

# Project 4

By: Eric Dockery

## 1) Chapter 7 Silver Challenge

### **Problem:**

Add pinch-to-zoom to the Hypnosister project from chapter 5. The first step is to give the scroll view a delegate:

\*EADAppDelegate should conform to the UIScrollViewDelegate protocol.

\* In application:didFinishLaunchingWithOptions:, set the scroll view's delegate property.

To perform as the scroll view's delegate, EADAppDelegate will need a property that points to the instance of EADHypnosisView. Add this property in a class extension in EADAppDelegate.m and update the rest of the code to use the property instead of the EADHypnosisView local variable.

To set up the scroll view, you will need to give it one EADHypnosisView as a subview then turn off the paging. The scroll view also needs limits on how much it can zoom in and out. Find the relevant UIScrollView properties to set in this class's reference page in the documentation.

Finally, you will need to implement the scroll view delegate method viewForZoomingInScrollView: to return the EADHypnosisView.

If you get stuck, visit the reference pages for the UIScrollView class and for the UIScrollViewDelegate protocol.

To simulate two fingers in the simulator to test your zooming, hold down the Option key while using the mouse.

### **Solution:**

For this problem you needed to remove the second screen, which was deleting about 5 lines of code from the EADAppDelegate. Then add some lines of code from the documentation of scrolling from the

[https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/UIScrollView\\_pg/ZoomZoom/ZoomZoom.html](https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/UIScrollView_pg/ZoomZoom/ZoomZoom.html) I did not include the Ensuring that Zoomed Content is Sharp when Zoomed

### **Code:**

```
///# inside the - (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions{  
    // scroll details  
    scrollView.minimumZoomScale=0.5;  
    scrollView.maximumZoomScale=4.0;  
    scrollView.contentSize=CGSizeMake(1280, 960);  
    scrollView.delegate=self;
```

### **Separate**

```
- (CGRect)zoomRectForScrollView:(UIScrollView *)scrollView  
withScale:(float)scale withCenter:(CGPoint)center {
```

```
    CGRect zoomRect;
```

```
    // The zoom rect is in the content view's coordinates.
```

```
// At a zoom scale of 1.0, it would be the size of the
// imageScrollView's bounds.
// As the zoom scale decreases, so more content is visible,
// the size of the rect grows.
zoomRect.size.height = scrollView.frame.size.height / scale;
zoomRect.size.width = scrollView.frame.size.width / scale;

// choose an origin so as to get the right center.
zoomRect.origin.x = center.x - (zoomRect.size.width / 2.0);
zoomRect.origin.y = center.y - (zoomRect.size.height / 2.0);

return zoomRect;
}
```

## Screenshots:

### Zoom in:



**Zoom out:**



## ***2) Chapter 8 Bronze and Silver Challenge***

### **Problem:**

Bronze:

Have the UITableView display two sections – one for items worth more than \$50 and one for the rest. Before you start this challenge, copy the folder containing the project and all of its source files in the finder. Then tackle the challenge in the copied project; you will need the original to build on in the coming chapters.

Silver:

Make it so the last row of the UITableView always has the text No more items!. Make sure this row appears regardless of the number of items in the store (including 0 items).

### **Solution:**

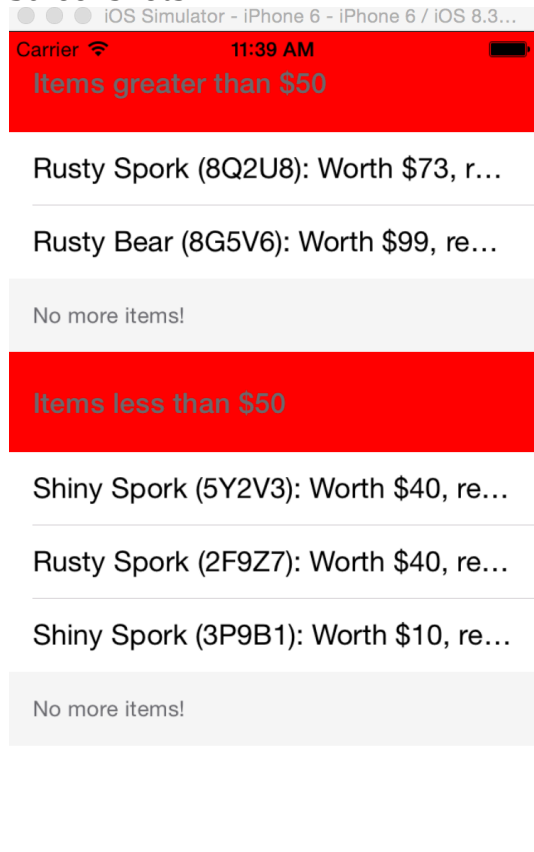
Bronze:

For this solution the easiest implementation is to add two sections to the table using two NSMutableArray for each section. When adding the items to the sections you will use a simple for loop with a if, else if statement to determine the placement of the items. For the top portion of the challenge to label the differences between the two sections I added a header that listed @"items greater than \$50" and @"items less than \$50."

Silver:

Since I had already incorporated a header for my two views of less than or more than \$50 I added a footer that said @”No more items!”

**Screenshots:**



### **3) Chapter 9 Silver Challenge**

**Problem:**

Make it so the table view always shows a final row that says No more items! (this part is the same as a challenge from the last chapter. If you have already done it, great!) Then make it so that this row cannot be moved.

**Solution:**

Previously solved on the last challenge problem. It can not be moved because it is a footer object.

**Screenshots:**

