Android Notes 02/07/2019

Responsiveness and Threads

* It is important that your app “feels” fast to the user
* Most phones/computers these days have several processors – so use them!
  + Multithreading and such
* Never block the main thread – if this is blocked for more than 5 seconds you’ll get an App Not Responding error (ANR)
* Main Thread Events:
  + onCreate, onResume calls
  + when UI views are drawn
  + when user touch events are handled
  + callbacks from networking and other services
* Message Queue
  + Holds messages from user in the order they are received – this is a large queue but it can over fill (this can be dangerous since it’s a circular queue)
  + Every time you touch the screen in anyway or start moving your phone around there are messages being sent from all sorts of sensors
  + Every thread has its own personal message queue – very handy for keeping a threads operations separate from the main thread (good for organization and keeping main moving along quickly)
* Handlers
  + A handler allows you to send and process Message and Runnable objects associated with a thread’s message queue
  + Two main issues:
    - To schedule messages and runnables to be executed at some future point
    - To enqueue an action to be performed on a different thread than your own
* Message Queue:
  + You can directly create your own messages
    - You can use these to communicate between threads (tell a thread that, “hey! This other thing I defined happened, do this thing I defined!”)
* Runnables:
  + You can directly add your own Runnables
  + A thread essentially
* AsyncTasks:
  + Provides some handy tools
    - onPreExecute
    - doInBackground()
    - onProgressUpdate()
    - onPostExecute()
  + don’t use this for long running threads
  + doInBackground:
    - do the time-consuming stuff here (like calculations and such)
    - should be one single pass to do anything you need to do here
    - then return the results back to the main thread so that it can be output on the screen
  + onPostExecute:
    - update the UI and other things as needed – this call will close out the activity of this particular thread