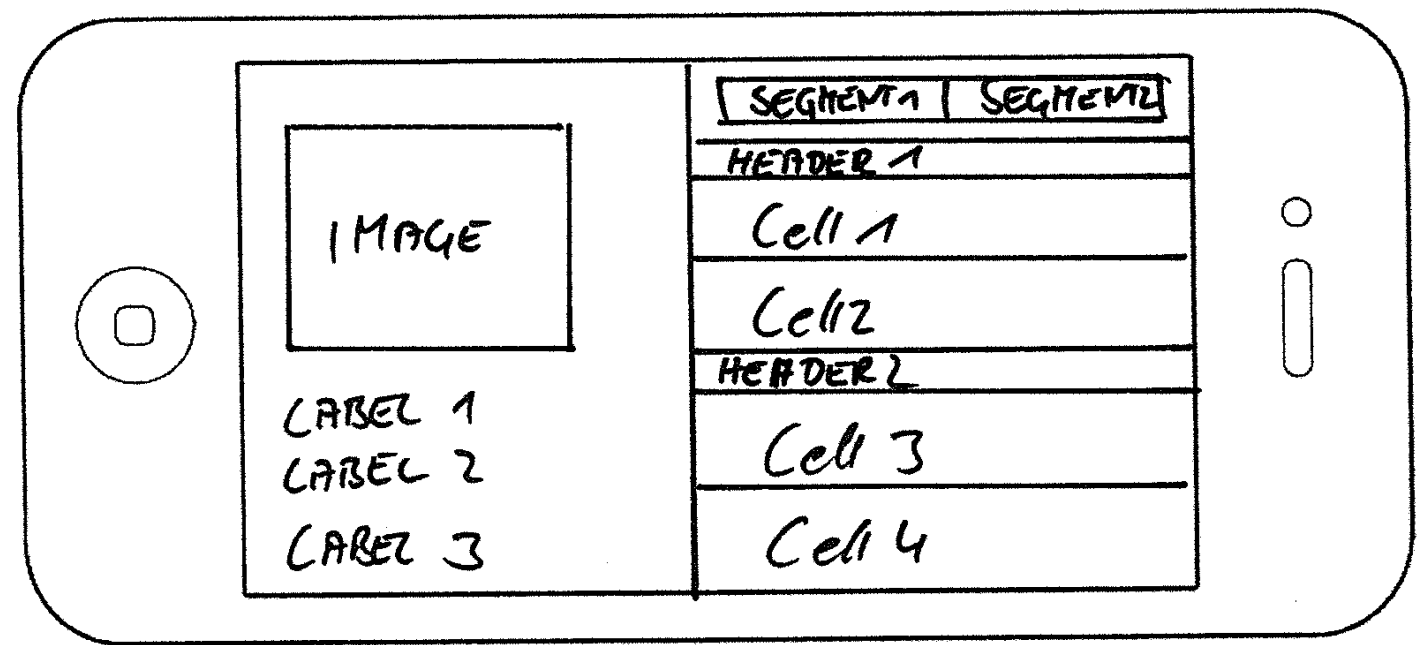
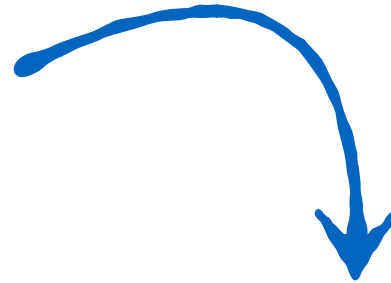
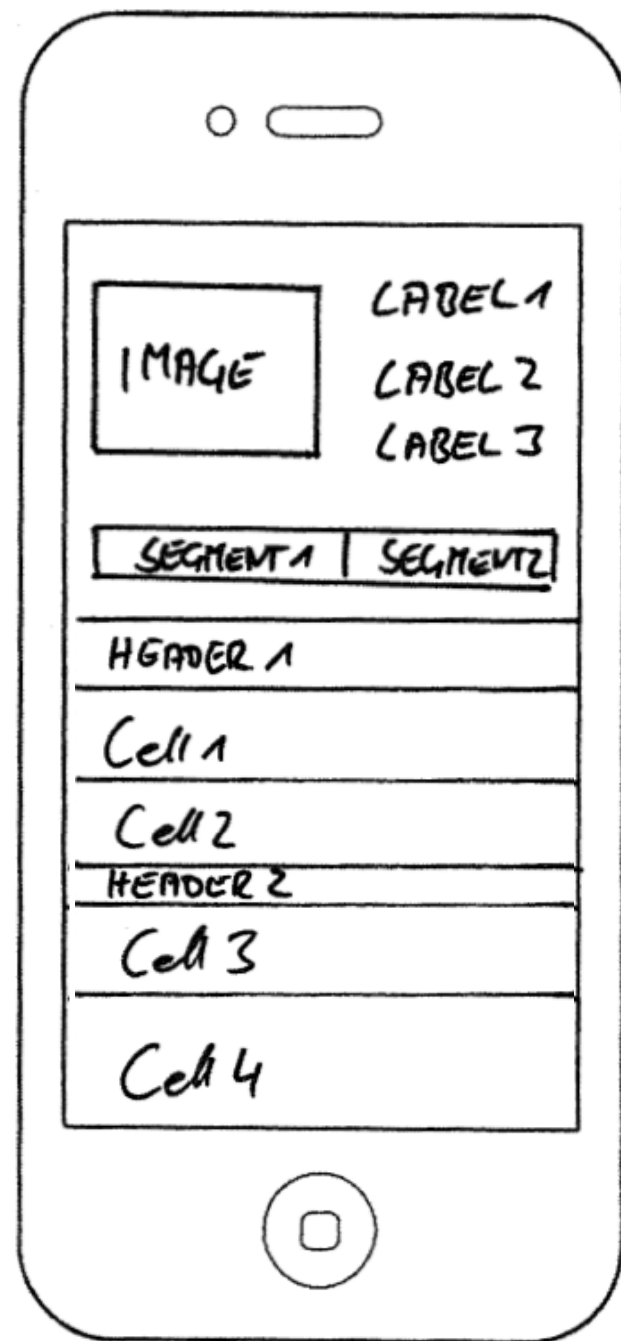


AutoLayout

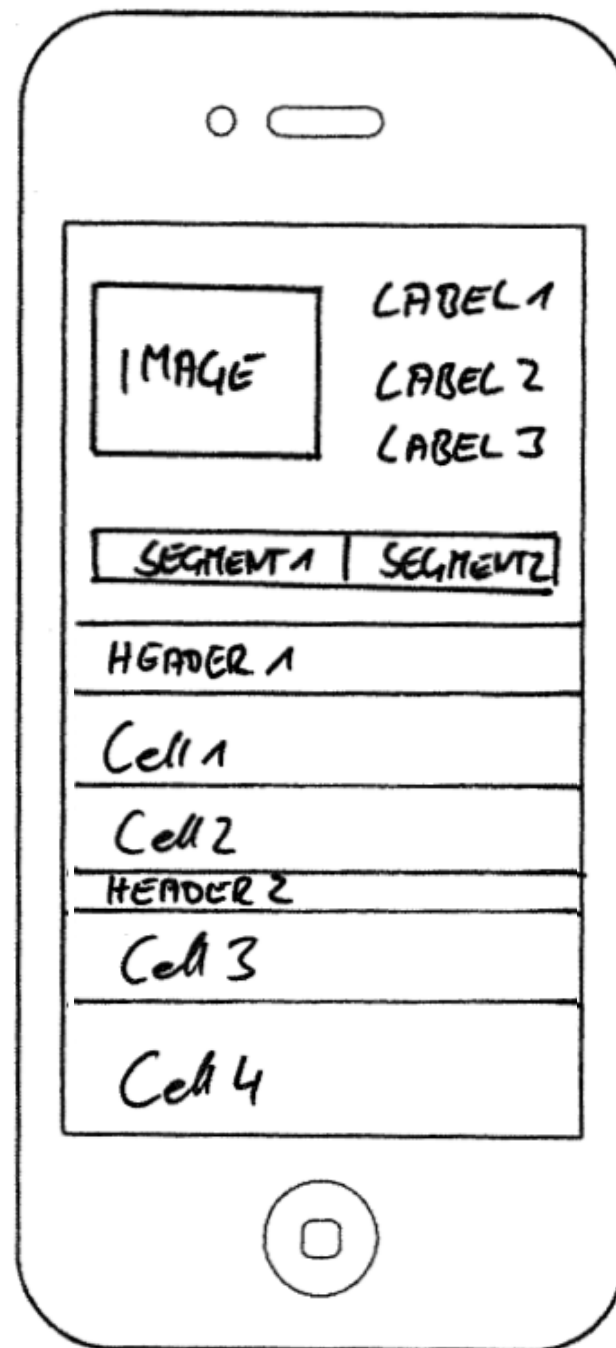
A few hints

Pit Garbe

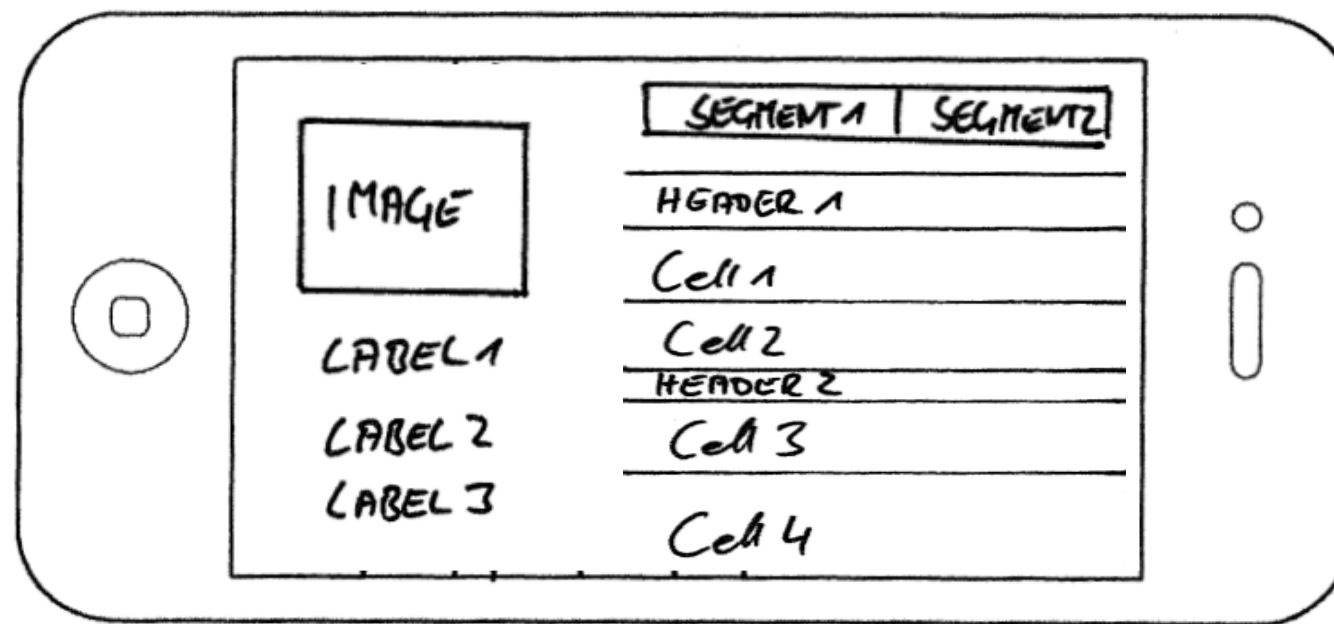
CocoaHeads Dresden, 14.01.2014



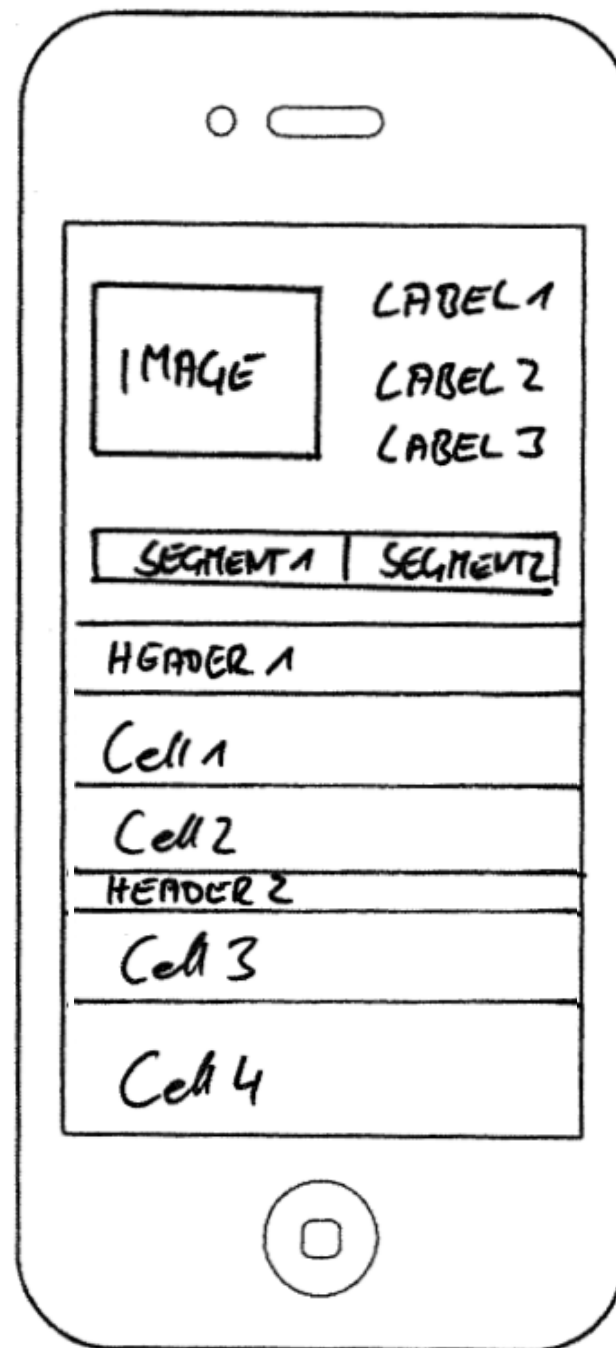
Separate Layouts
for Portrait and Landscape



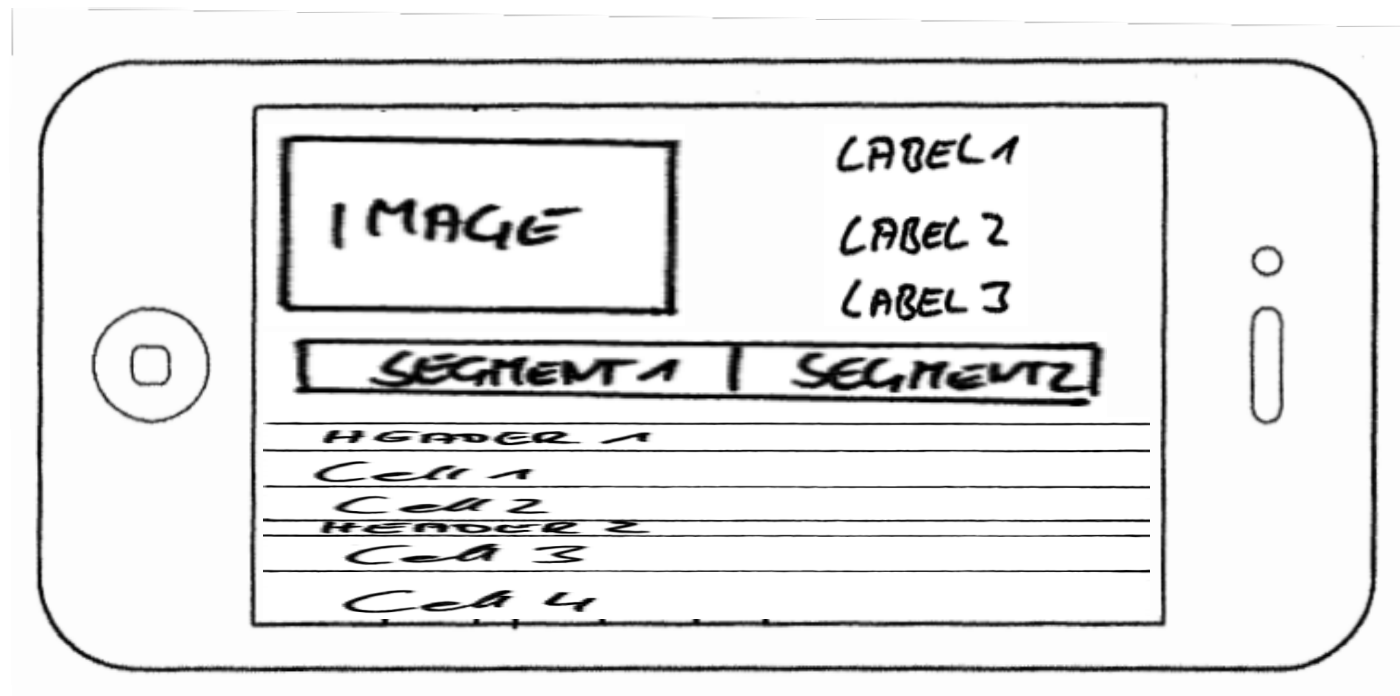
Rotation - Desired Behaviour



Rotation - Desired Behaviour



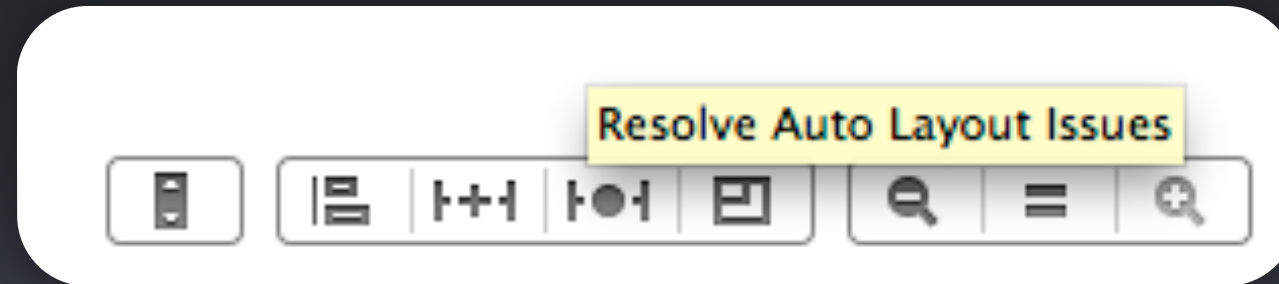
Rotation - Probable Behaviour



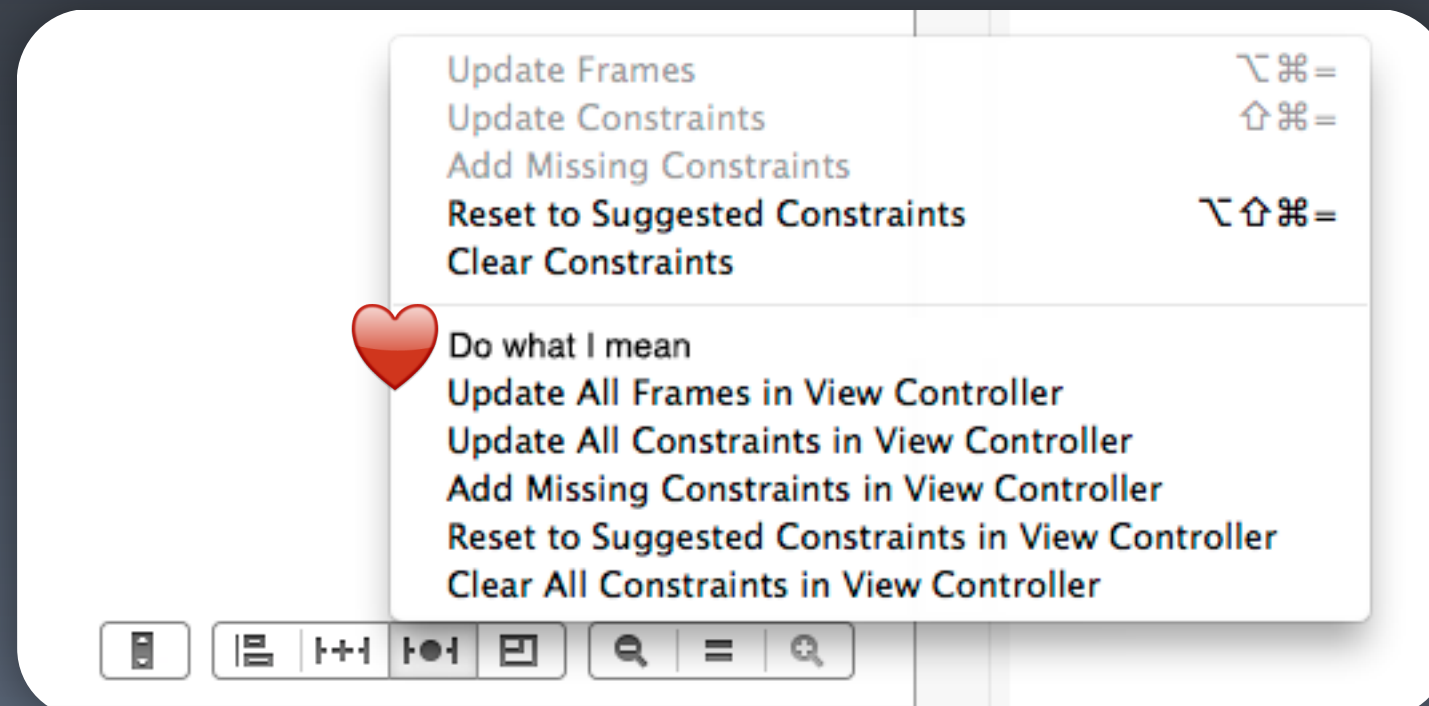
Rotation - Probable Behaviour

The Solution

1.



2.

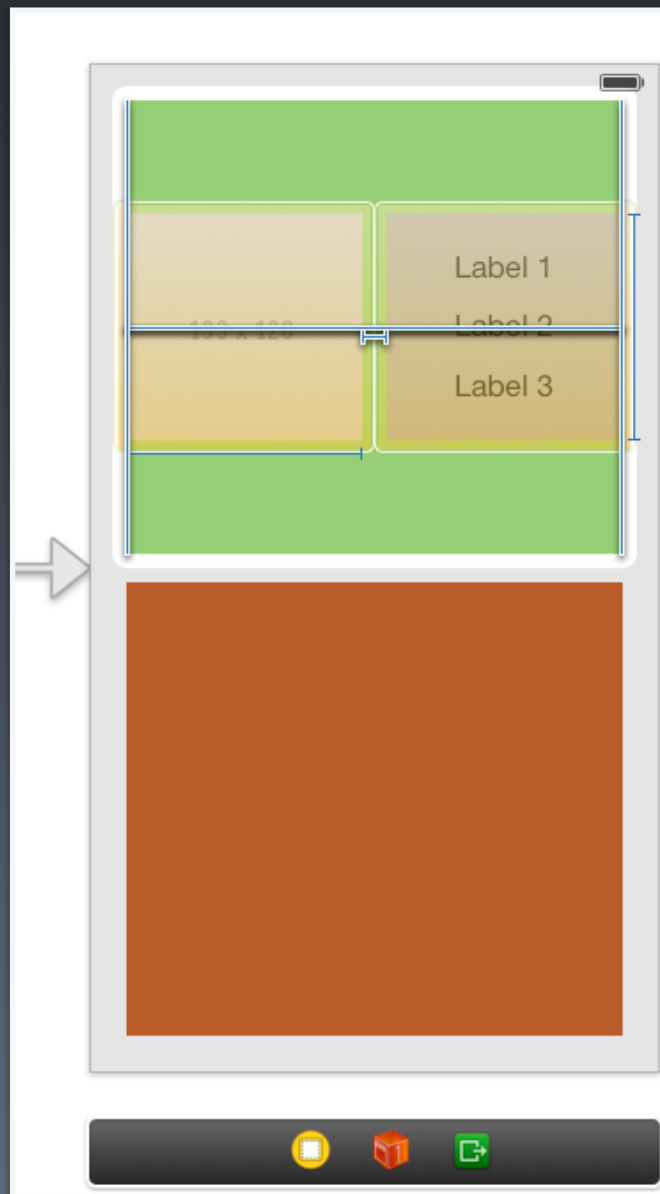


A (real) Solution

- Add constraints for one orientation completely in IB
- Add constraints, that ONLY exist in this orientation but NOT in the other to an IBOutletCollection
- in `-updateConstraints` create the constraints for the other orientation (once)
- then, depending on the current orientation, remove the old constraints and install the others

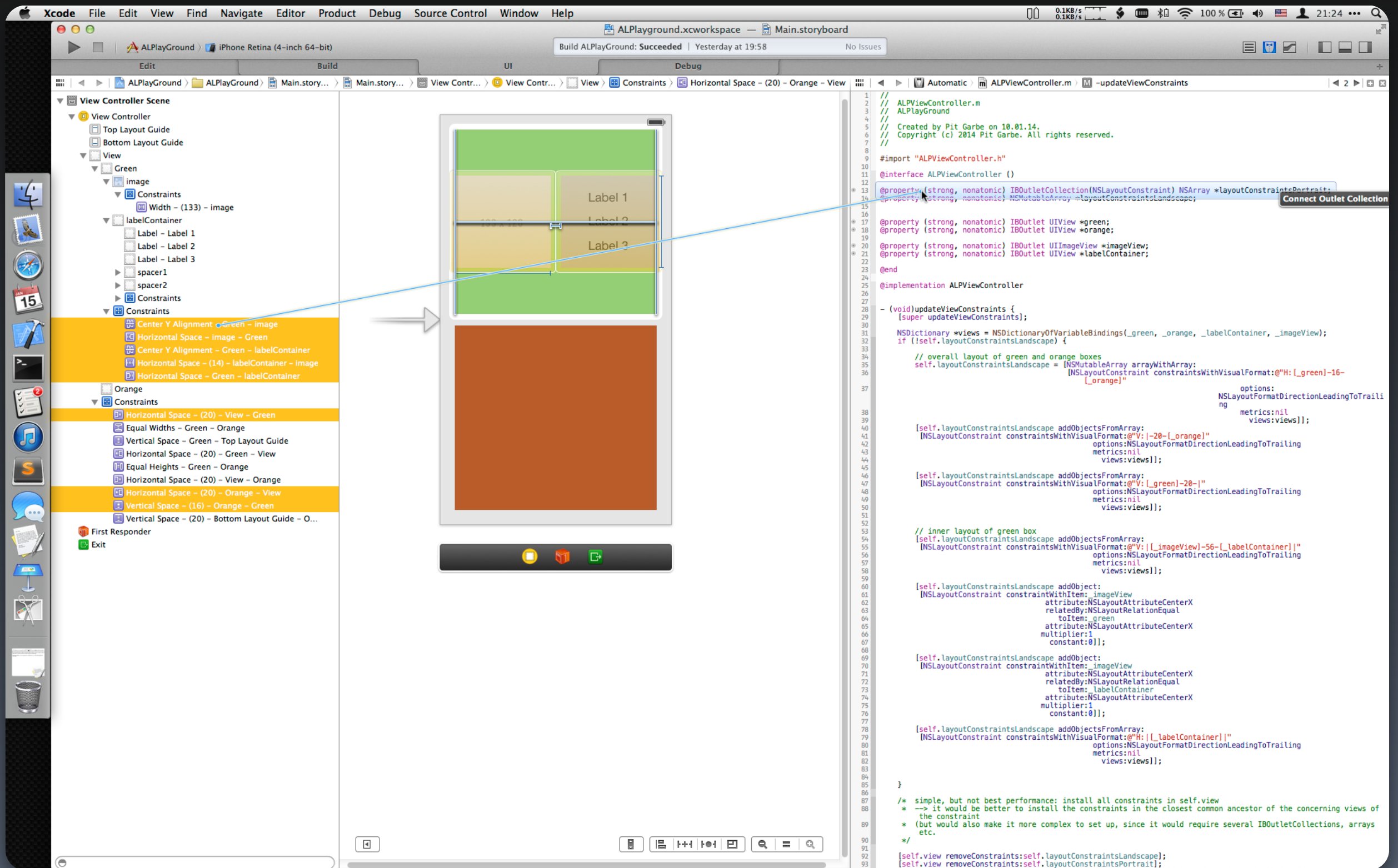
Select relevant constraints

Constraints concerning the inner layout inside the green subview



Constraints concerning the layout of the green and orange subviews





Put them all into an IBOutletCollection

0.1KB/s 0.1KB/s 100 % 21:24

Main.storyboard

at 19:58 No Issues

space - (20) - Orange - View Automatic ALPViewController.m -updateViewConstraints

```
1 //
2 // ALPViewController.m
3 // ALPlayGround
4 //
5 // Created by Pit Garbe on 10.01.14.
6 // Copyright (c) 2014 Pit Garbe. All rights reserved.
7 //
8
9 #import "ALPViewController.h"
10
11 @interface ALPViewController ()
12
13 @property (strong, nonatomic) IBOutletCollection(NSLayoutConstraint) NSArray *layoutConstraintsPortrait;
14 @property (strong, nonatomic) NSMutableArray *layoutConstraintsLandscape;
15
16
17 @property (strong, nonatomic) IBOutlet UIView *green;
18 @property (strong, nonatomic) IBOutlet UIView *orange;
19
20 @property (strong, nonatomic) IBOutlet UIImageView *imageView;
21 @property (strong, nonatomic) IBOutlet UIView *labelContainer;
22
23 @end
24
25 @implementation ALPViewController
26
27
28
29
30
31
32 if (!self.layoutConstraintsLandscape) {
33
34     // overall layout of green and orange boxes
35     self.layoutConstraintsLandscape = [NSMutableArray arrayWithArray:
36                                     [NSLayoutConstraint constraintsWithVisualFormat:@"H: [_green]-16-
37                                     [_orange]"
38                                     options:
39                                     NSLayoutFormatDirectionLeadingToTrailing
40                                     metrics:nil
41                                     views:views]];
42
43     [self.layoutConstraintsLandscape addObjectsFromArray:
44     [NSLayoutConstraint constraintsWithVisualFormat:@"V: | -20- [_orange]"
45     options: NSLayoutFormatDirectionLeadingToTrailing
46     metrics:nil
47     views:views]];
48
49     [self.layoutConstraintsLandscape addObjectsFromArray:
50     [NSLayoutConstraint constraintsWithVisualFormat:@"V: [_green]-20-|"
51     options: NSLayoutFormatDirectionLeadingToTrailing
52     metrics:nil
53     views:views]];
```

Connect Outlet Collection

Also add IBOutlets for the important views and an array for the constraints that are going to be created in code.


```

26
27
28 - (void)updateViewConstraints {
29     [super updateViewConstraints];
30
31     NSDictionary *views = NSDictionaryOfVariableBindings(_green, _orange, _labelContainer, _imageView);
32     if (!self.layoutConstraintsLandscape) {
33
34         // overall layout of green and orange boxes
35         self.layoutConstraintsLandscape = [NSMutableArray arrayWithArray:
36             [NSLayoutConstraint constraintsWithVisualFormat:@"H: [_green]-16-

```

in **-updateViewConstraints** create all the constraints for Landscape orientation (once)

```

                                options:
                                NSLayoutFormatDirectionLeadingToTrailing
                                metrics:nil
                                views:views]]];
40     [self.layoutConstraintsLandscape addObjectsFromArray:
41     [NSLayoutConstraint constraintsWithVisualFormat:@"V: |-20-[_orange]"
42         options:NSLayoutFormatDirectionLeadingToTrailing
43         metrics:nil
44         views:views]];
45
46     [self.layoutConstraintsLandscape addObjectsFromArray:
47     [NSLayoutConstraint constraintsWithVisualFormat:@"V: [_green]-20-|"
48         options:NSLayoutFormatDirectionLeadingToTrailing
49         metrics:nil
50         views:views]];
51
52
53     // inner layout of green box
54     [self.layoutConstraintsLandscape addObjectsFromArray:
55     [NSLayoutConstraint constraintsWithVisualFormat:@"V: |[_imageView]-56-[_labelContainer]|"
56         options:NSLayoutFormatDirectionLeadingToTrailing
57         metrics:nil
58         views:views]];
59
60     [self.layoutConstraintsLandscape addObject:
61     [NSLayoutConstraint constraintWithItem:_imageView
62         attribute:NSLayoutAttributeCenterX
63         relatedBy:NSLayoutRelationEqual
64         toItem:_green
65         attribute:NSLayoutAttributeCenterX
66         multiplier:1
67         constant:0]];
68
69     [self.layoutConstraintsLandscape addObject:
70     [NSLayoutConstraint constraintWithItem:_imageView
71         attribute:NSLayoutAttributeCenterX
72         relatedBy:NSLayoutRelationEqual
73         toItem:_labelContainer
74         attribute:NSLayoutAttributeCenterX
75         multiplier:1
76         constant:0]];
77
78     [self.layoutConstraintsLandscape addObjectsFromArray:
79     [NSLayoutConstraint constraintsWithVisualFormat:@"H: |[_labelContainer]|"
80         options:NSLayoutFormatDirectionLeadingToTrailing
81         metrics:nil
82         views:views]];
83
84
85     }
86

```

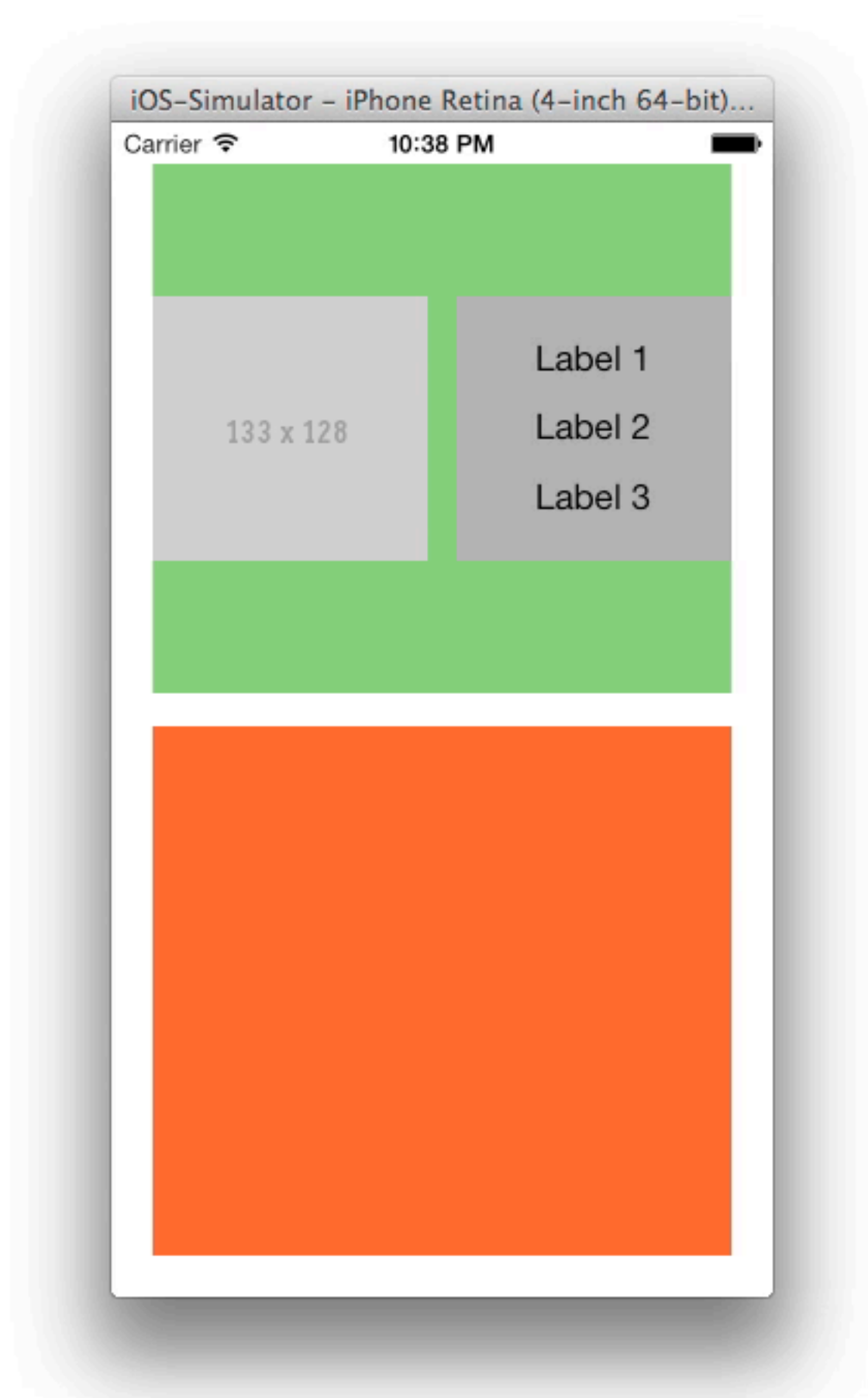
```

59
60
61 [self.layoutConstraintsLandscape addObject:
62     [NSLayoutConstraint constraintWithItem:_imageView
63         attribute:NSLayoutAttributeCenterX
64         relatedBy:NSLayoutRelationEqual
65         toItem:_green
66         attribute:NSLayoutAttributeCenterX
67         multiplier:1
68         constant:0]];
69
70 [self.layoutConstraintsLandscape addObject:
71     [NSLayoutConstraint constraintWithItem:_imageView
72         attribute:NSLayoutAttributeCenterX
73         relatedBy:NSLayoutRelationEqual
74         toItem:_labelContainer
75         attribute:NSLayoutAttributeCenterX
76         multiplier:1
77         constant:0]];
78
79 [self.layoutConstraintsLandscape addObjectsFromArray:
80     [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[_labelContainer]|"
81         options:NSLayoutFormatDirectionLeadingToTrailing
82         metrics:nil
83         views:views]];
84
85 }
86
87 /* simple, but not best performance: install all constraints in self.view
88  * --> it would be better to install the constraints in the closest common ancestor of the concerning views of
89  * the constraint
90  * (but would also make it more complex to set up, since it would require several IBOutletCollections, arrays
91  * etc.
92  */
93
94 [self.view removeConstraints:self.layoutConstraintsLandscape];
95 [self.view removeConstraints:self.layoutConstraintsPortrait];
96 [self.view addConstraints:(UIInterfaceOrientationIsPortrait(self.interfaceOrientation)) ? self.
97     layoutConstraintsPortrait : self.layoutConstraintsLandscape];

```

Depending on the orientation, remove the old constraints and install the needed constraints. The changes to the frames will automatically animate.

GitHub: <https://github.com/leberwurstsaft/ALPlayGround>



Magical Solution (not AutoLayout)

- TPMultiLayoutViewController
 - <http://atastypixel.com/blog/seamlessly-manage-portrait-and-landscape-view-controller-layouts/>
 - <https://github.com/michaeltyson/TPMultiLayoutViewController>
- 2 years old, doesn't know about AutoLayout and doesn't fully cover Springs+Struts (but it works)
- **Improved fork** (Animations, etc.)
 - <https://github.com/amolloy/TPMultiLayoutViewController>

Also worth watching

Introduction to Auto Layout for iOS and OS X	WWDC 2012	
Best Practices for Mastering Auto Layout	WWDC 2012	
Auto Layout by Example	WWDC 2012	
Interface Builder Core Concepts	Nob Hill Wednesday 9:00AM	