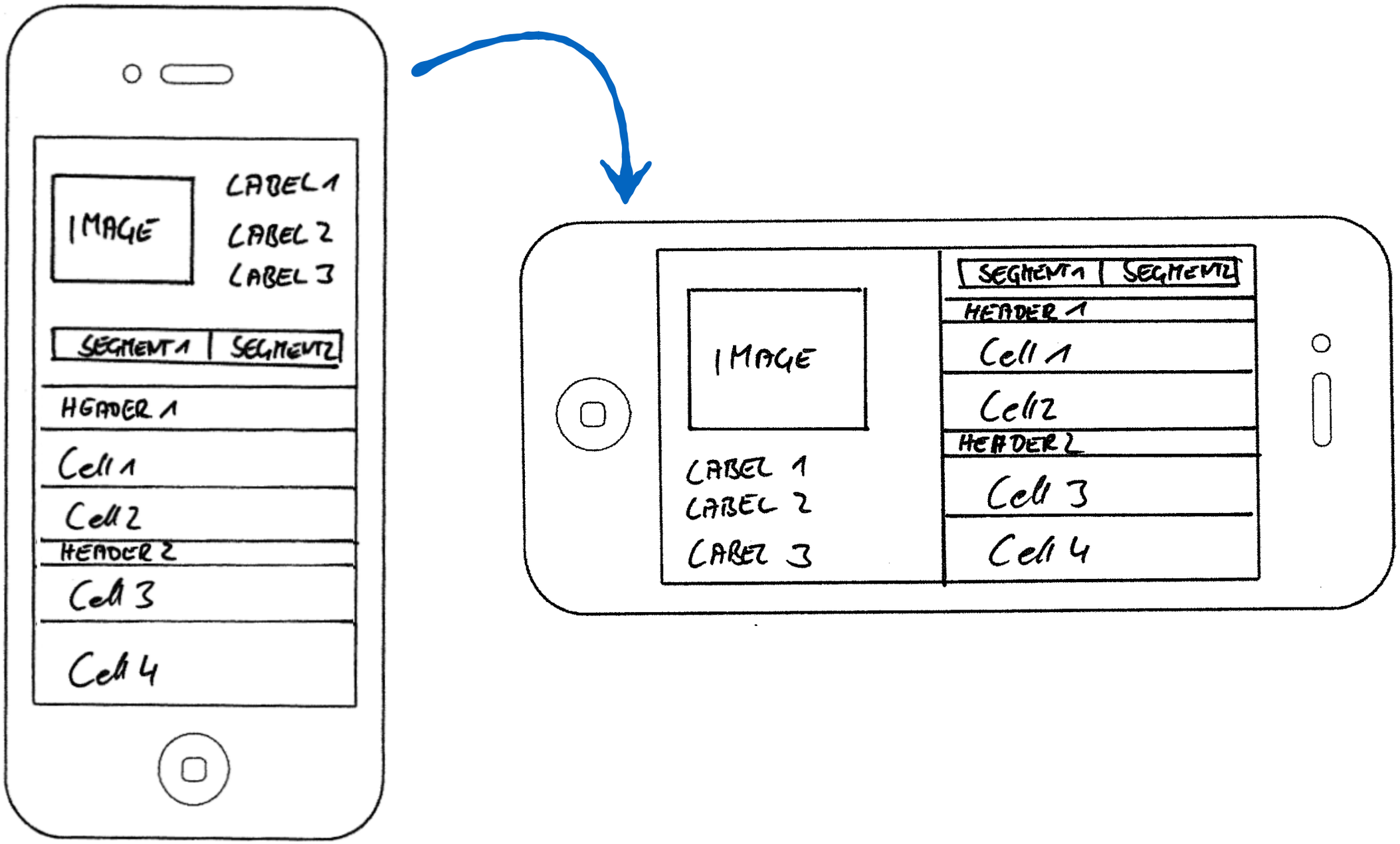
AutoLayout

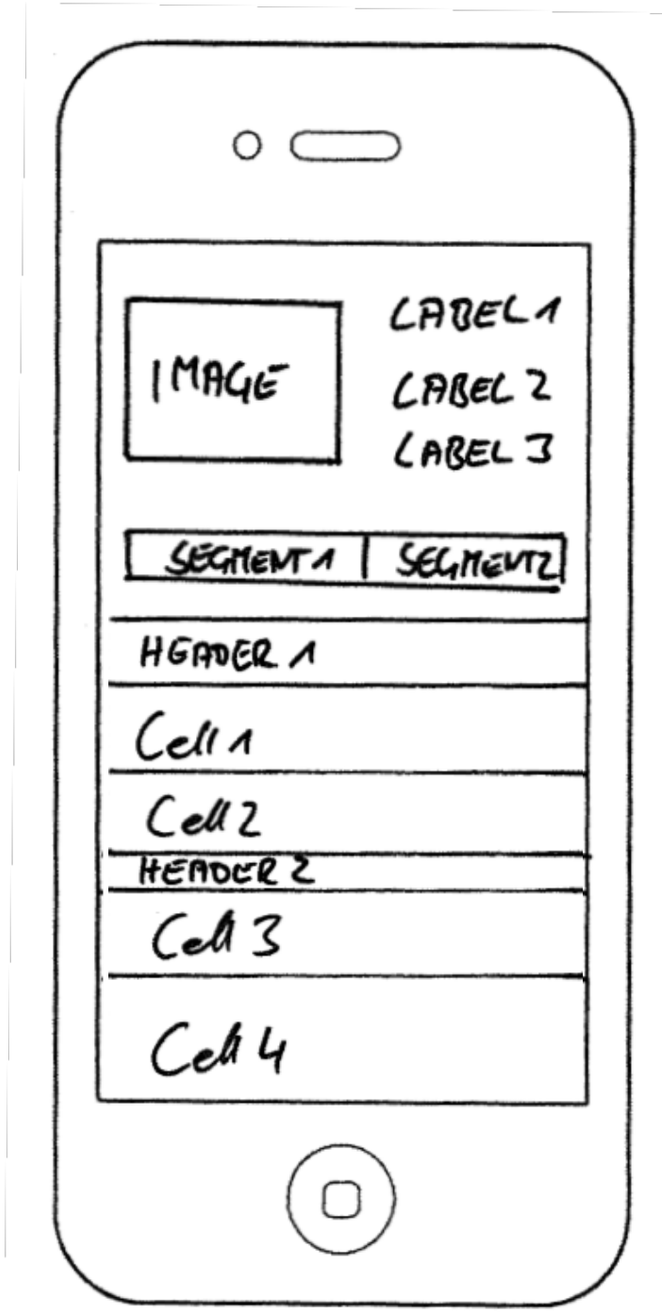
A few hints

Pit Garbe

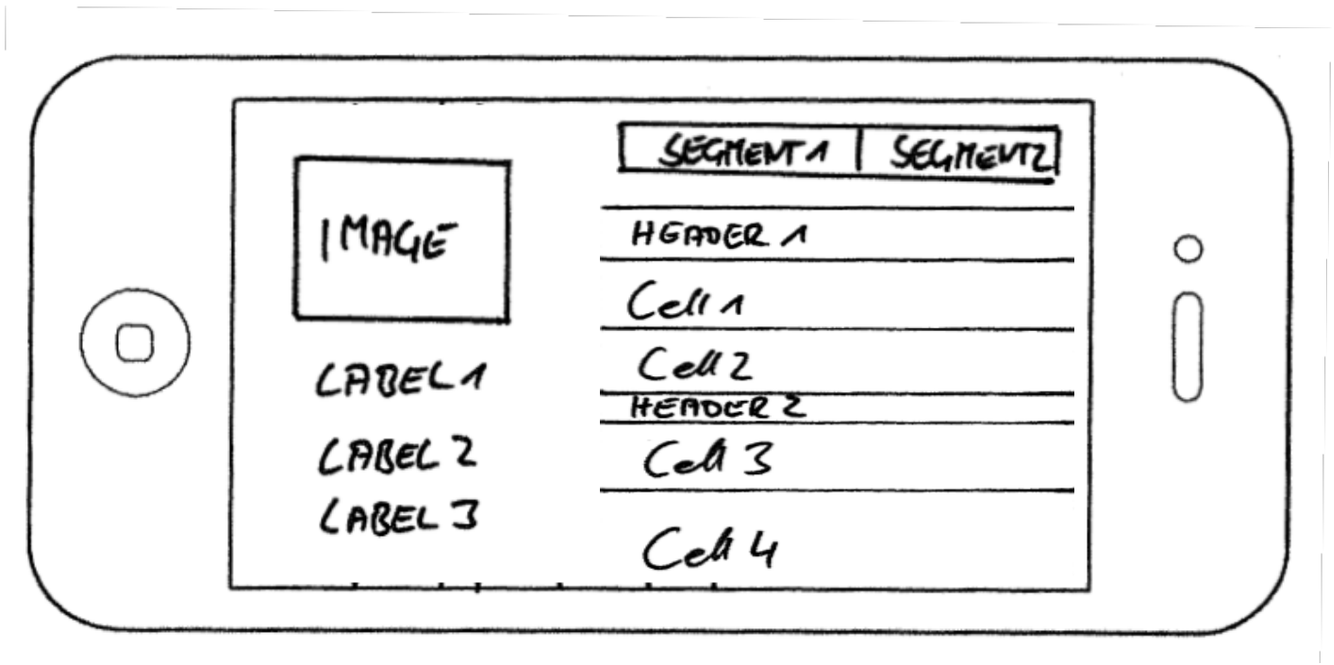
CocoaHeads Dresden, 14.01.2014



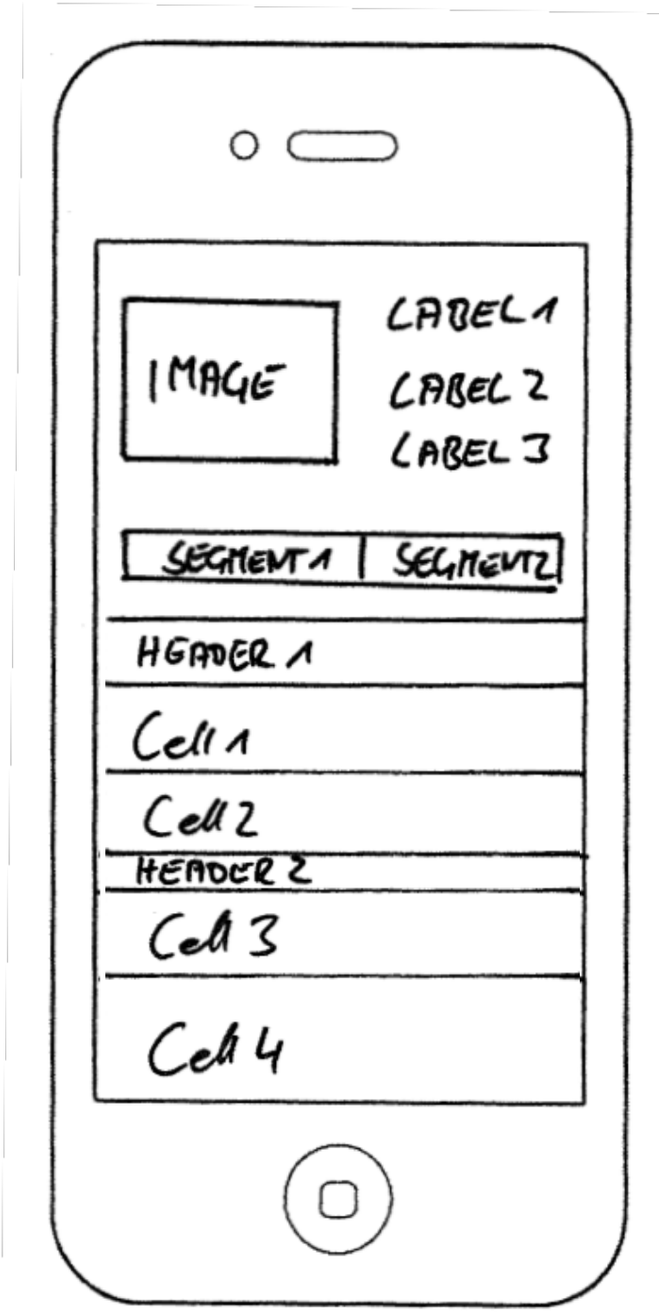
Separate Layouts for Portrait and Landscape



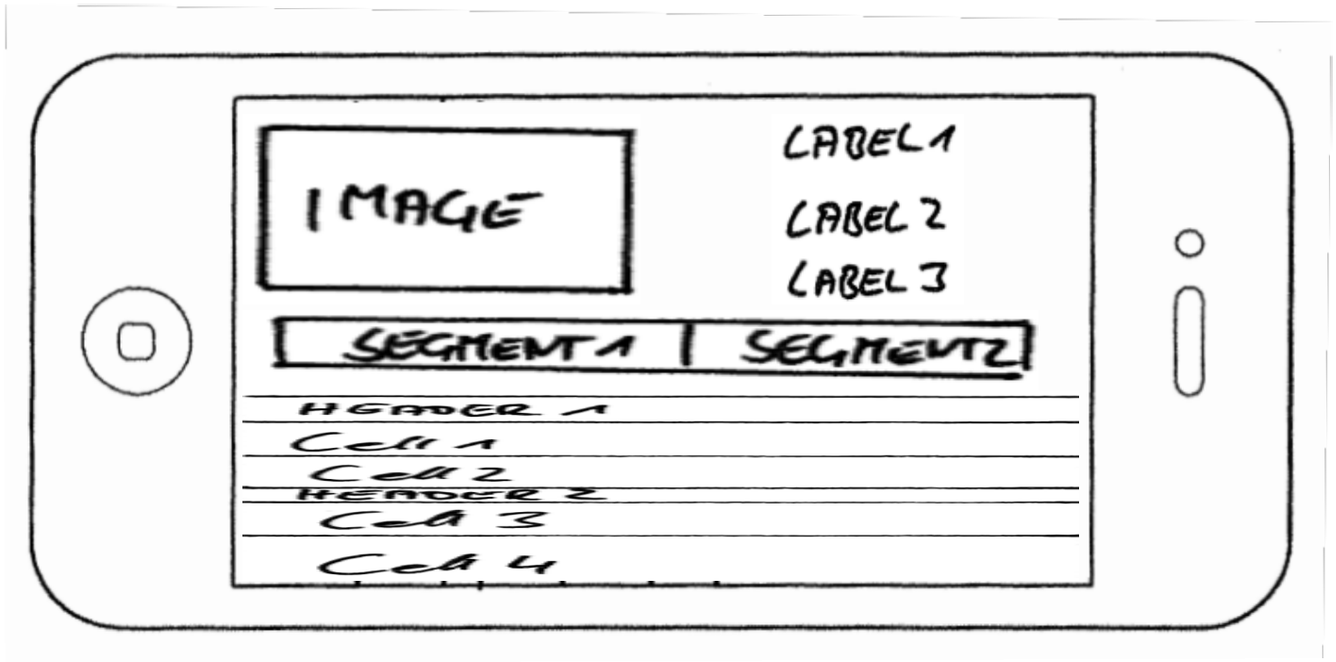
Rotation - Desired Behaviour



Rotation - Desired Behaviour



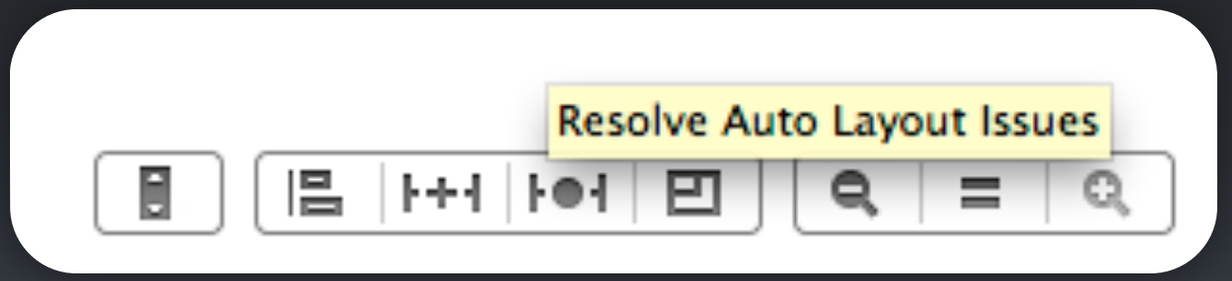
Rotation - Probable Behaviour



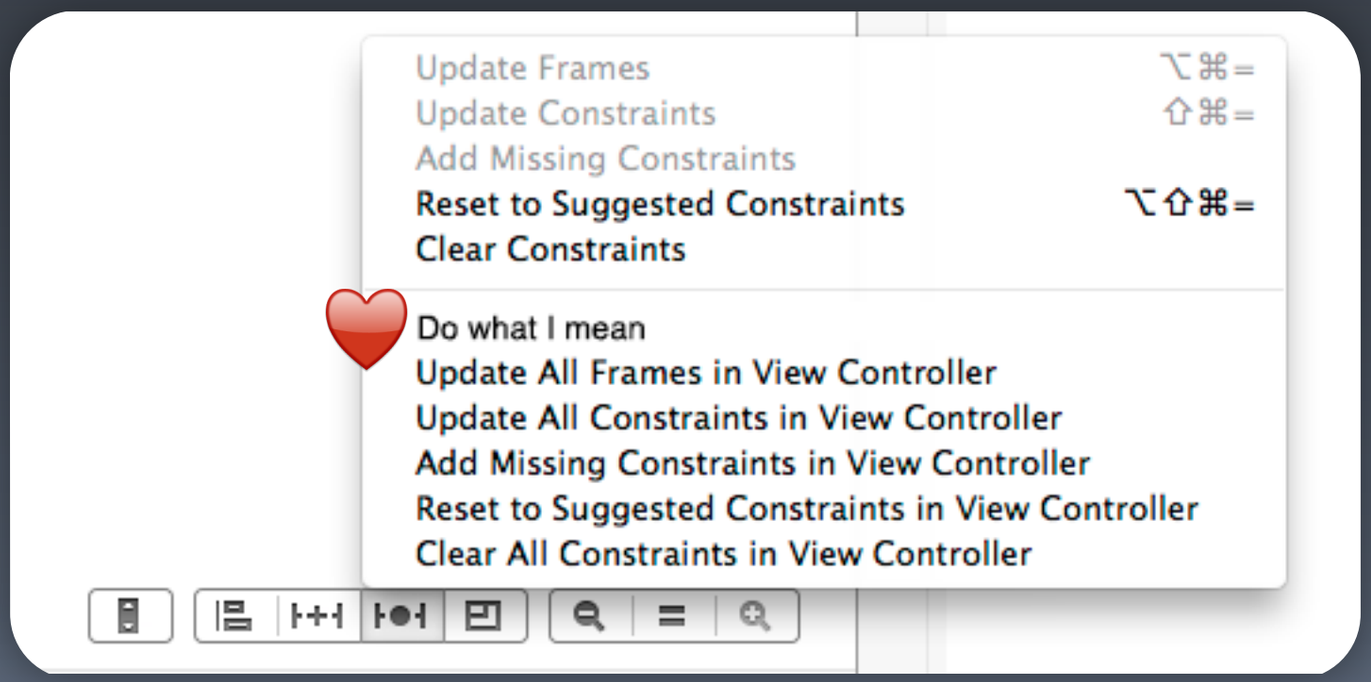
Rotation - Probable Behaviour

The Solution

1.



2.



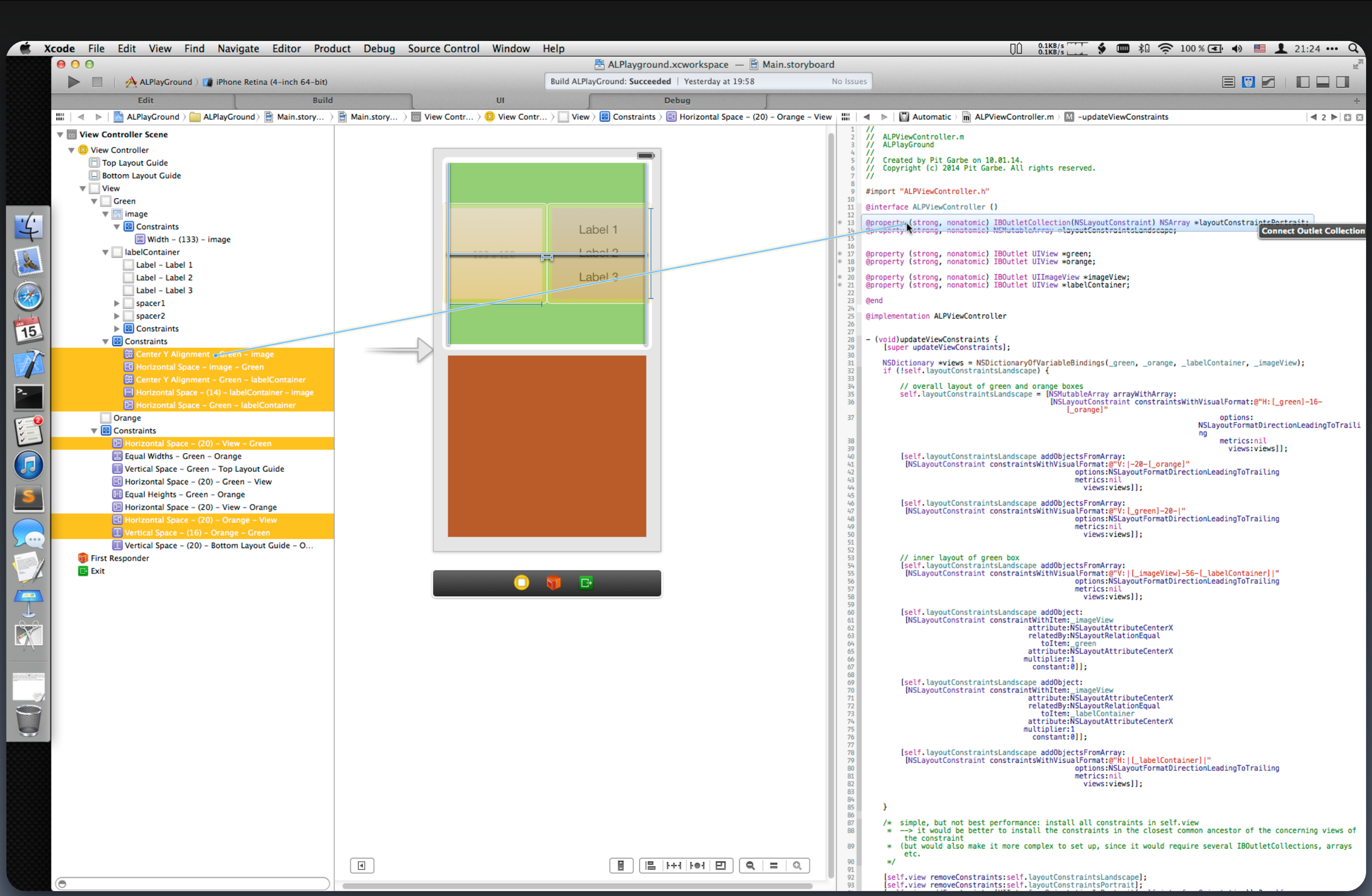
A (real) Solution

- 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

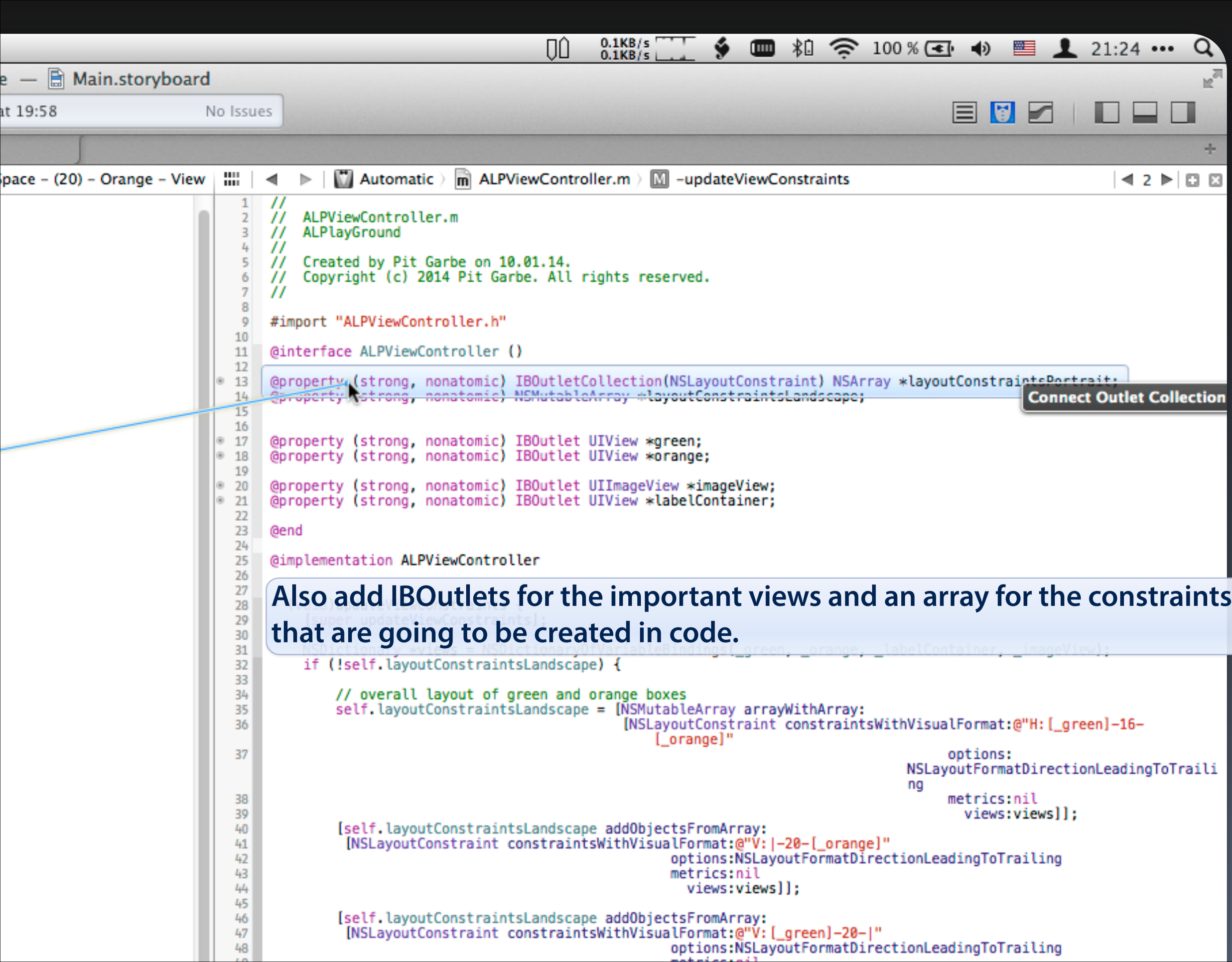
Select relevant constraints

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




```
//
//  ALPViewController.m
//  ALPlayGround
//
//  Created by Pit Garbe on 10.01.14.
//  Copyright (c) 2014 Pit Garbe. All rights reserved.
//

#import "ALPViewController.h"

@interface ALPViewController ()

@property (strong, nonatomic) IBOutletCollection(NSLayoutConstraint) NSArray *layoutConstraintsPortrait;
@property (strong, nonatomic) NSMutableArray *layoutConstraintsLandscape;


@property (strong, nonatomic) IBOutlet UIView *green;
@property (strong, nonatomic) IBOutlet UIView *orange;

@property (strong, nonatomic) IBOutlet UIImageView *imageView;
@property (strong, nonatomic) IBOutlet UIView *labelContainer;

@end

@implementation ALPViewController

- (void)updateViewConstraints {
    [super updateViewConstraints];

    NSDictionary *views = NSDictionaryOfVariableBindings(_green, _orange, _labelContainer, _imageView);

    if (!self.layoutConstraintsLandscape) {

        // overall layout of green and orange boxes
        self.layoutConstraintsLandscape = [NSMutableArray arrayWithArray:
                                           [NSLayoutConstraint constraintsWithVisualFormat:@"H:[_green]-16-[_orange]"
                                                                                   options:NSLayoutFormatDirectionLeadingToTrailing
                                                                                   metrics:nil
                                                                                     views:views]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"V:|-20-[_orange]"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"V:[_green]-20-|"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
```

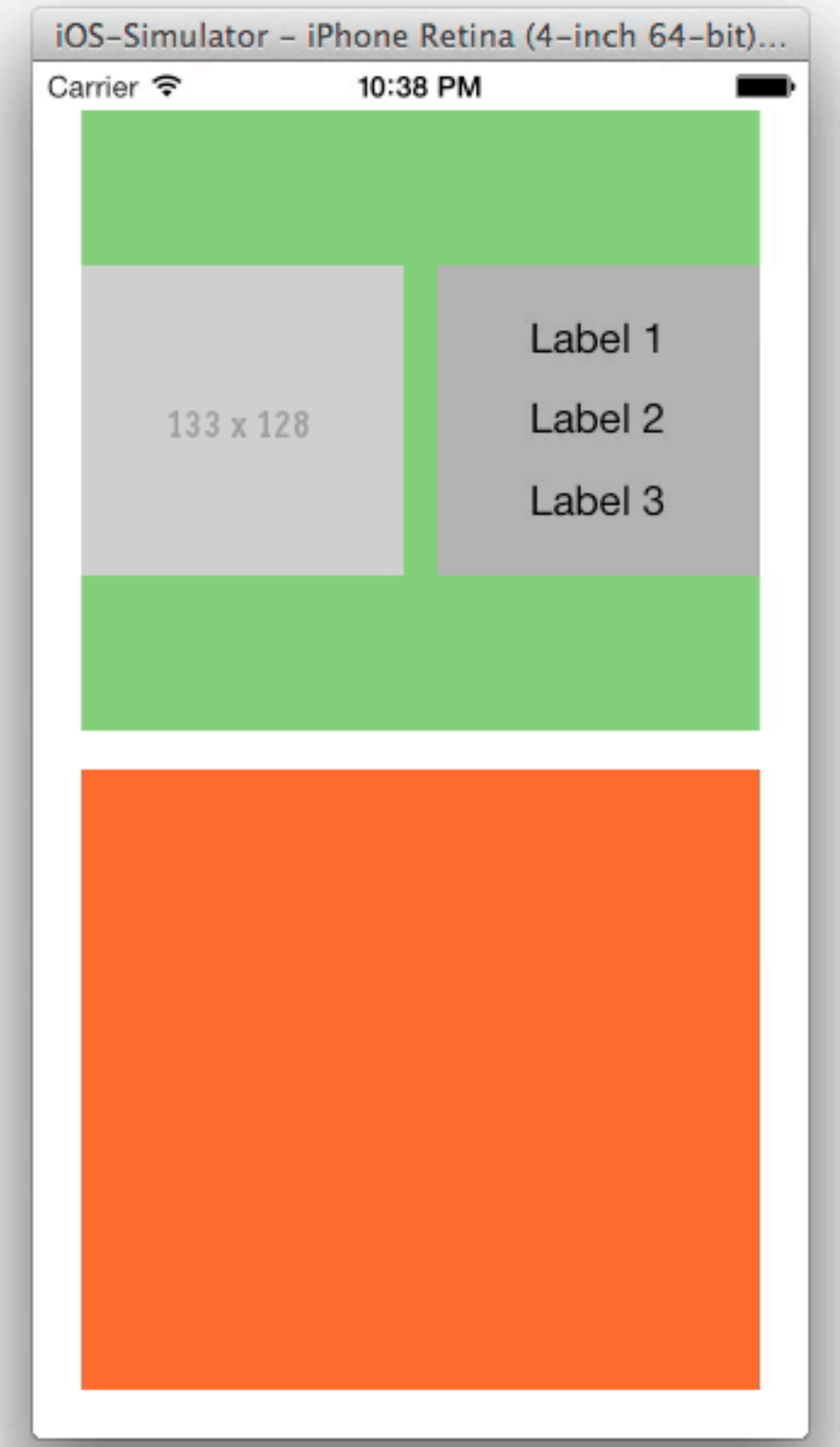
in **–updateViewConstraints** create all the constraints for Landscape orientation (once)

```
- (void)updateViewConstraints {
    [super updateViewConstraints];

    NSDictionary *views = NSDictionaryOfVariableBindings(_green, _orange, _labelContainer, _imageView);
    if (!self.layoutConstraintsLandscape) {

        // overall layout of green and orange boxes
        self.layoutConstraintsLandscape = [NSMutableArray arrayWithArray:
                                           [NSLayoutConstraint constraintsWithVisualFormat:@"H:[_green]-16-
                                               [_orange]"
                                                                                   options:
                                            NSLayoutFormatDirectionLeadingToTraili
                                            ng
                                                                                   metrics:nil
                                                                                     views:views]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"V:|-20-[_orange]"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"V:[_green]-20-|"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];


        // inner layout of green box
        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"V:|[_imageView]-56-[_labelContainer]|"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];

        [self.layoutConstraintsLandscape addObject:
         [NSLayoutConstraint constraintWithItem:_imageView
                                      attribute:NSLayoutAttributeCenterX
                                      relatedBy:NSLayoutRelationEqual
                                         toItem:_green
                                      attribute:NSLayoutAttributeCenterX
                                     multiplier:1
                                       constant:0]];

        [self.layoutConstraintsLandscape addObject:
         [NSLayoutConstraint constraintWithItem:_imageView
                                      attribute:NSLayoutAttributeCenterX
                                      relatedBy:NSLayoutRelationEqual
                                         toItem:_labelContainer
                                      attribute:NSLayoutAttributeCenterX
                                     multiplier:1
                                       constant:0]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[_labelContainer]|"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];


    }
```

```objc
        [self.layoutConstraintsLandscape addObject:
         [NSLayoutConstraint constraintWithItem:_imageView
                                      attribute:NSLayoutAttributeCenterX
                                      relatedBy:NSLayoutRelationEqual
                                         toItem:_green
                                      attribute:NSLayoutAttributeCenterX
                                     multiplier:1
                                       constant:0]];

        [self.layoutConstraintsLandscape addObject:
         [NSLayoutConstraint constraintWithItem:_imageView
                                      attribute:NSLayoutAttributeCenterX
                                      relatedBy:NSLayoutRelationEqual
                                         toItem:_labelContainer
                                      attribute:NSLayoutAttributeCenterX
                                     multiplier:1
                                       constant:0]];

        [self.layoutConstraintsLandscape addObjectsFromArray:
         [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[_labelContainer]|"
                                                 options:NSLayoutFormatDirectionLeadingToTrailing
                                                 metrics:nil
                                                   views:views]];


    }

    /*  simple, but not best performance: install all constraints in self.view
     *  --> it would be better to install the constraints in the closest common ancestor of the concerning views of
         the constraint
     *  (but would also make it more complex to set up, since it would require several IBOutletCollections, arrays
         etc.
     */

    [self.view removeConstraints:self.layoutConstraintsLandscape];
    [self.view removeConstraints:self.layoutConstraintsPortrait];
    [self.view addConstraints:(UIInterfaceOrientationIsPortrait(self.interfaceOrientation)) ? self.
        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


iOS-Simulator - iPhone Retina (4-inch 64-bit)...
Carrier
10:38 PM
133 x 128
Label 1
Label 2
Label 3

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