Notepad Extra Credit

In this exercise, you will use the debugger to look at the work you did in Exercise 3. This exercise demonstrates:

- How to set breakpoints to observe execution
- How to run your application in debug mode

Step 1

Using the working Notepadv3, put breakpoints in the code at the beginning of the `onCreate()`, `onPause()`, `onSaveInstanceState()` and `onResume()` methods in the `NoteEdit` class (if you are not familiar with Eclipse, just right click in the narrow grey border on the left of the edit window at the line you want a breakpoint, and select 'Toggle Breakpoint', you should see a blue dot appear).

Step 2

Now start the notepad demo in debug mode:

a. Right click on the `Notepadv3` project and from the Debug menu select 'Debug As -> Android Application'.

b. The Android emulator should say "waiting for debugger to connect" briefly and then run the application.

c. If it gets stuck on the waiting... screen, quit the emulator and Eclipse, from the command line do an `adb kill-server`, and then restart Eclipse and try again.

Step 3

When you edit or create a new note you should see the breakpoints getting hit and the execution stopping.

Step 4

Hit the Resume button to let execution continue (yellow rectangle with a green triangle to its right in the Eclipse toolbars near the top).

Step 5

Experiment a bit with the confirm and back buttons, and try pressing Home and making other mode changes. Watch what life-cycle events are generated and when.

The Android Eclipse plugin not only offers excellent debugging support for your application development, but also superb profiling support. You can also try using `Traceview` to profile your application. If your application is running too slow, this can help you find the bottlenecks and fix them.