Week 10
(Module 8)

CS 5254

Swipe gestures

- This isn't in BNRG, but it's quite straightforward to implement for any RecyclerView
- The **ItemTouchHelper** object is the starting point for everything
  - Construct an ItemTouchHelper by passing in an **ItemTouchHelper.SimpleCallback** object
    - Requires two functions – `onMove()` and `onSwiped()` – as overrides
      - The `onSwiped()` function receives a ViewHolder representing the swiped item
    - Also override `getSwipeDirs()` to specify valid swipe directions for a given ViewHolder
  - Then just attach the ItemTouchHelper to the RecyclerView component

Requesting pictures from a camera app

- Note: Pictures from the camera are far too large to store anywhere but the device filesystem
- Each app has a dedicated folder ("sandbox") in the device filesystem to hold files
  - This folder is private, and normally only the owning app can read/write at this location
    - However, this is a problem, because the camera app can't access this folder
  - A **ContentProvider** allows an app to expose its sandbox to other apps in a controlled manner
    - A **FileProvider** is a simplified implementation of ContentProvider specifically for files
- Declare the FileProvider as a `<provider>` element in the AndroidManifest
  - Use `android:authorities="..."` to specify a globally-unique authority specifier
  - Use `android:exported="false"` to denote that shared resources aren't public to all apps
  - Use `android:grantUriPermissions="true"` to denote temporary access within an intent
  - Include a `<meta-data>` sub element to indicate which paths are being shared
    - Typically this is done via a raw XML resource file (usually `@xml/files`)
- To request a picture, call the `registerForActivityResult()` function
  - Specify `ActivityResultsContracts.TakePicture()` as the contract
  - Specify a callback with a Boolean parameter indicating whether the picture was taken or not
  - Call `launch()` with a URI provided from the FileProvider, given
    - The authority string, as declared in the manifest
    - A new **File** object, constructed with the path and the name of the file

Asking permission and detecting the camera

- Apps must declare that they may access the camera, so users can decide to install
  - In modern Android devices, the user is asked whether to allow these actions
  - The manifest must include a section requesting permission to use the camera:
  - Within the `<queries>` element, specify an `<intent>` containing an action:
    - In this case: `<action android:name="android.media.action.IMAGE_CAPTURE"/>`
- Detecting a camera is done by querying the PackageManager with an implicit intent
  - If no app is available to handle the intent, then the app can assume there is no camera
    - This isn't strictly correct, but the camera is basically useless if no apps offer to take a picture
  - BNRG Listing 17.12 specifies a null URI, but it must be non-null so use `Uri.EMPTY` instead

Displaying the picture

- There are two major problems involved with displaying a picture from the camera
  - Pixel-for-pixel, a picture from the camera app is many times larger than the entire screen
  - We must use a **Bitmap** object, which is uncompressed, for display purposes
    - The memory usage for a full-scale image would be completely unacceptable
- BNRG provides a utility function to construct a Bitmap object that will fit a specified area
  - Typically we just use the size of the ImageView upon layout, however...
    - ...sometimes the exact area is unknown until *after* the layout is completed
      - In these cases we can generally use the area of a close-fitting container instead
- Once a properly scaled Bitmap is available, just call `setImageBitmap()` on the ImageView
- Note: This process isn't required to display the deferred/fulfilled icons in the RecyclerView
  - Those images are v24 drawable resources, so the build process converts and scales them

Hints and Tips for Project 2C

- As noted in Module 7, please regularly export your project to **make backups** along the way
  - This is the single biggest source of problems encountered in three-week academic assignments
  - Please make **at least** one backup per completed feature