Stage 5.2 - Notifications Example

Scene 1: Screencast

==Now we're going to create an iOS app that uses notifications to drive a simple user interfacl.

We'll start with an existing storyboard containing a view with two buttons and a label

Two buttons ("red" and "blue") will drive changes to the label, but not in the typical manner. In order to demonstrate more usage of notifications, the UI elements will communicate by observing and posting notifications.

All the code for this application will live in the ViewController.m module, containing the viewController for our interface view with the buttons and the label.

// add first
First we'll define two constants to store our notification identification strings: RED_NOTIF & BLUE_NOTIF As with dictionary keys, in order to avoid bugs associated with typos, it's good practice to use either #defines, or constants for notification names.

There are existing IBActions connected to the red and blue buttons in the view and an IBOutlet connected to the UILabel.

// add second
In -viewDidLoad, we'll create two observer blocks: one for the RED_NOTIF and one for BLUE_NOTIF. Each will set the text property of notifLabel to indicate which notification was received.

// add third
In the -blueButton and -redButton IBAction methods, we'll add calls to -postNotification for BLUE_NOTIF and RED_NOTIF.

Running the app, we see the text label change in response to button presses.

// remove?
Interestingly, looking at the log messages, we see that the red and blue notifications are also observed by the block we added to the app delegate. This illustrates that any number of observers may be notified as the result of a single notification posting.