

# Instant Gratification

in Interface Builder

# Contents

- UIView refresher
- Getting instant feedback in IB
- Tips and Tricks
- Problems and workarounds

Demo

**GOOD  
COP**

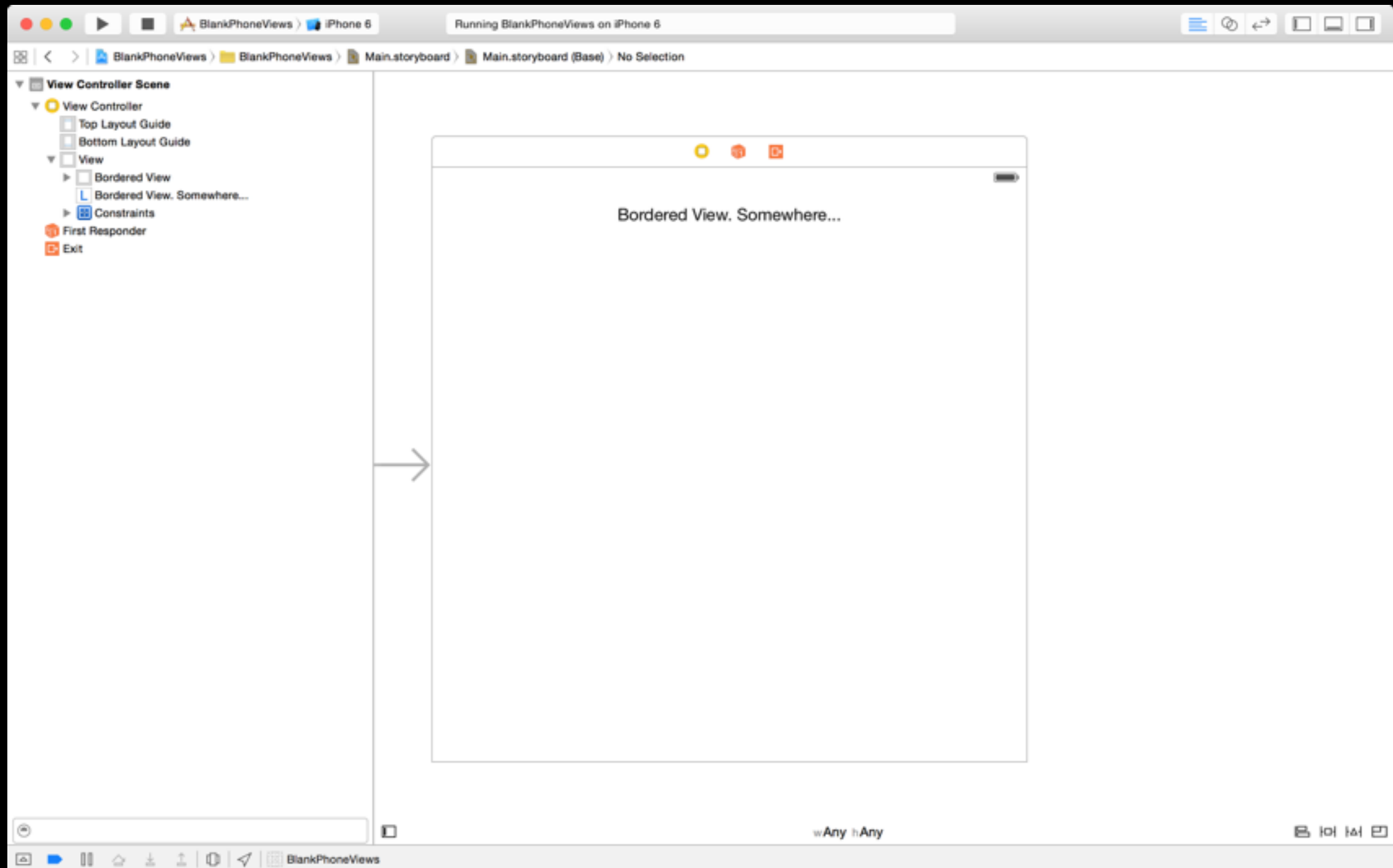


**BAD  
COP**

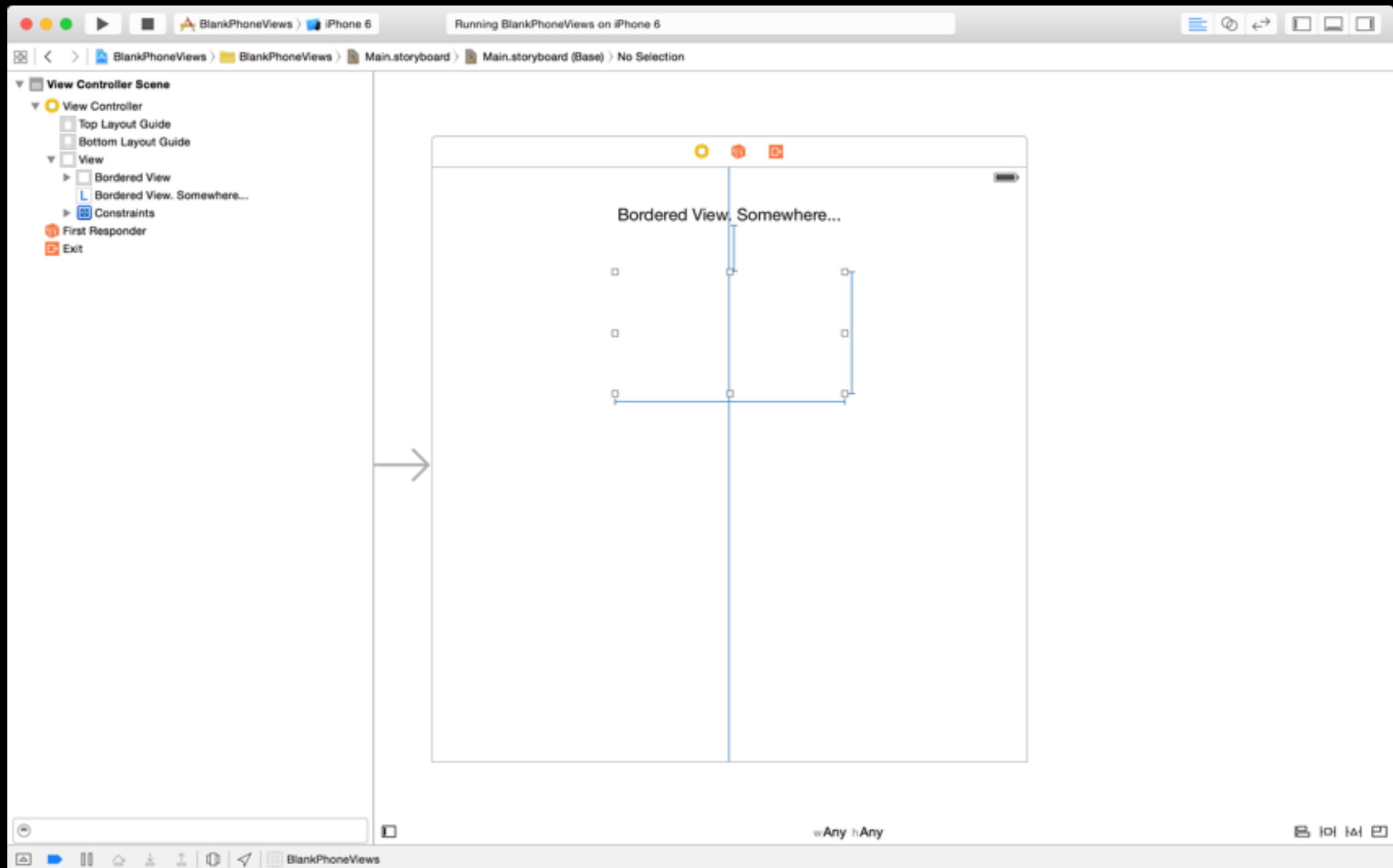


Before

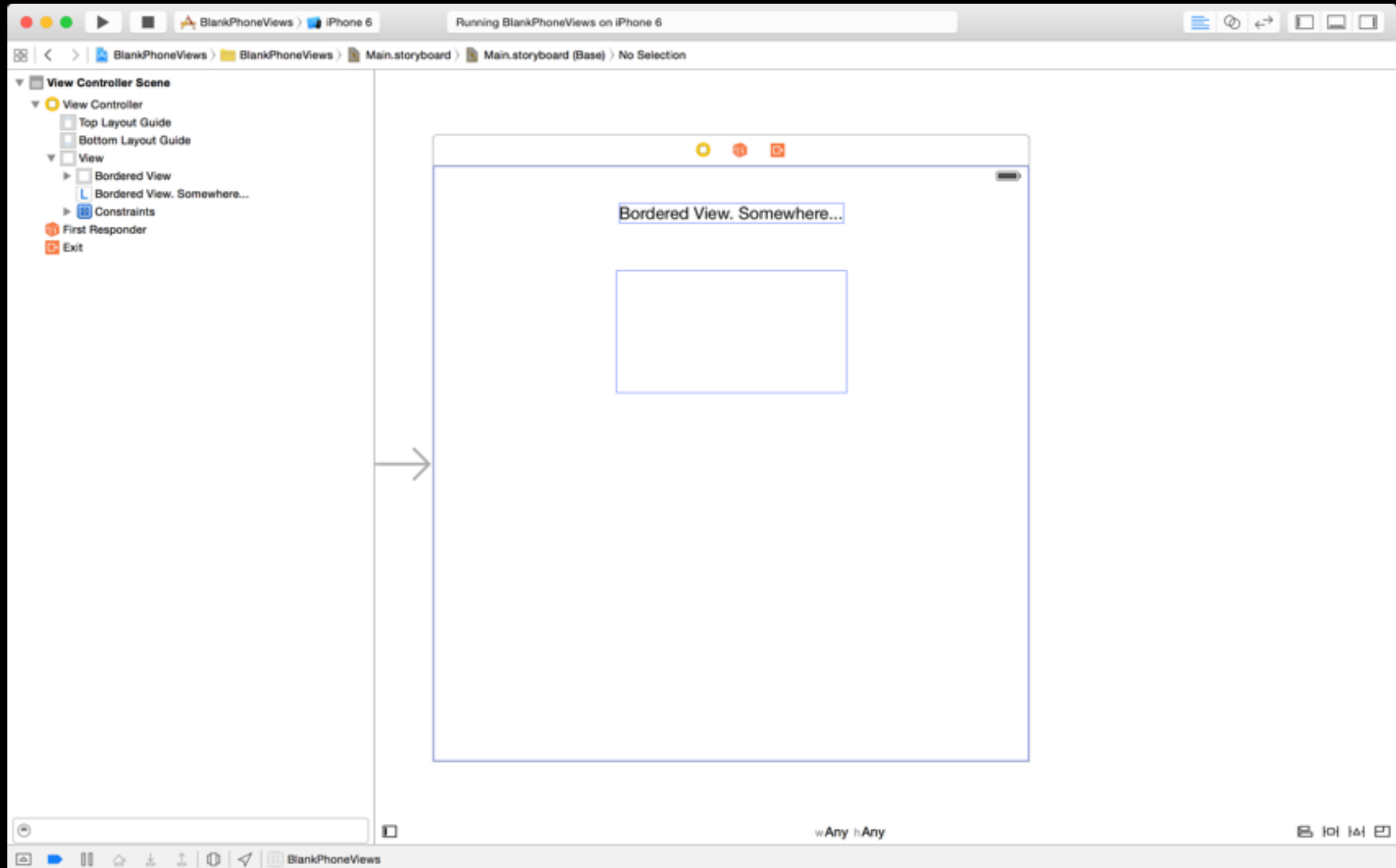
# Subview?



# See Constraints



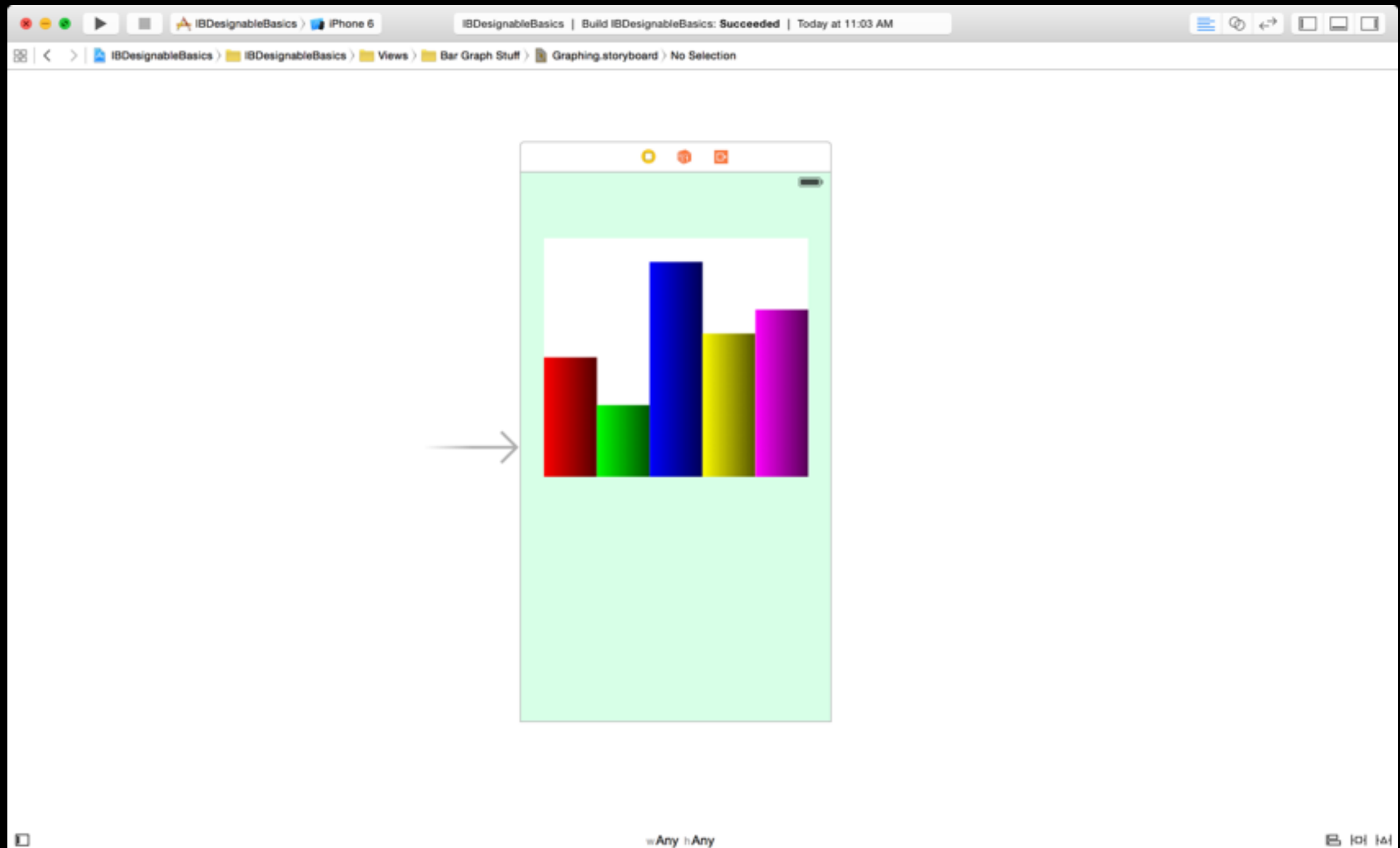
# Show Bounds





Now

# Instant Feedback



```
class BarGraph: UIView {  
    // code  
}
```

```
@IBDesignable
```

```
class BarGraph: UIView {  
    // code  
}
```

# Custom Drawing

Overview

```
import UIKit
```

```
class ColoredView: UIView {
```

```
}
```

```
import AppKit
```

```
class ColoredView: NSView {
```

```
}
```

```
override func drawRect(rect: CGRect) {  
    super.drawRect(rect)  
  
    let orange = UIColor.orangeColor()  
    orange.set()  
  
    let path = UIBezierPath(rect: rect)  
    path.fill()  
  
}
```



```
override func drawRect(rect: CGRect) {  
    super.drawRect(rect)  
  
    let context: CGContextRef = UIGraphicsGetCurrentContext()  
  
    // set color  
    let red = UIColor.redColor()  
    CGContextSetFillColorWithColor(context, red.CGColor)  
  
    // draw  
    CGContextFillRect(context, rect)  
  
}
```

```
override func drawRect(dirtyRect: NSRect) {  
    super.drawRect(dirtyRect)  
  
    let nsContext = NSGraphicsContext.currentContext()  
    let context: CGContextRef = nsContext!.CGContext  
  
    // color  
    let red = NSColor.redColor()  
    CGContextSetFillColorWithColor(context, red.CGColor)  
  
    // draw  
    CGContextFillRect(context, dirtyRect)  
  
}
```

My first view

# Configurable Views

```
var borderColor: UIColor = UIColor.blackColor()
```

```
var cornerRadius: CGFloat = 10.0
```

```
var borderWidth: CGFloat = 2.0
```

```
private func updateLayerForVariables() {  
    layer.borderWidth = borderWidth  
    layer.borderColor = borderColor.CGColor  
    layer.cornerRadius = cornerRadius  
    layer.backgroundColor = fillColor.CGColor  
}
```

```
var cornerRadius: CGFloat = 10.0 {  
    didSet {  
        updateLayerForVariables()  
    }  
}
```

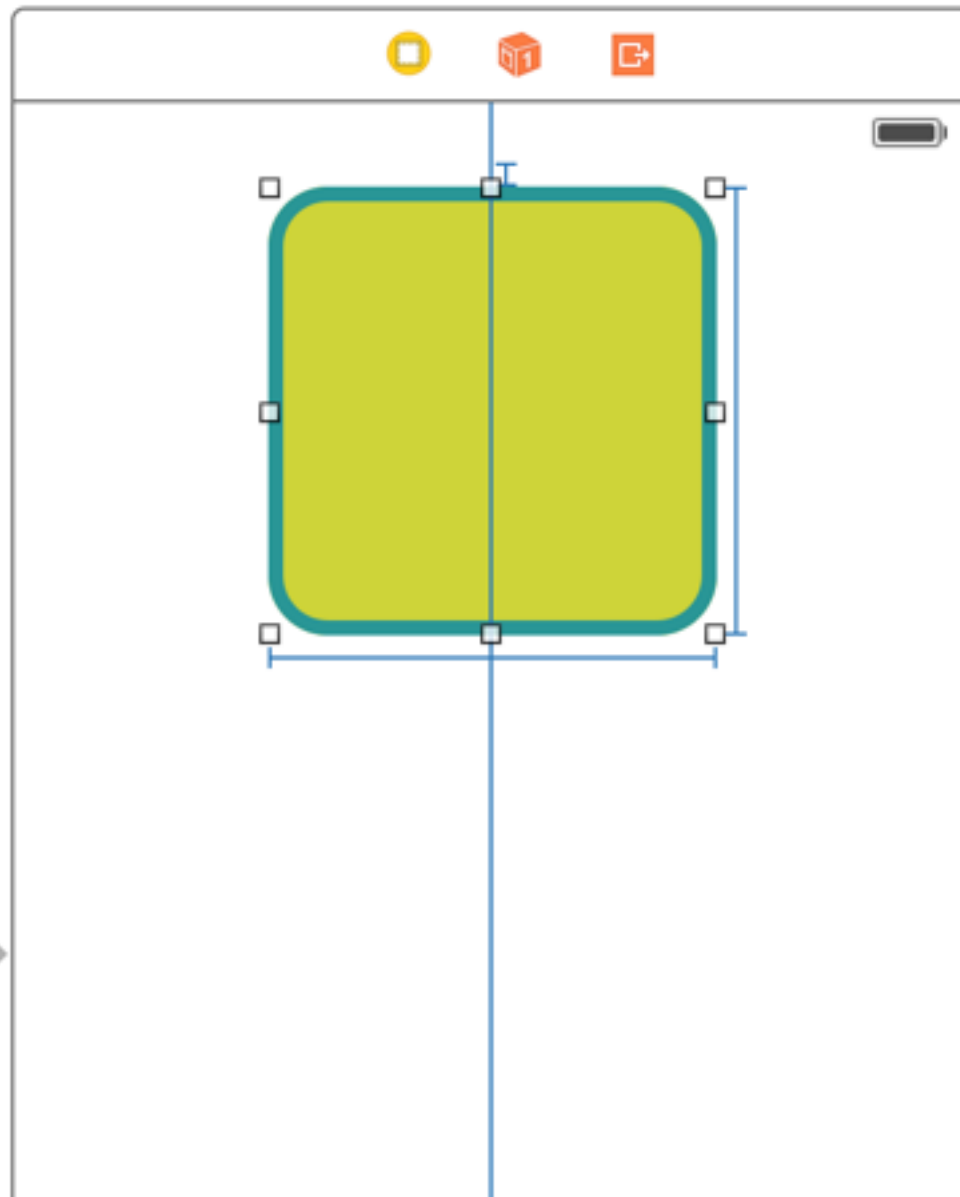
IBinspectable



```
@IBInspectable var borderColor: UIColor = UIColor.blackColor()
```

```
@IBInspectable var cornerRadius: CGFloat = 10.0
```

```
@IBInspectable var borderWidth: CGFloat = 2.0
```



### Bordered View

Border Width

Corner Radius

Border Color

Fill Color

### View

Show

Bool

Integer

Float

String

CGRect

CGPoint

CGSize

Color

Image

Views that rely on data

```
protocol BarGraphDataSource: class {  
    func numberOfBarsInGraph(bg: BarGraph) -> Int  
    func barGraph(bg: BarGraph, percentAtIndex idx: Int) -> CGFloat  
}
```

```
@IBDesignable
class BarGraph: UIView {

    weak var dataSource: BarGraphDataSource?

    // Draw based on data source

}
```

Dummy Data Source

```
private class DummyDataSource: BarGraphDataSource {  
    private let dummyData: [CGFloat] = [0.2, 0.6, 0.12, 0.85, 0.23]  
  
    func numberOfBarsInGraph(bg: BarGraph) -> Int {  
        return dummyData.count  
    }  
  
    func barGraph(bg: BarGraph, percentAtIndex idx: Int) -> CGFloat {  
        return dummyData[idx]  
    }  
}
```



```
override fun prepareForInterfaceBuilder() {  
    super.prepareForInterfaceBuilder()  
    dataSource = DummyDataSource()  
}
```

# Wrapper Methods

```
// default / dummy data
private var defaultHeights: [CGFloat] {
    return [0.5, 0.3, 0.9, 0.6, 0.7]
}

// ask data source
// if data source is nil, use defaults
private var numberOfBars: Int {
    return dataSource?.numberOfBarsInGraph(self) ??
        defaultHeights.count
}

private func percentHeightForBarIndex(idx: Int) -> CGFloat {
    return
        dataSource?.barGraph(self, percentAtIndex: idx) ??
        defaultHeights[idx]
}
```

# IBDesignable Problems and Workarounds

NSFont • UIFont

# Enumeration Support

# Debugging / Logging

```
public func liveDebugLog(message: String) {  
  
    #if !(TARGET_OS_IPHONE)  
  
        // ENSURE LOG FILE EXISTS  
        let logPath = "/tmp/XcodeLiveRendering.log"  
        let fm = NSFileManager.createIfNecessary(logPath)  
  
        // GET FILE HANDLE  
        var fileHandle = NSFileHandle(forWritingAtPath: logPath)!  
        fileHandle.seekToEndOfFile()  
  
        // WRITE OUT LOG  
        let data = message.asUTF8Data  
        fileHandle.writeData(data)  
  
    #endif  
  
}
```



# CocoaPods



```
$ gem update cocoapods
```

```
# build for all architectures
post_install do |installer|
  installer.project.targets.each do |target|
    target.build_configurations.each do |config|
      config.build_settings['ONLY_ACTIVE_ARCH'] = 'NO'
    end
  end
end
end
```

Can't drag and drop from  
subview UIControls to  
create IBActions

# Use Target-Action

```
videoControlsView?.playButton.addTarget(self,  
    action: Selector("tappedPlayButton"),  
    forControlEvents: UIControlEvents.TouchUpInside)
```

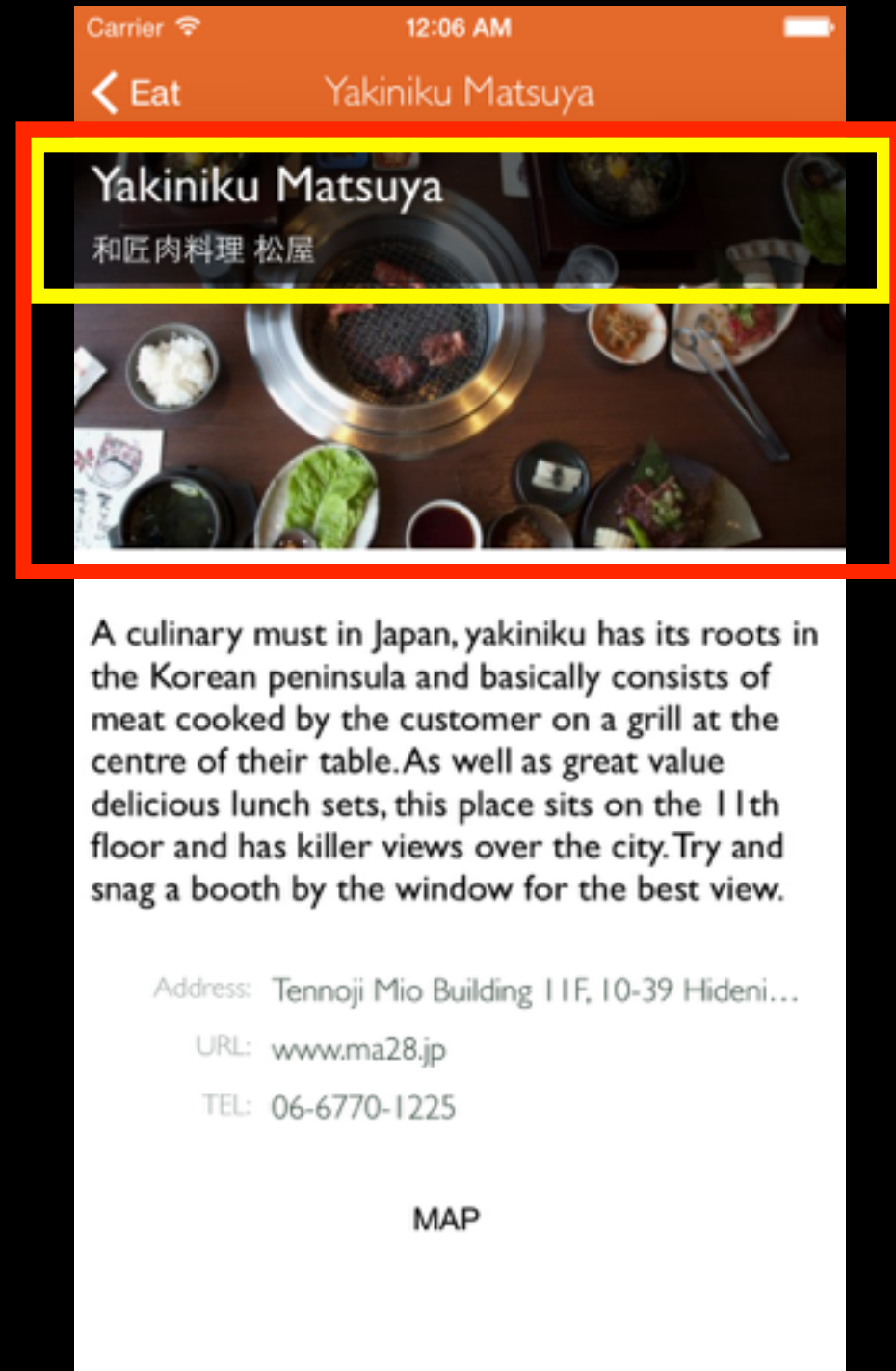
Inspectable properties  
of subviews aren't  
exposed.



# Bar



# Bar & Image



layer.backgroundColor is  
not rendered when set in  
init methods



# Benefits

- DRY
- MVC
- Quick Iteration

# Preemptive Q&A

Do I have to use a  
separate framework?

No

Does this work in  
Objective-C?

```
#import <UIKit/UIKit.h>
```

```
IB_DESIGNABLE
```

```
@interface CustomView : UIView
```

```
@property (strong, nonatomic) IBInspectable UIImage* img;
```

```
@end
```

Can I use emoji?

```
@IBInspectable var padding: CGFloat = 2.0
```

```
@IBInspectable var padding🐔: CGFloat = 2.0
```

## Padded Label

Padding

5

Padding🐔

--

## View

@MartinJNash

Seattle



