Arrays

Array Concepts

Kind of like a spreadsheet with only 1 column

"Unlimited" rows

Numbered starting at 0 (zero)

Can hold any object

(even different objects)

NSArray

Traditional Objective-C (i.e. wordy)

```
NSArray *myArray = [NSArray arrayWithObjects:
   @“Vanilla",@"Chocolate",
   @"Strawberry",@"Rocky Road", nil];
NSString *myFlavorString = [myArray objectAtIndex:0];
```

Modern Objective-C (i.e. slightly less wordy)

```
NSArray *myArray = @[@“Vanilla”,@"Chocolate",
      @"Strawberry",@"Rocky Road"];
NSString *myFlavorString = myArray[0];
```

Initialized with another array, one or more objects

Cannot be changed once it’s created :-/

NSMutableArray

Pretty much the same as NSArray

```
NSMutableArray *myArray = [NSMutableArray arrayWithObjects:
    @“Vanilla",@"Chocolate",
    @"Strawberry",@"Rocky Road", nil];
NSString *myFlavorString = [myArray objectAtIndex:0];
```

Initialized with another array, one or more objects,

AND with capacity

More importantly can be changed (Mutable):

```
[myMutableArray addObject:@“Chocolate Peanut Butter"];
[myMutableArray insertObject:@"Superman" atIndex:2];
[myMutableArray removeObjectAtIndex:1];
[myMutableArray removeLastObject];
[myMutableArray removeAllObjects];
[myMutableArray sortUsingSelector:
    @selector(localizedCaseInsensitiveCompare:)];
```

More Code Basics

Comparison

Comparison	Description	Example
==	Equivalent (equal)	x == y
>	Greater than	x > y
>=	Greater than or equal to	x >= y
<	Less than	x < y
<=	Less than or equal to	x <= y
!=	Not equal to	x != y
!	Not (inverse true/false)	!(x = y)
isEqualToString	String comparison	[x isEqualToString: y]
isEqualToDate	Date comparison	[x isEqualToDate: y]
compare	Date comparison	[x compare:y] == NSOrderedAscending
&&	And	x == y && a == b
||	Or	x == y || x == z

Very Common Mistakes

== is **NOT** the same as =

== is comparison, = is assign

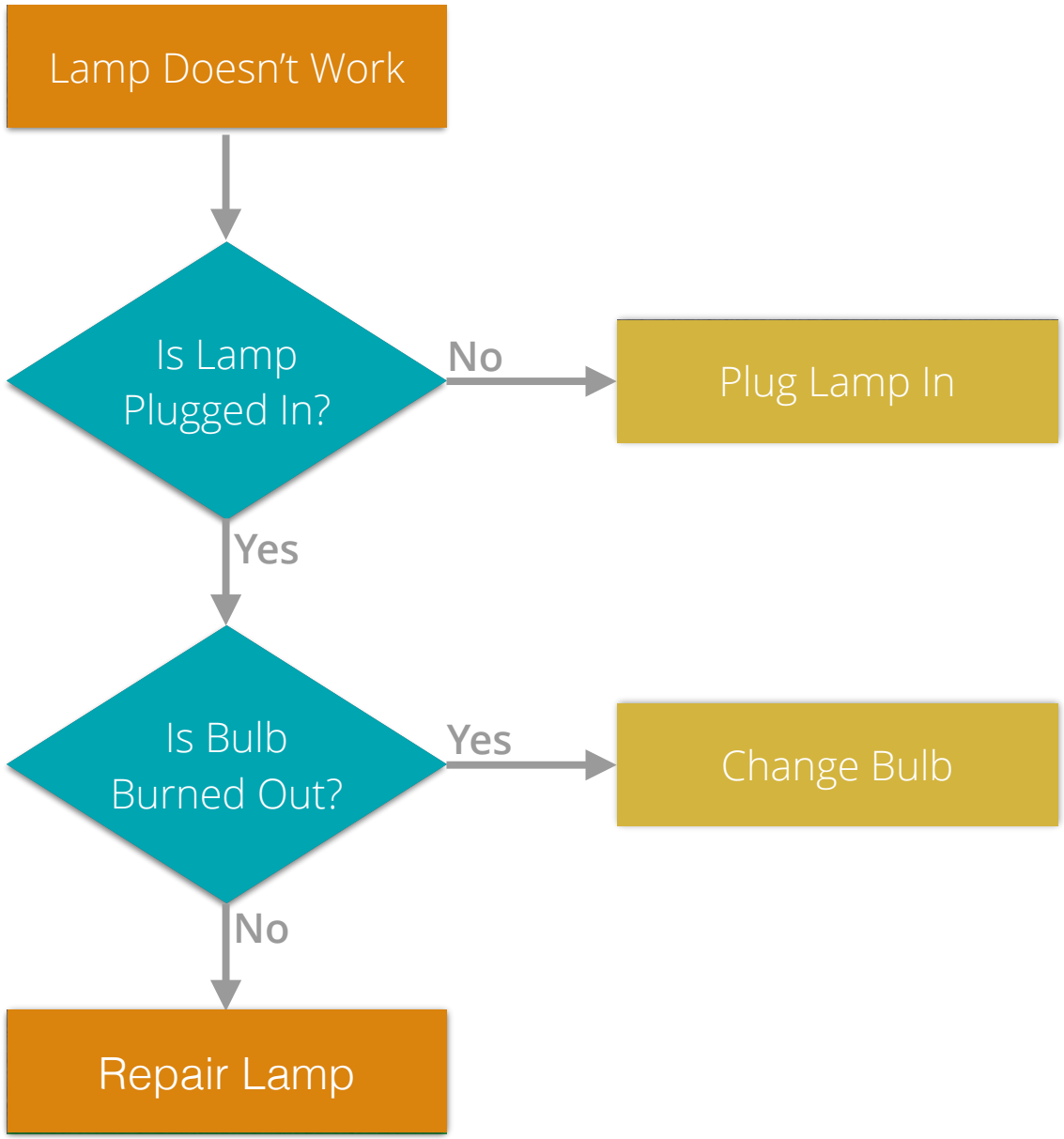
@"text" == @"text" will not evaluate correctly

use isEqualToString instead

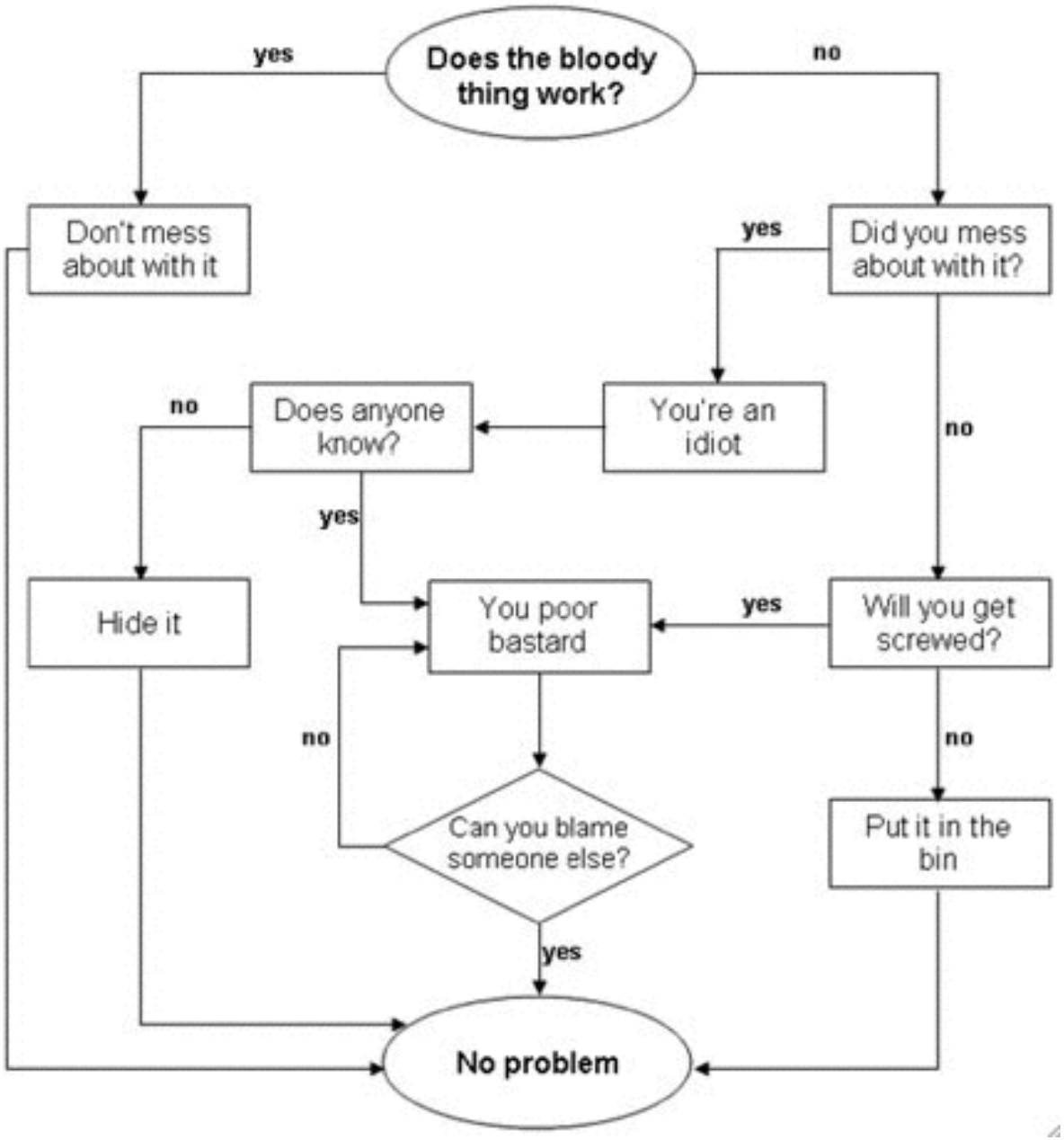
xDate == yDate will not evaluate correctly

use isEqualToDate instead

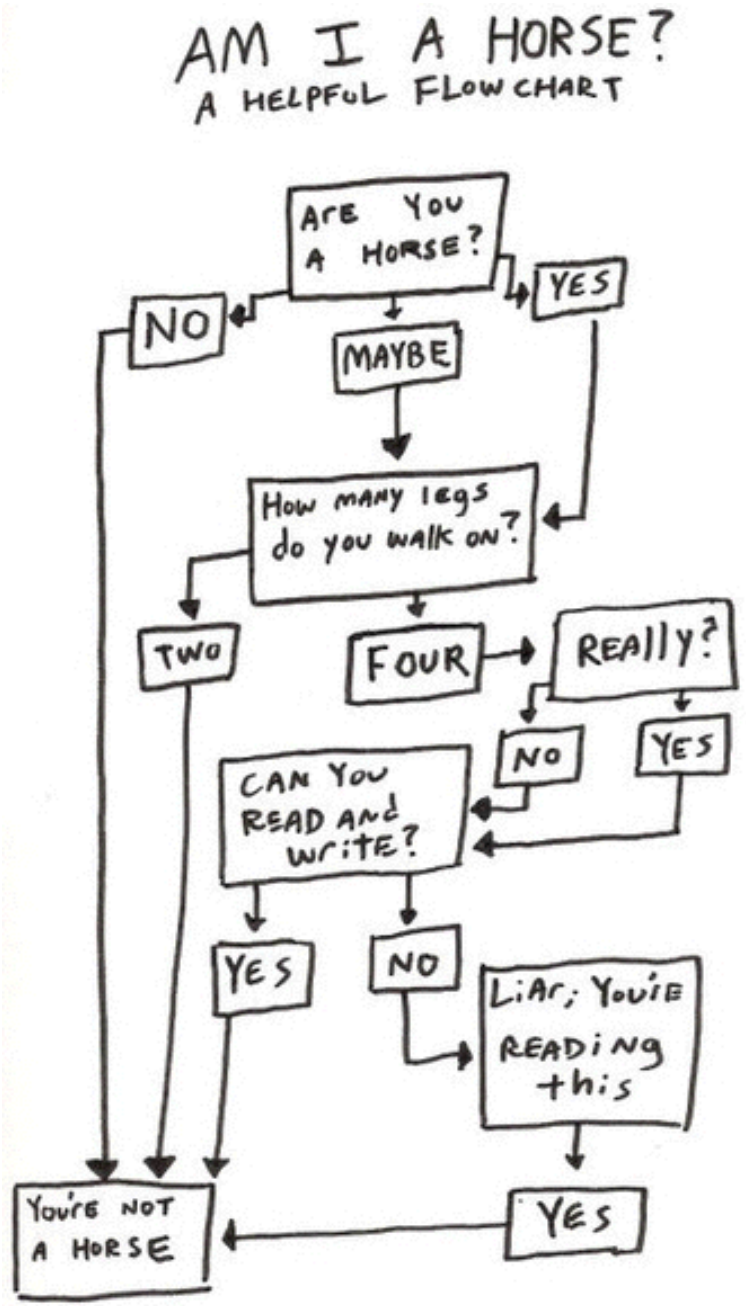
Conditional Statements



Conditional Statements



Conditional Statements



Source: Mental Floss

Conditional Statements

If-Then-Else

```
if (myInt == 1) {
    NSLog(@"Got 1");
} else {
    NSLog(@"Didn't Get 1");
}
```

Short If-Then-Else

```
if (myInt == 1) NSLog(@"Got 1"); else NSLog(@"Didn't get 1”);
```

Shorter If-Then-Else

```
myInt == 1 ? NSLog(@"Got 1") : NSLog(@"Didn't get 1”);
```

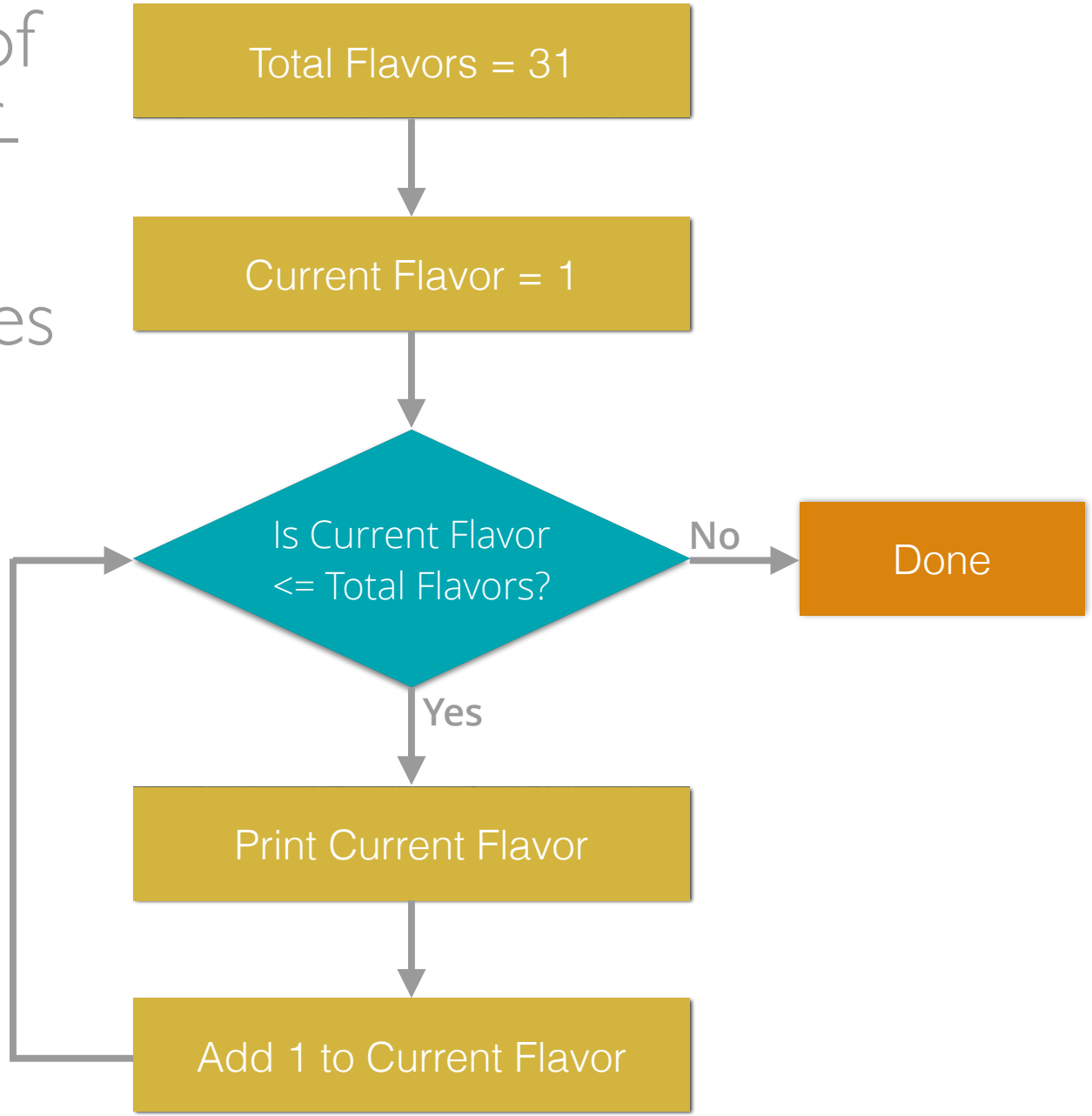
Conditional Statements

Case

```
switch (myInt) {
    case 1: {
        NSLog(@"Got 1");
        break;
    }
    default: {
        NSLog(@"Didn't Get 1");
        break;
    }
}
```

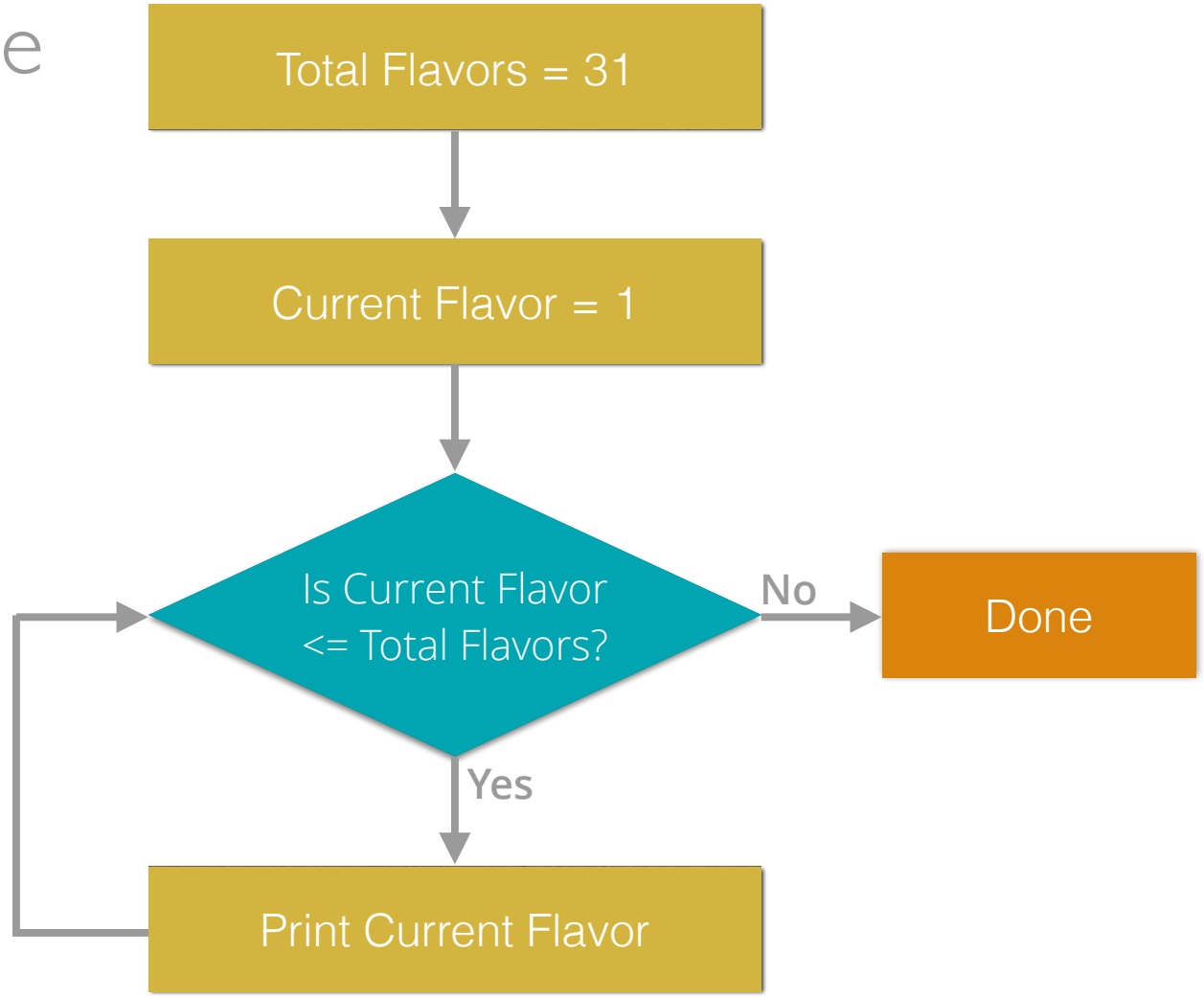
Loops

Loops are sort of like specialized If-Statements that repeat themselves

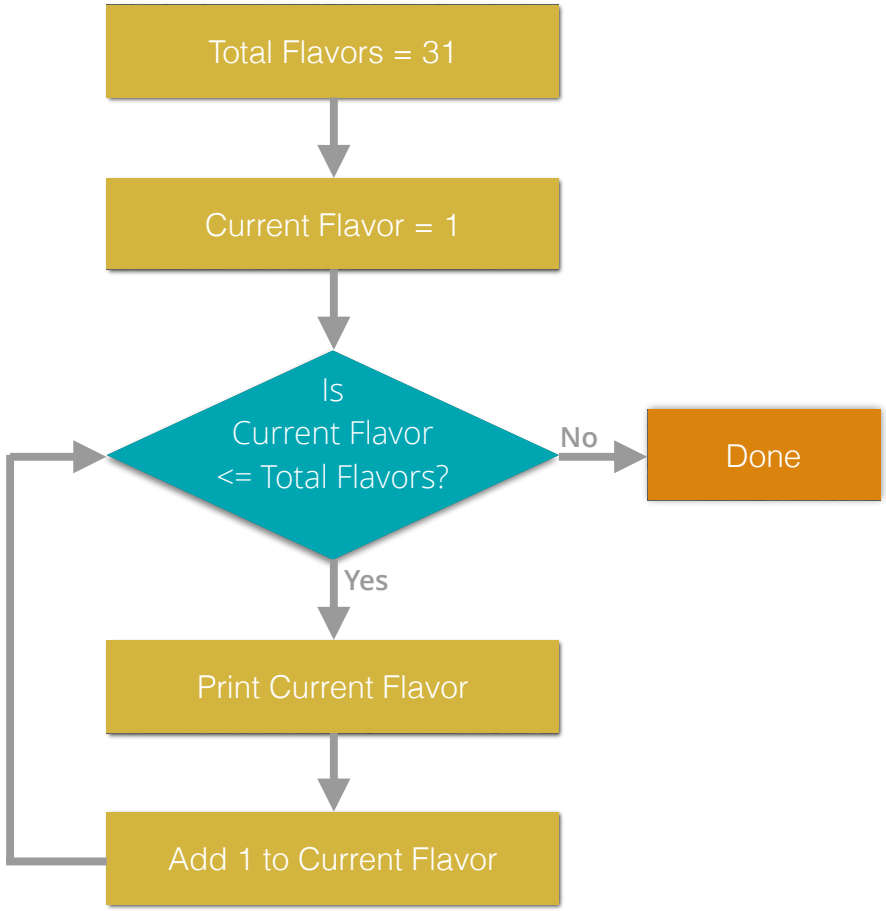


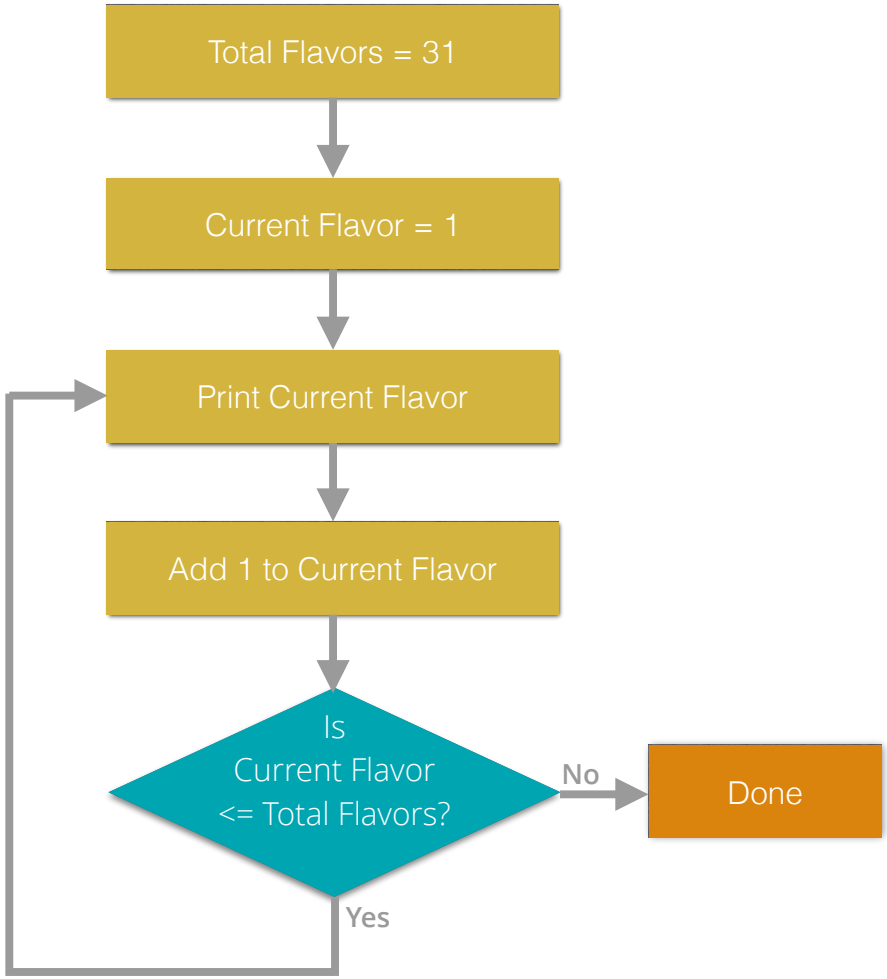
Loops

Beware of infinite loops



Loop Testing Location





Loops can have different locations for testing

Loops

While

```
int i = 0;
while (i < [myArray count]) {
    NSLog(@"Flavor: %@",myArray[i]);
    i++;
}
```

Do…While

```
int i = 0;
do {
    NSLog(@"Flavor: %@",myArray[i]);
    i++;
} while (i < [myArray count]);
```

Loops

For

```
for (int i = 0; i < [myArray count]; i++) {
    NSLog(@"Flavor: %@",myArray[i]);
}
```

Fast Enumeration For

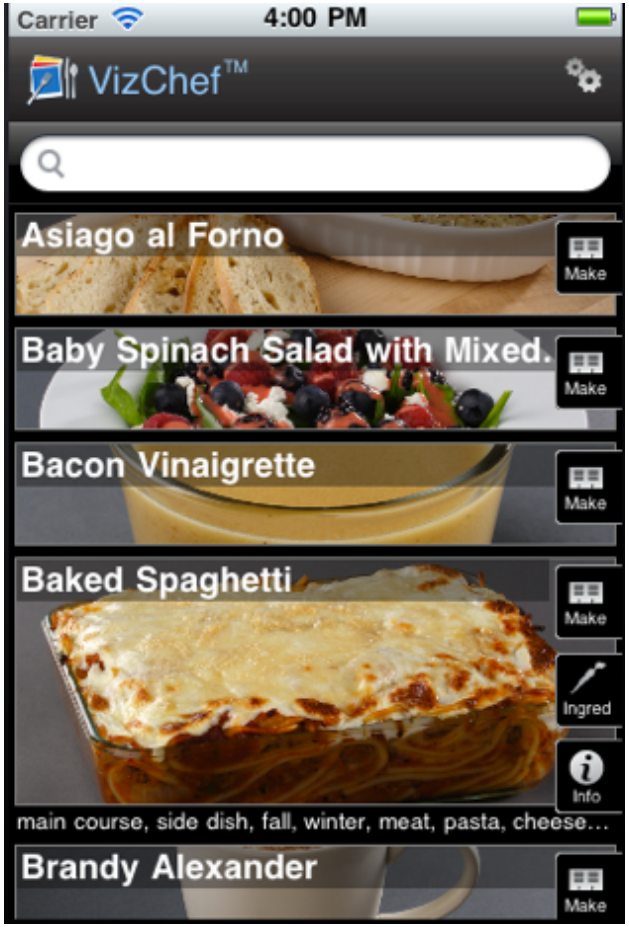
```
for (NSString *flavorString in myArray) {
    NSLog(@"Flavor: %@",flavorString);
}
```

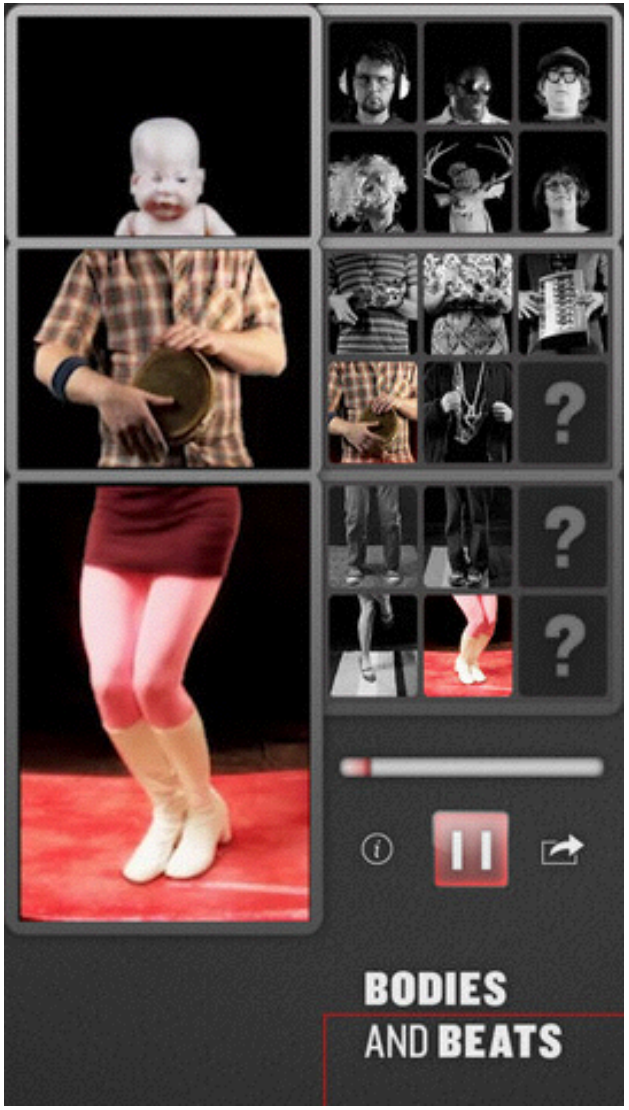
Images

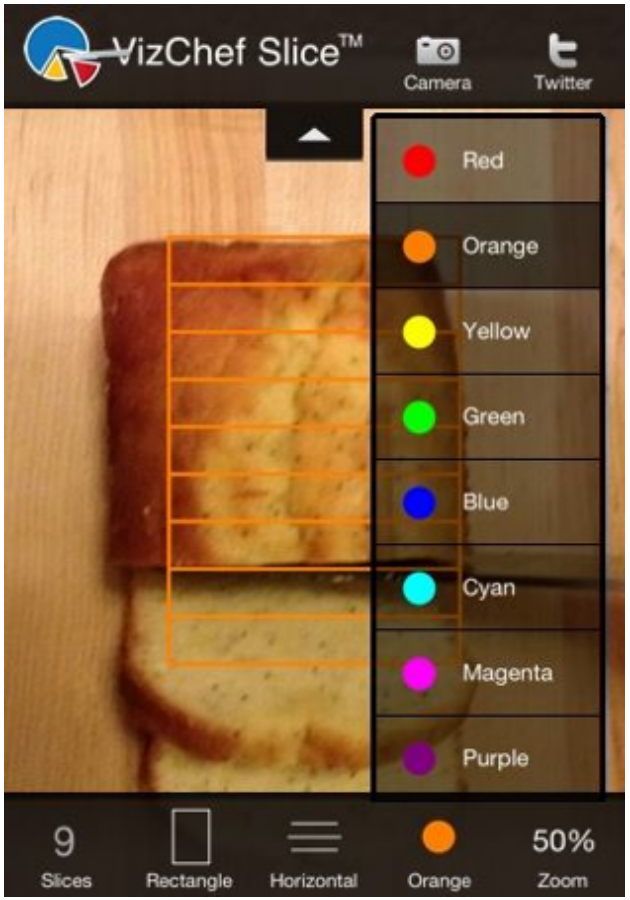
UIImage vs. UIImageView

- UIImage
  - An object that holds image data
- UIImageView can
  - Display one UIImage at a time
  - Can animate a series of images like a flip book
  - Can stretch images (with or without insets)
  - Can scale (zoom), pan, or rotate a UIImage
    (in code or with user interaction)

Pick the Images







UIImageViews

Used to display images such as photos, duh

Also used to display interface elements

(i.e. borders, backgrounds, etc)

Typically use .PNG (preferred) or .JPG

.PNGs can have transparency (use sparingly)

Can resize, but design for the size you need

Retina Display

Old iPhones are 320x480 pixels

Retina has double! (640x960)

We still lay controls out on 320x480 (or 568)

Apple automatically doubles the numbers

For best results, we provide retina & non-retina images with different names

myimage.png

myimage**@2x**.png

Naming Conventions

Type	Example
Non-retina	myimage.png
Retina	myimage@2x.png
Retina HD	myimage@3x.png
Tall Launch Image	myimage-568h@2x.png
iPhone	myimage~iphone.png
iPhone Retina	myimage@2x~iphone.png
iPad	myimage~ipad.png
iPad Retina	myimage@2x~ipad.png

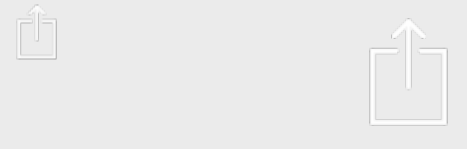
Still only one line of code!

```
[_myImageView setImage:[UIImage imageNamed:@“myimage.png"]];
```

Tip: use all lower case if there’s a chance of going cross platform

Retina

Non-Retina Retina Retina HD



Stretchable Images

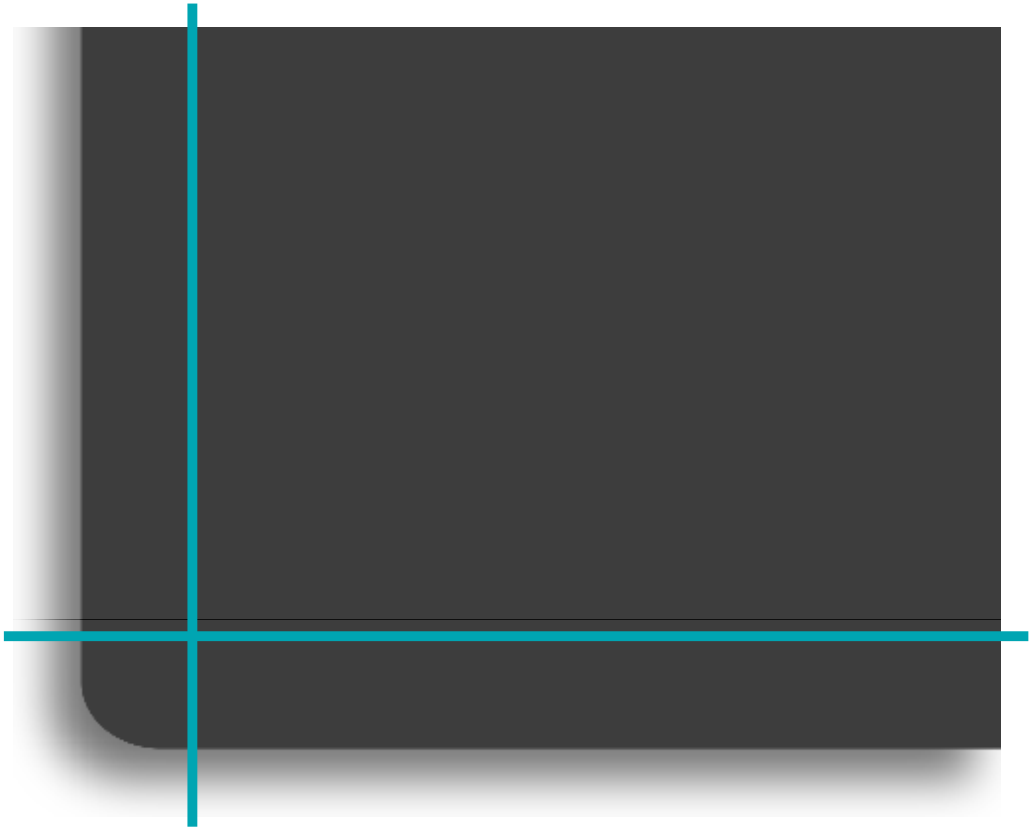
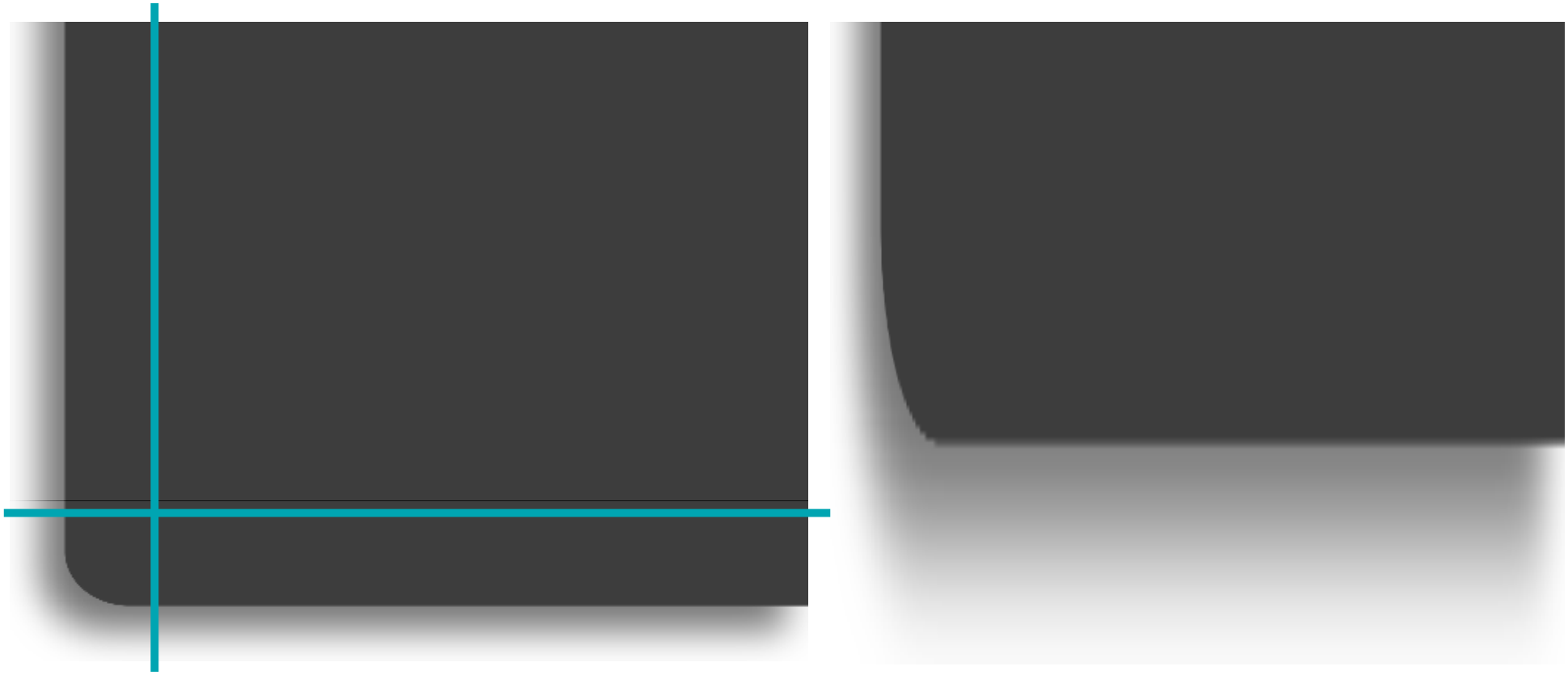


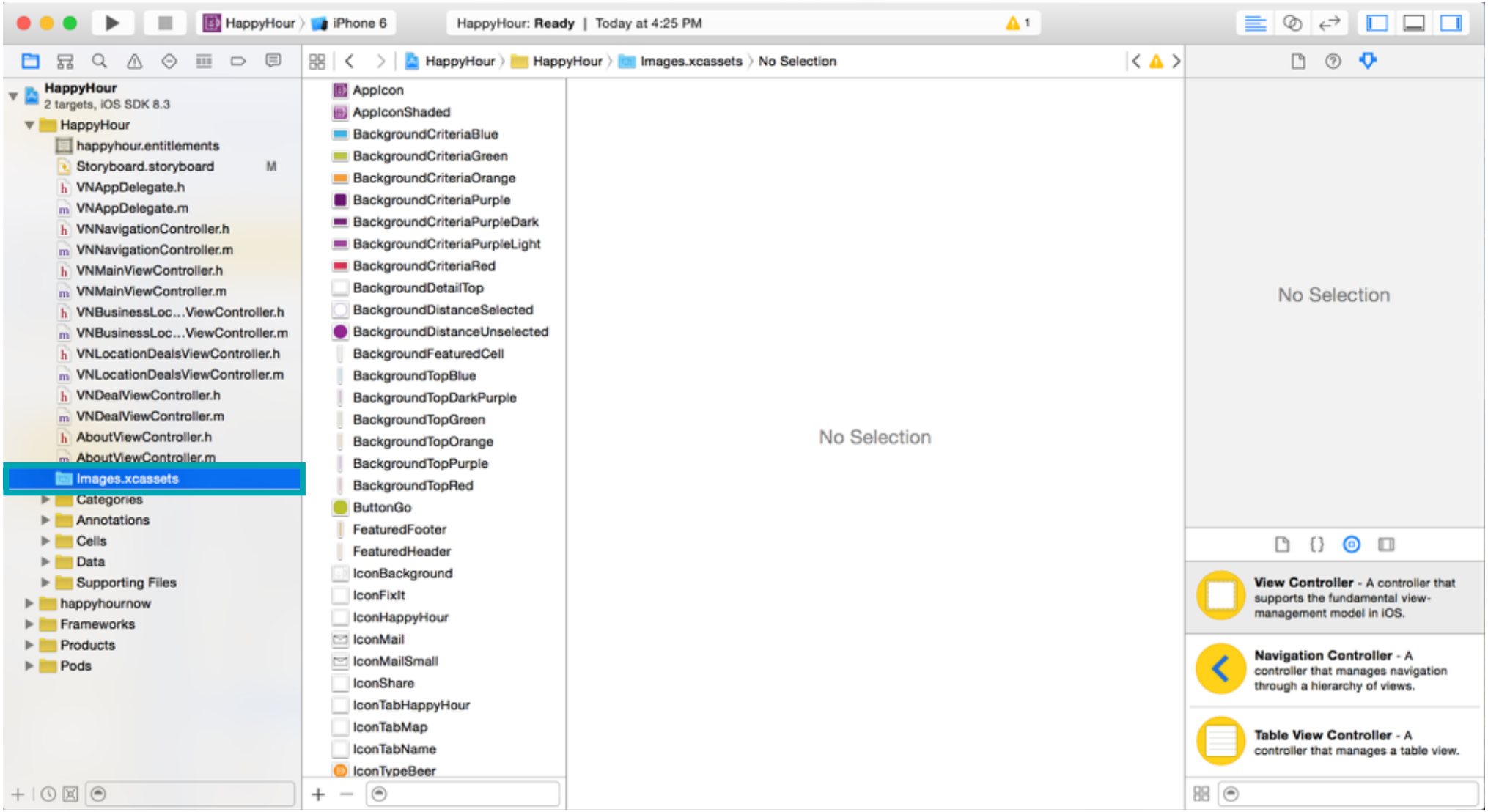
Image Insets



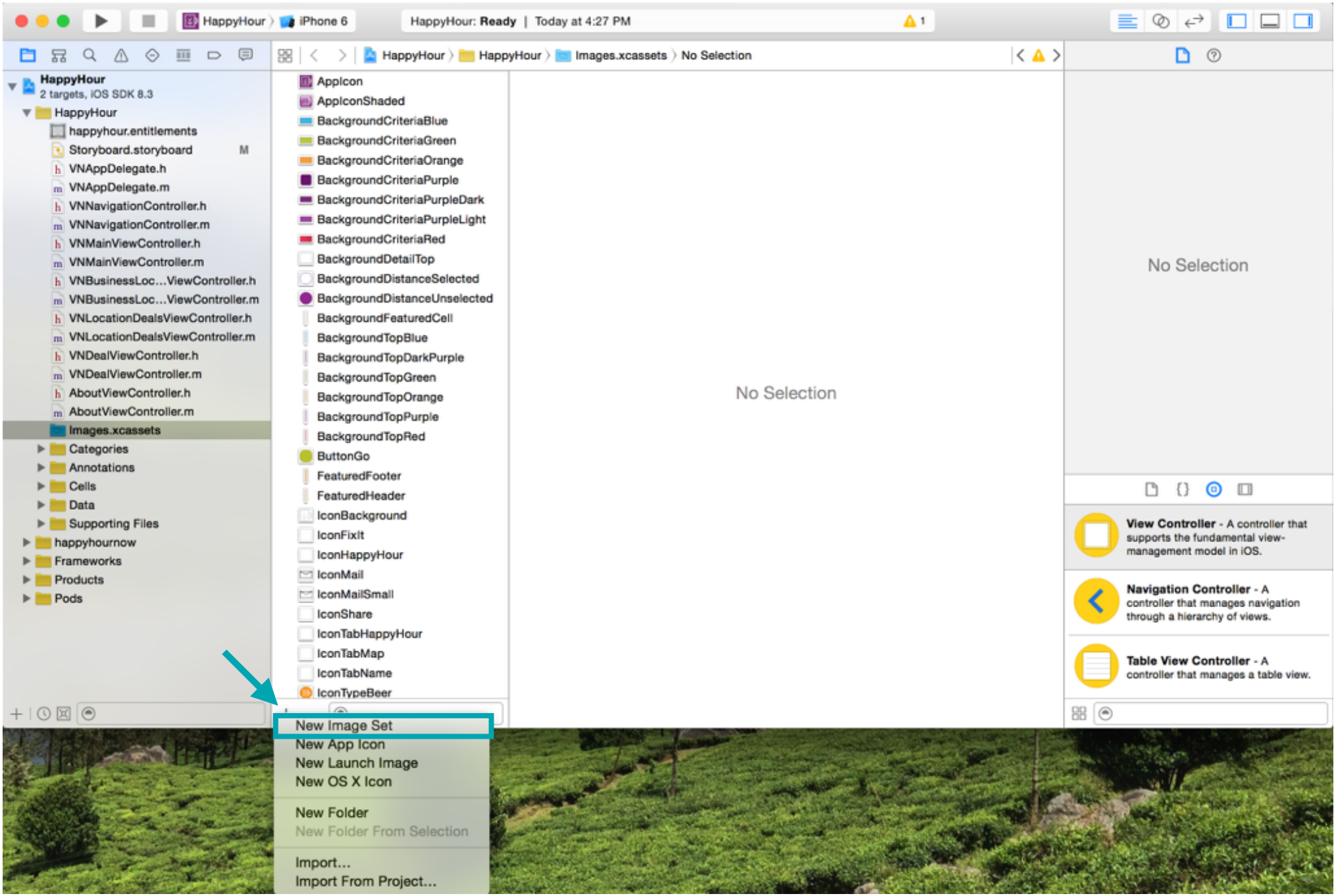
```
[_myImageView setImage:
    [[UIImage imageNamed:@“myimage.png"]
    resizableImageWithCapInsets:
    UIEdgeInsetsMake(0.0, 10.0, 10.0, 0.0)]];
```

Adding Images

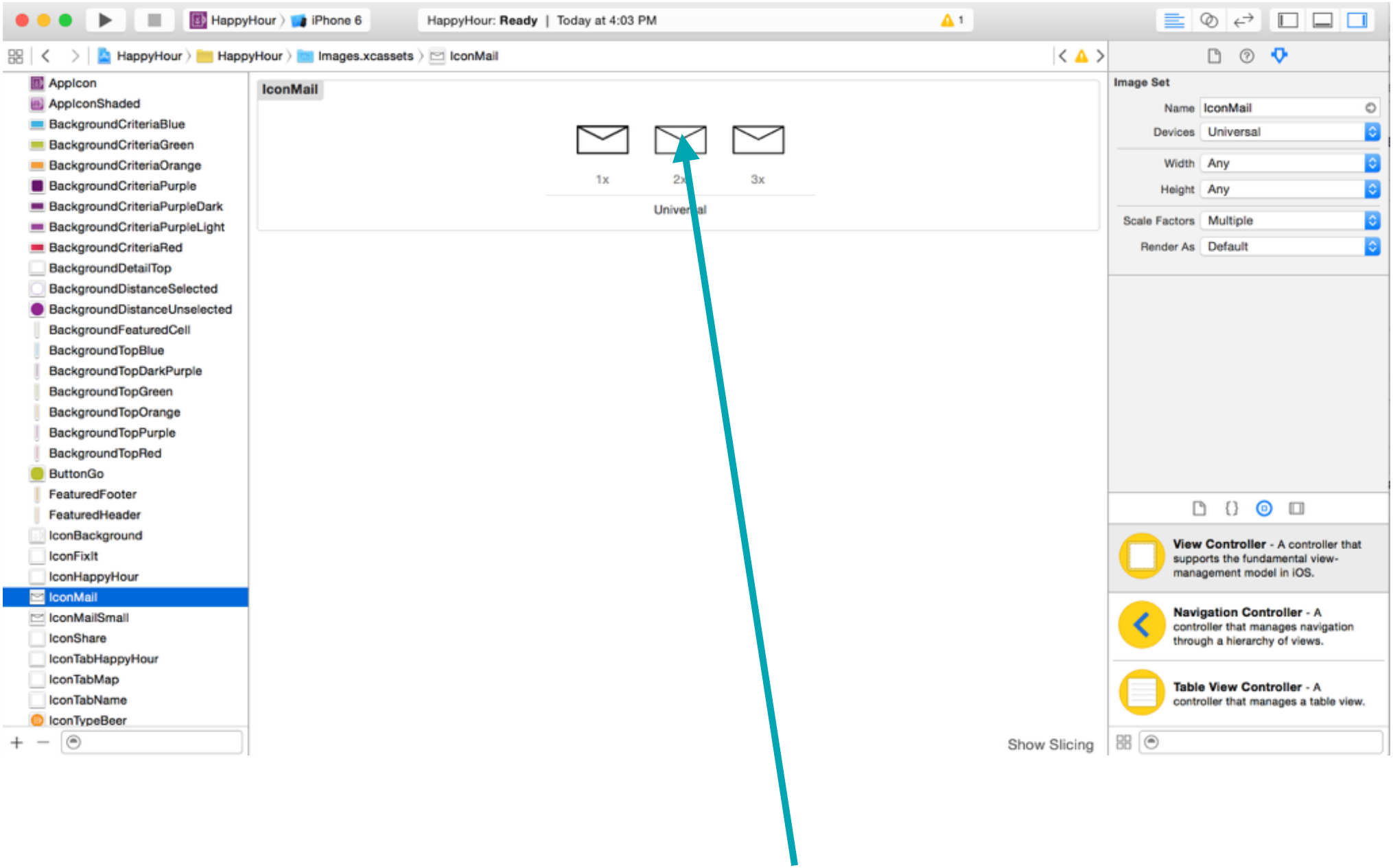
Adding Images



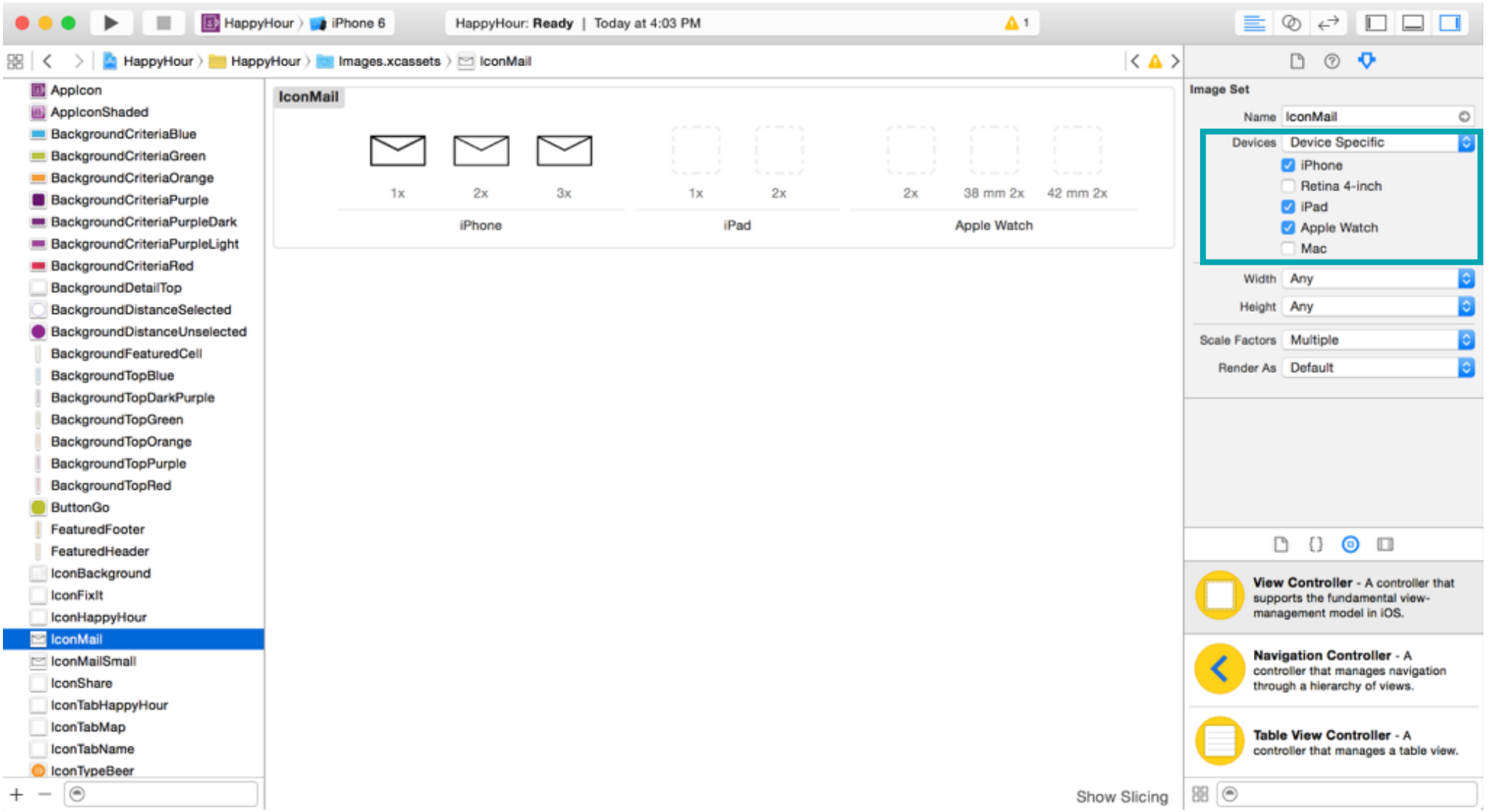
Adding Images



Adding Images



Adding Images



Adding Files

Adding Files

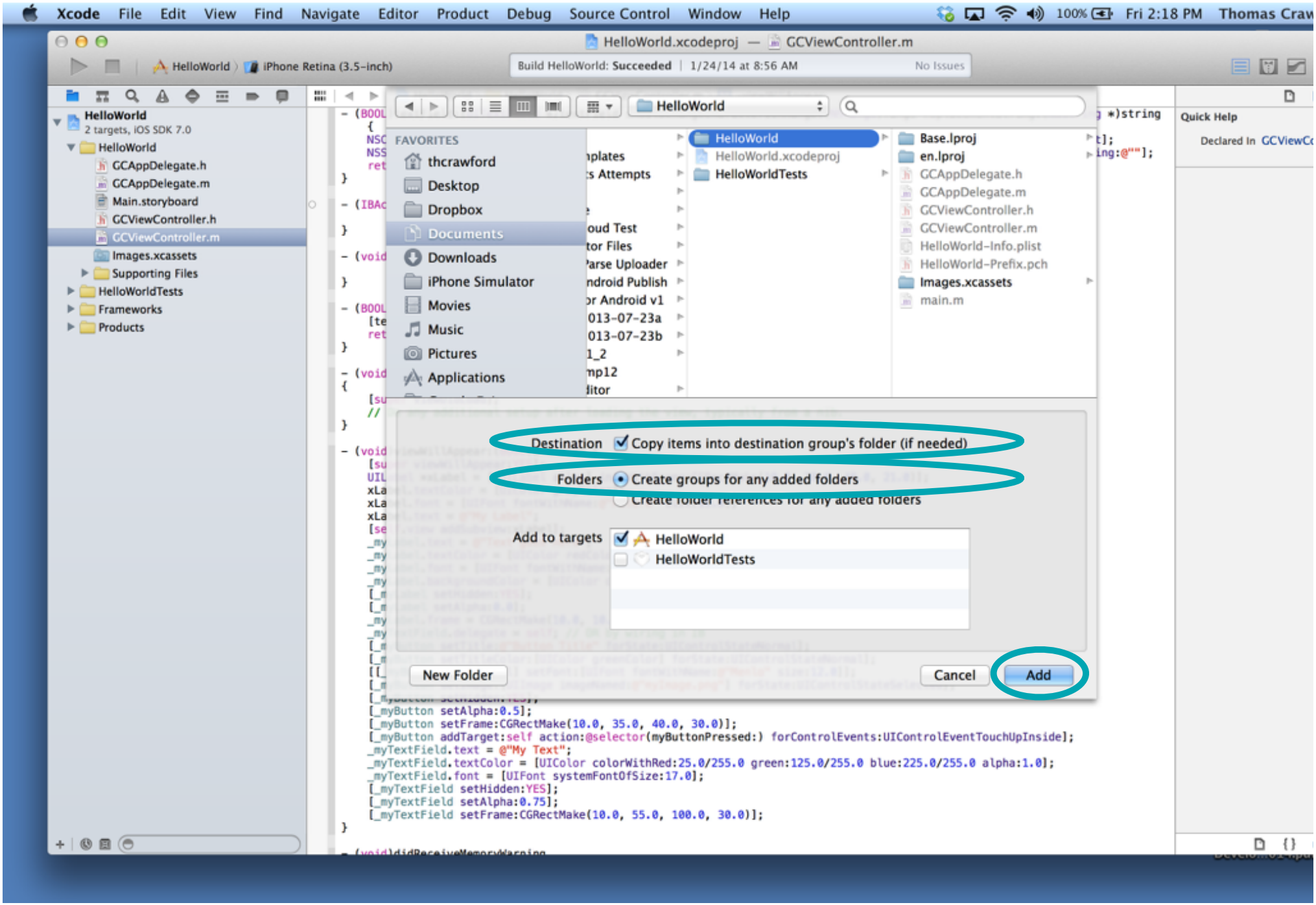
There are **multiple** ways to add files

There's only **one** that works consistently:

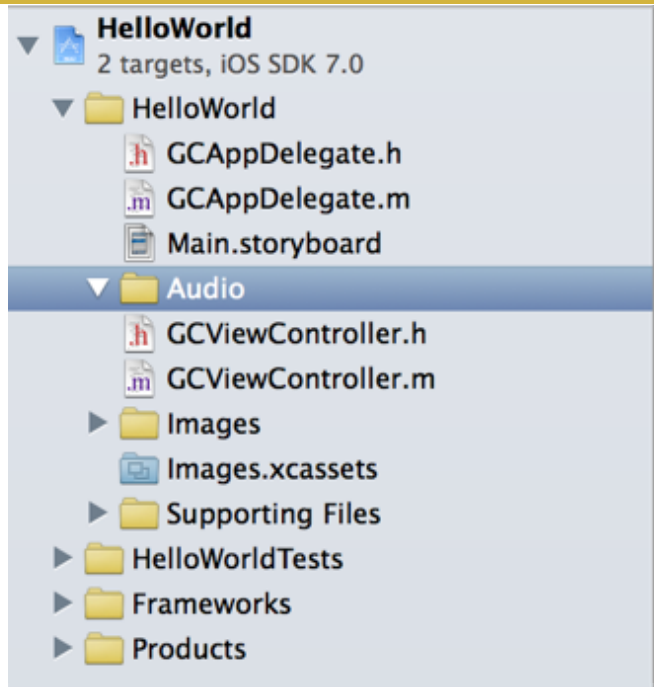
1. Place the files to be added into the project directory on your hard drive
2. File > Add Files to "yourproject"...
3. Select the files or folders to add
4. **Always, always, always** select "Copy items into destination's group folder (if needed)"
5. Select "Create Groups for any added folders"
6. Click "Add"

Other ways seem easier, but don't do it!

Adding Files



File Groups in Xcode



Groups look like folders

They are **not**

They are just logical

groupings of files and have nothing to do with the file structure on your hard drive

Organize them any way you want

Random Numbers

Random Numbers

```
int totalOptions = 451;
int randomIndex = arc4random_uniform((uint32_t)totalOptions);
```

Parsing CSV

Parsing CSV

```
- (NSString *)readBundleFileToString:(NSString *)filename ofType:
  (NSString *)type {

    NSString *path = [[NSBundle mainBundle]
        pathForResource:filename ofType:type];
   return [NSString stringWithContentsOfFile:path
        encoding:NSUTF8StringEncoding error:NULL];

}

- (NSArray *)convertCSVStringToArray:(NSString *)csvString {

    NSString *cleanString = [[csvString
        componentsSeparatedByCharactersInSet:[NSCharacterSet
        newlineCharacterSet]] componentsJoinedByString:@""];
    NSCharacterSet *set = [NSCharacterSet
        characterSetWithCharactersInString:@","];
    return [cleanString
        componentsSeparatedByCharactersInSet:set];

}
```