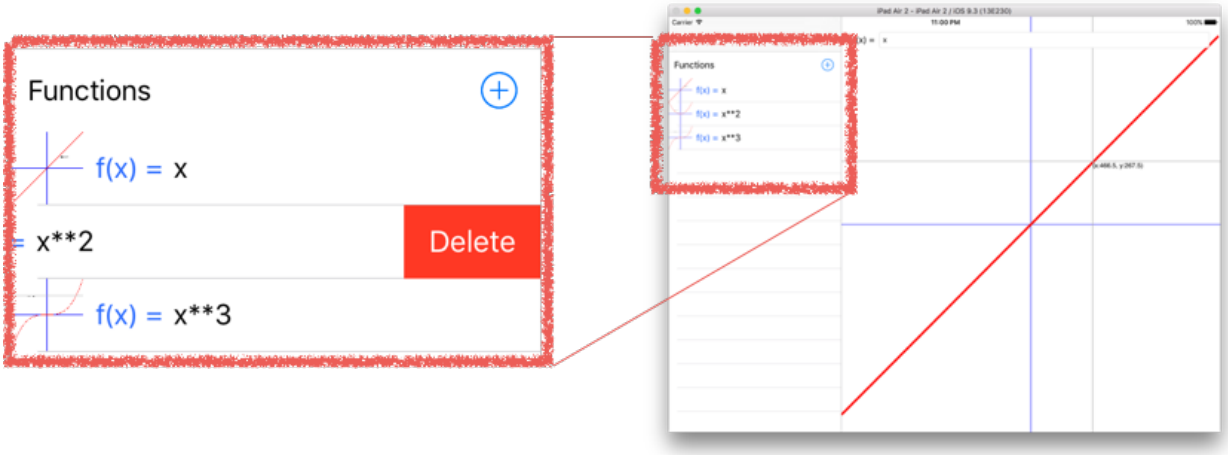
CS2048—Homework & Reading List 3

Reading

Read the following sections from the Cocoa Core Competencies guide: *NSNotification*
In the View Controller Programming Guide for iOS: *Using Segues*

Programming Assignment (due Oct 2nd)

In this assignment you will add extra functionality to the split view graphing calculator we wrote in class.



TODO

TODO1: Swipe to delete button *[20 points]*

In this lecture you saw how to incorporate a `UITableViewController` and the `UITableView` into an app. The `UITableView` is a very powerful and customizable view with two delegates. One nice feature that comes with the class is the ability to swipe to delete a row in the table. Check out the documentation for the *UITableViewDataSource* and implement the necessary method for swipe to delete.

TODO2: Button to add a new equation *[40 points]*

One way to add the "+" button in the figure above is to create a custom `UIView` for the `UITableViewHeader`. In order to do this you will override the function

```
override func tableView(tableView: UITableView,
                        viewForHeaderInSection section: Int)-> UIView?
```

in your table view controller. You can create the view for the header in code by creating a UIView that contains a UILabel and a UIButton. Remember that each view you create must have it's frame set, and must be added to its superview by calling the method `addSubview` on the superview

```
superView.addSubview(subView)
```

In order to figure out the height of the header (the width can be obtained from the frame of the UITableView) you can use (or override) the method

```
func tableView(tableView: UITableView, heightForHeaderInSection section: Int)
-> CGFloat
```

In all our lectures we connected buttons to actions in the storyboard, here you will be creating a view in code so you will not be able to graphically make the connection. Instead you will make the connection with the method addTarget from UIButton:

```
button.addTarget(self,
                 action: #selector(addNewFunctionPressed),
```

```
                    forControlEvents: .TouchUpInside)
```

TODO3: Add small thumbnail with function preview *[40 points]*

Method 1: By the end of last lecture we briefly saw how you could create custom table view cell classes and edit them in the storyboard. Add a `UIImageView` to the table view cell and connect it to your custom class. Write a method in your plot view controller that renders the view to an image every time the user interacts with the plotting view. The snippet below should help you get started

```
UIGraphicsBeginImageContextWithOptions(plottingView.frame.size, true, 0.0)
plottingView.drawViewHierarchyInRect(plottingView.frame,
                                     afterScreenUpdates: true)
let thumbnail = UIGraphicsGetImageFromCurrentImageContext()
UIGraphicsEndImageContext()
```

Make sure to fire a notification anytime the thumbnail is updated so that the table view has the most recent rendering of the function.

Method 2: Add a function plotting view to the custom `UITableViewCell` we created in class, and set the cell as it's delegate.

What to Hand In

Submit, via CMS, the entire project directory. If you project is called `MyProject`, Xcode will create a directory called `MyProject`, with a file called `MyProject.xcodeproj` and a subdirectory `MyProject`, please submit the **top level** `MyProject`.

Honesty and Integrity Policy

Projects are to be done individually. You may collaborate on the whiteboard, but each student's code must be their own.