcher app
  - The total effort per feature – and the amount of guidance provided – varies widely
  - Many of these features diverge from BNRG, or don't follow it at all
  - Ideally you should feel at least somewhat comfortable working with your system by now
- Please try to manage your time over the next three weeks, and try to get started early if possible
- Be sure to regularly export your project to make backups along the way
  - It's recommended that you make at least one backup per completed feature