



iOS Auto Layout

Orlando Code Camp | April 2016

Others Talk,
We Listen.

2Who is CapTech

We are a national IT management consulting firm that bridges the gap between business & technology.



#5 Best Firms to Work For
#2 Information Technology
#4 Meeting Client Needs

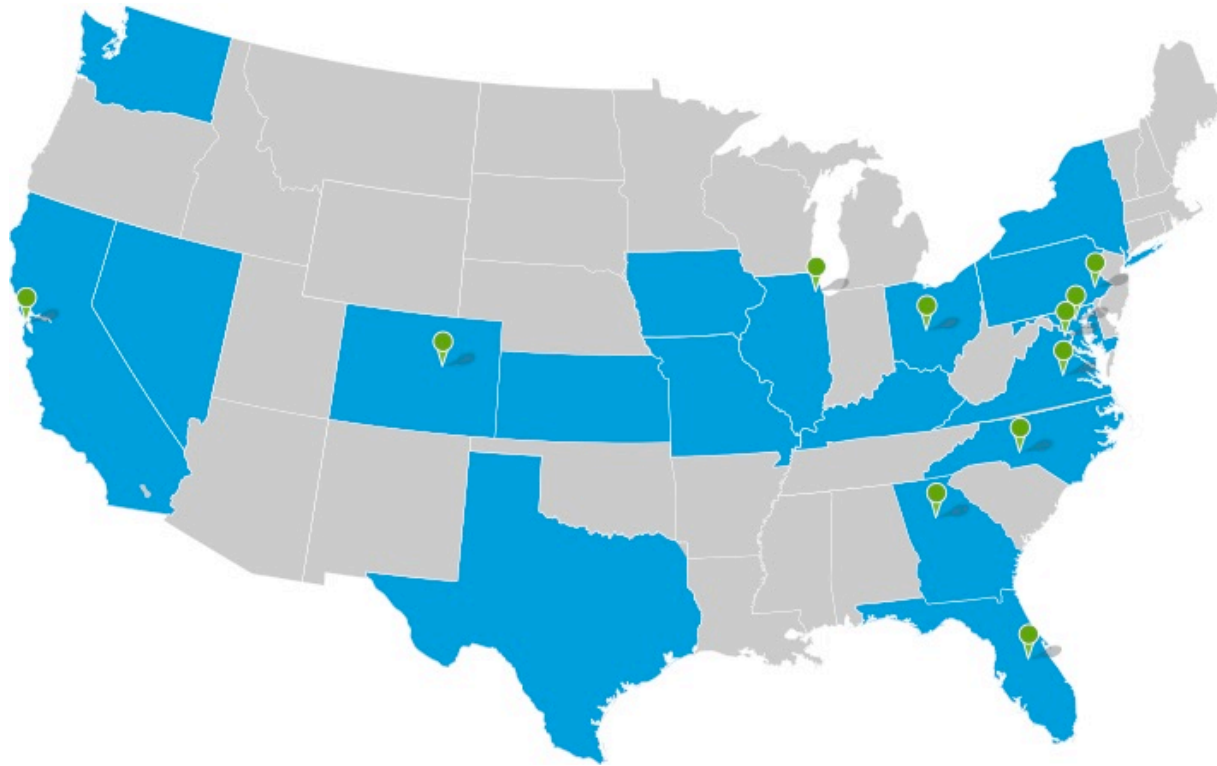


Ranked for 9 Years



Ranked on Vault's Consulting Top 50
#4 Best Consulting Firms for Hours
in the Office
#5 Best Consulting Firms for
Work/Life Balance

3Who is CapTech / National Consulting Firm



Locations

Atlanta, GA
Baltimore, MD
Charlotte, NC

Chicago, IL
Columbus, OH
Denver, CO

Orlando, FL
Philadelphia, PA
Richmond, VA

San Francisco, CA
Washington, D.C. Metro

4Services



Customer Experience

Digital Strategy
UX Research & Design
Integrated Marketing Management



Application Development

Mobile Design & Development
Secure Web Service & API Development
Back-End & Cloud Integration



Data & Analytics

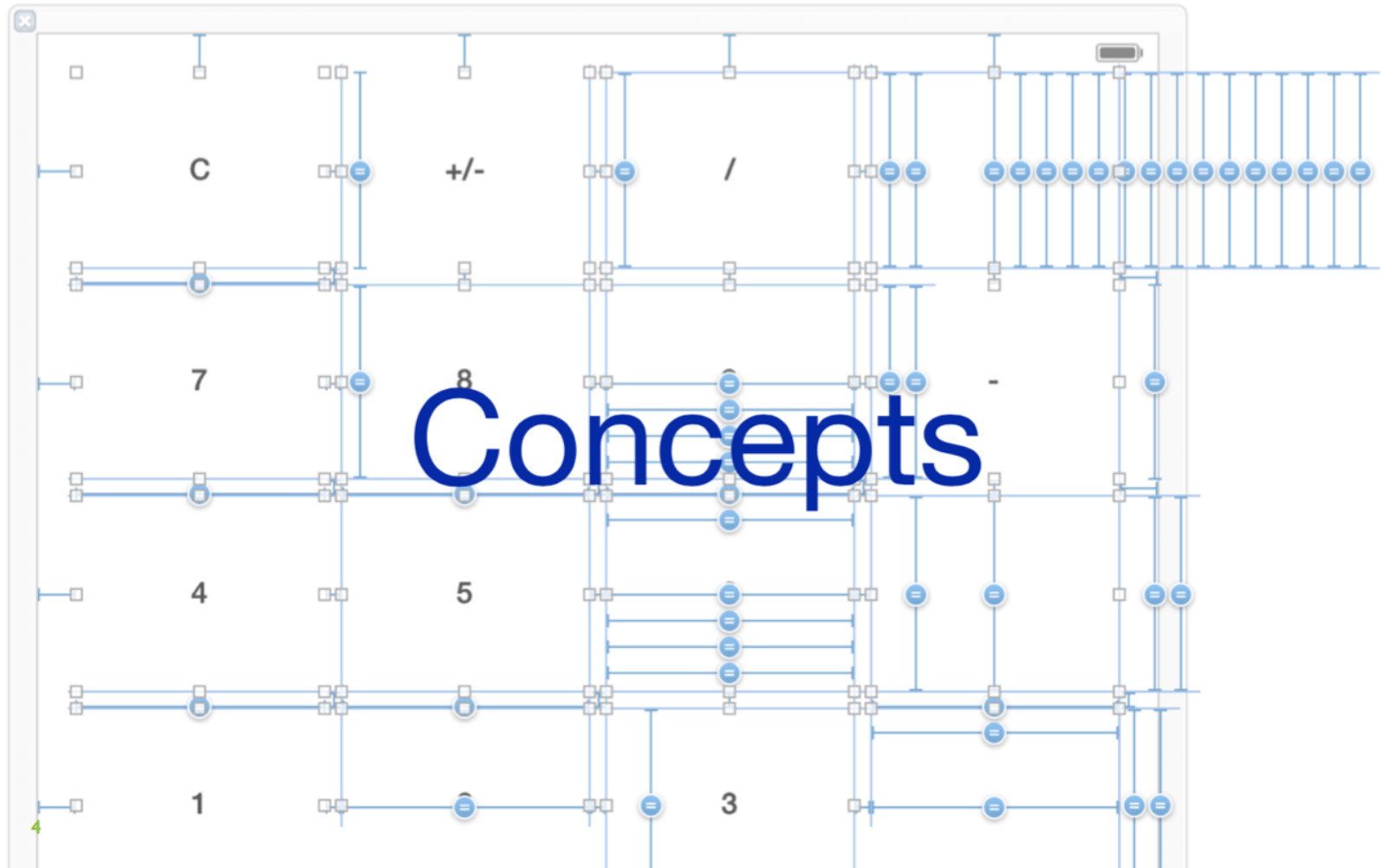
Big Data
Visualization & Reporting
Data Architecture
Data Engineering
Analytics & Data Science
Data Governance



Transformation

Agile Transformation & Coaching
Organizational Change Management
Program Governance & Strategy
Business Process Management
IT Service Management

5Auto Layout



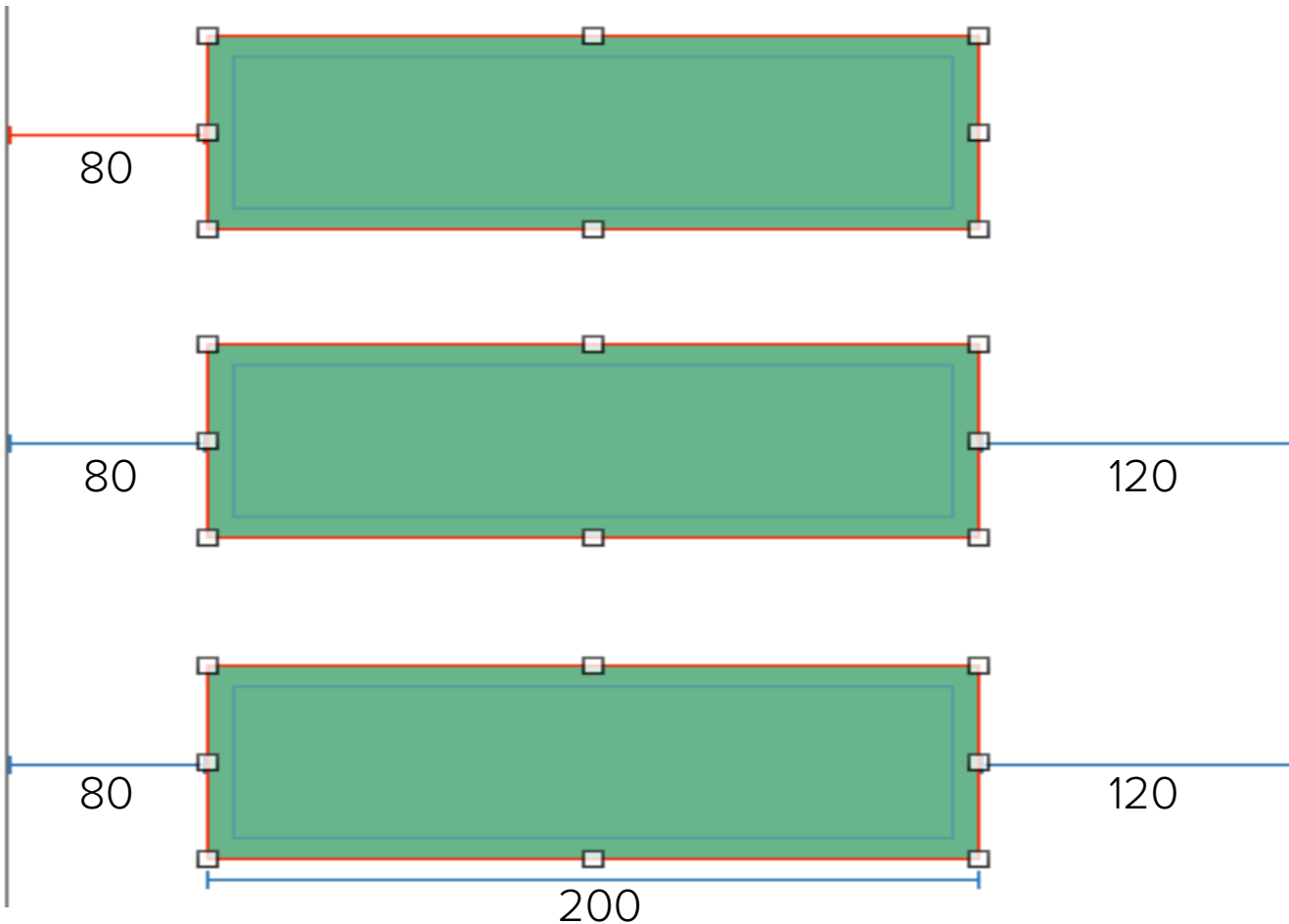
6What is Auto Layout?

Dynamic way of laying out views (size and position) based on relational constraints

Most common relations:

- To a view's superview
- To a sibling view

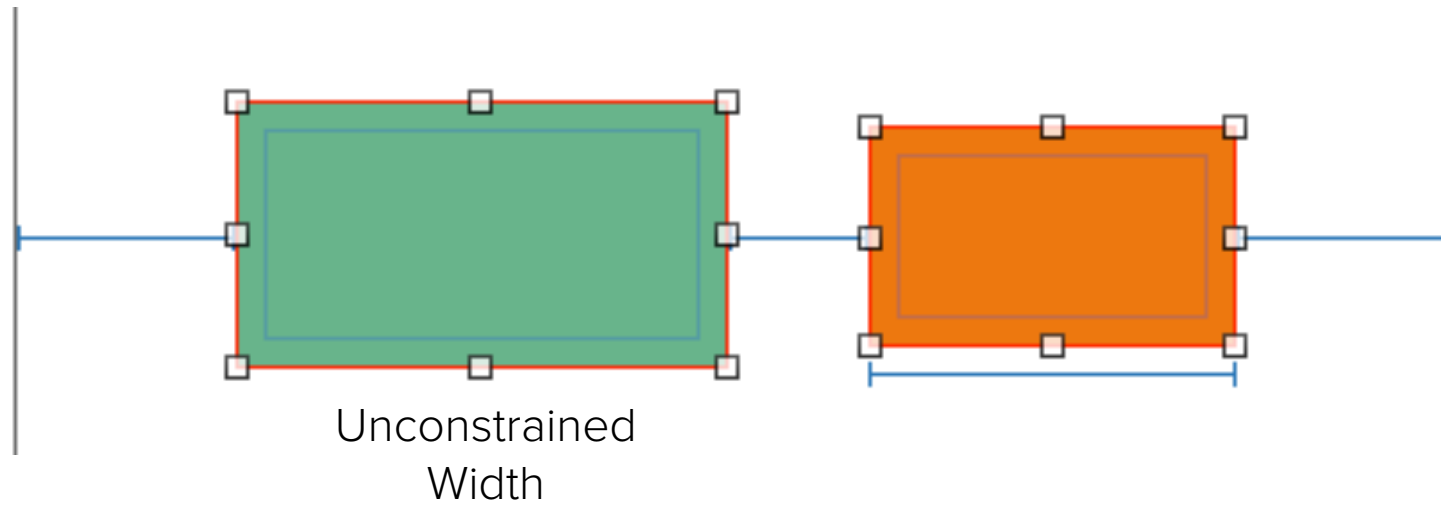
7Stretchability



Ambiguous
Under Constrained

Conflicting
Over Constrained

8Stretchability – Two Items



9 Constraint Properties

Horizontal Space Constraint

First Item

View.Leading

▼

Relation

Equal

⬆
⬇
⬆

Second Item

Superview.Leading Margin

▼

+

Constant

40

▼

⬆
⬇
⬆

Priority

1000

▼

⬆
⬇
⬆

Multiplier

1

▼

⬆
⬇
⬆

Identifier

Identifier

Placeholder

☐

Remove at build time

+

☐

Installed

10 Intrinsic Size

- How big the view wants to be
- Based on contents of view
- Child views can imbue their intrinsic size on parents
- Most standards controls have intrinsic size
- Views can define their own intrinsic size -
override func `intrinsicContentSize()` -> `CGSize`



11 Hugging and Resistance on a View

Hugging

- Higher priority – view won't grow past its intrinsic size

Compression Resistance

- Higher priority – view will be at least as big as its intrinsic size

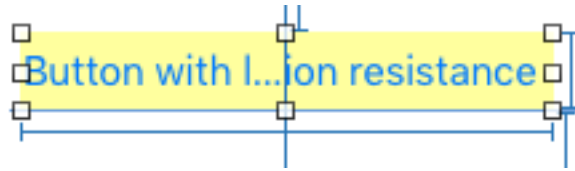
The image shows a settings panel with three sections. The first section, 'Content Hugging Priority', has two rows: 'Horizontal' with a value of 251 and 'Vertical' with a value of 251. The second section, 'Content Compression Resistance Priority', also has two rows: 'Horizontal' with a value of 750 and 'Vertical' with a value of 750. The third section, 'Intrinsic Size', has a single row with the value 'Default (System Defined)'. Each row consists of a text label, a numeric or text input field, and a small control icon (a downward arrow for the first two, and a blue square with a white double-headed arrow for the last).

Content Hugging Priority	
Horizontal	251
Vertical	251

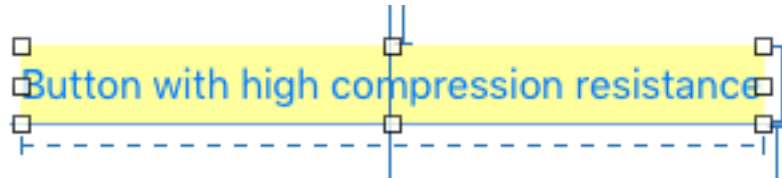
Content Compression Resistance Priority	
Horizontal	750
Vertical	750

Intrinsic Size	
	Default (System Defined)

12 Compression Resistance on a View

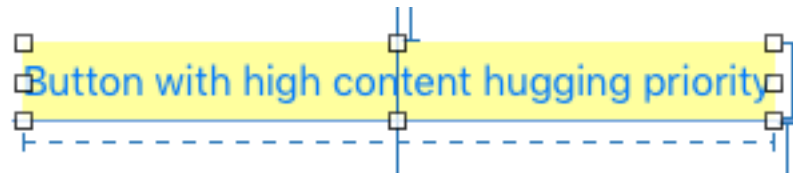


Button width constraint = 200, priority 1000,
UILabel compression resistance 750



Button width constraint = 200, priority 700,
UILabel compression resistance 750

13 Hugging Priority on a View

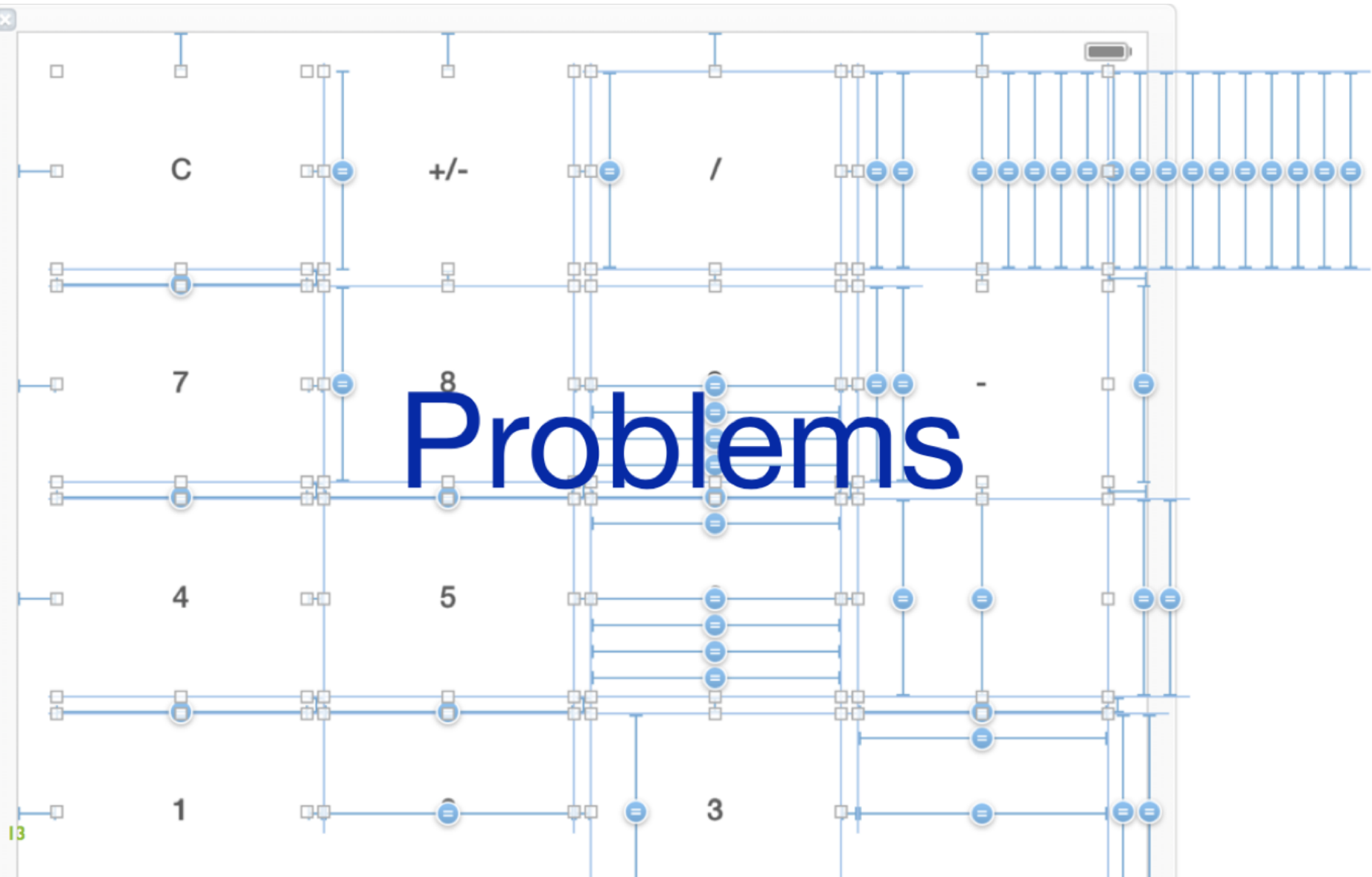


Button width constraint = 400, priority 500,
UILabel content hugging priority 750

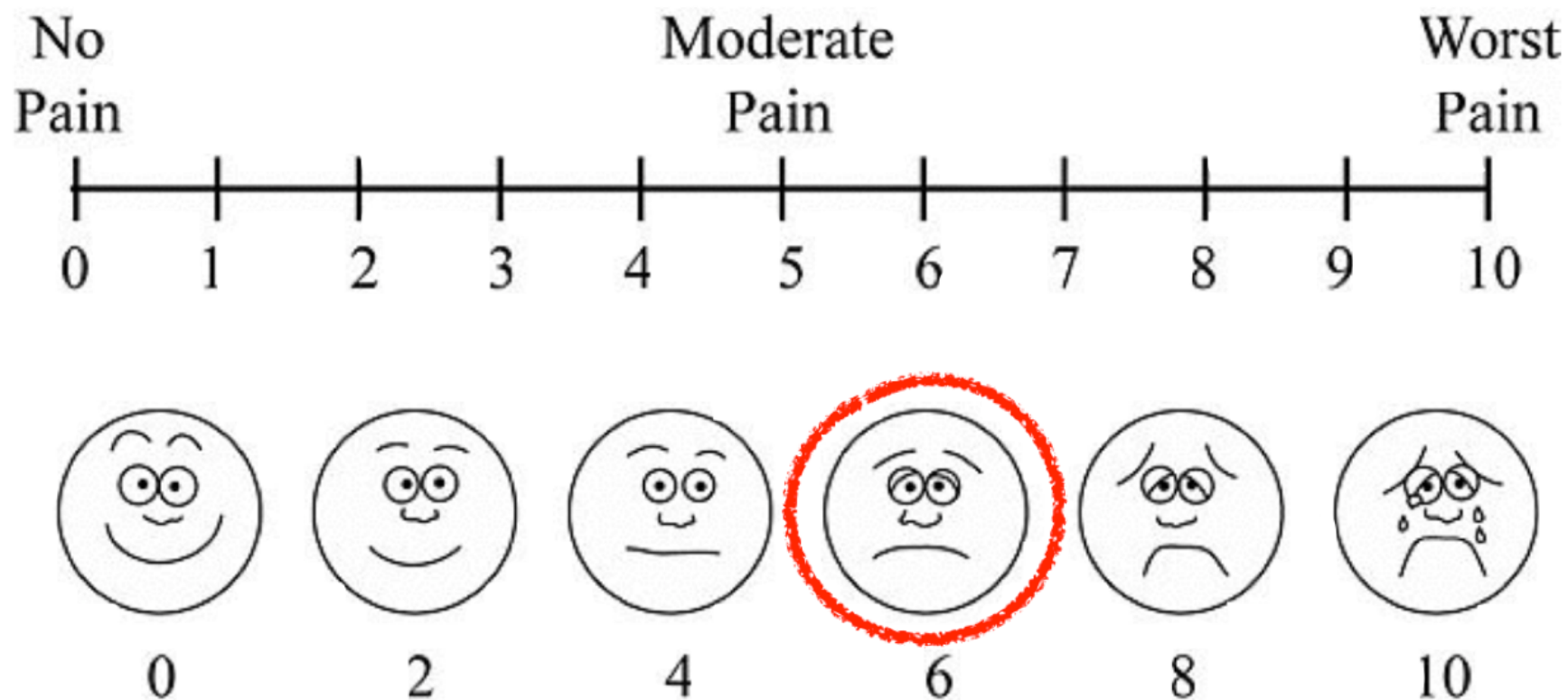


Button width constraint = 400, priority 1000,
UILabel content hugging priority 750

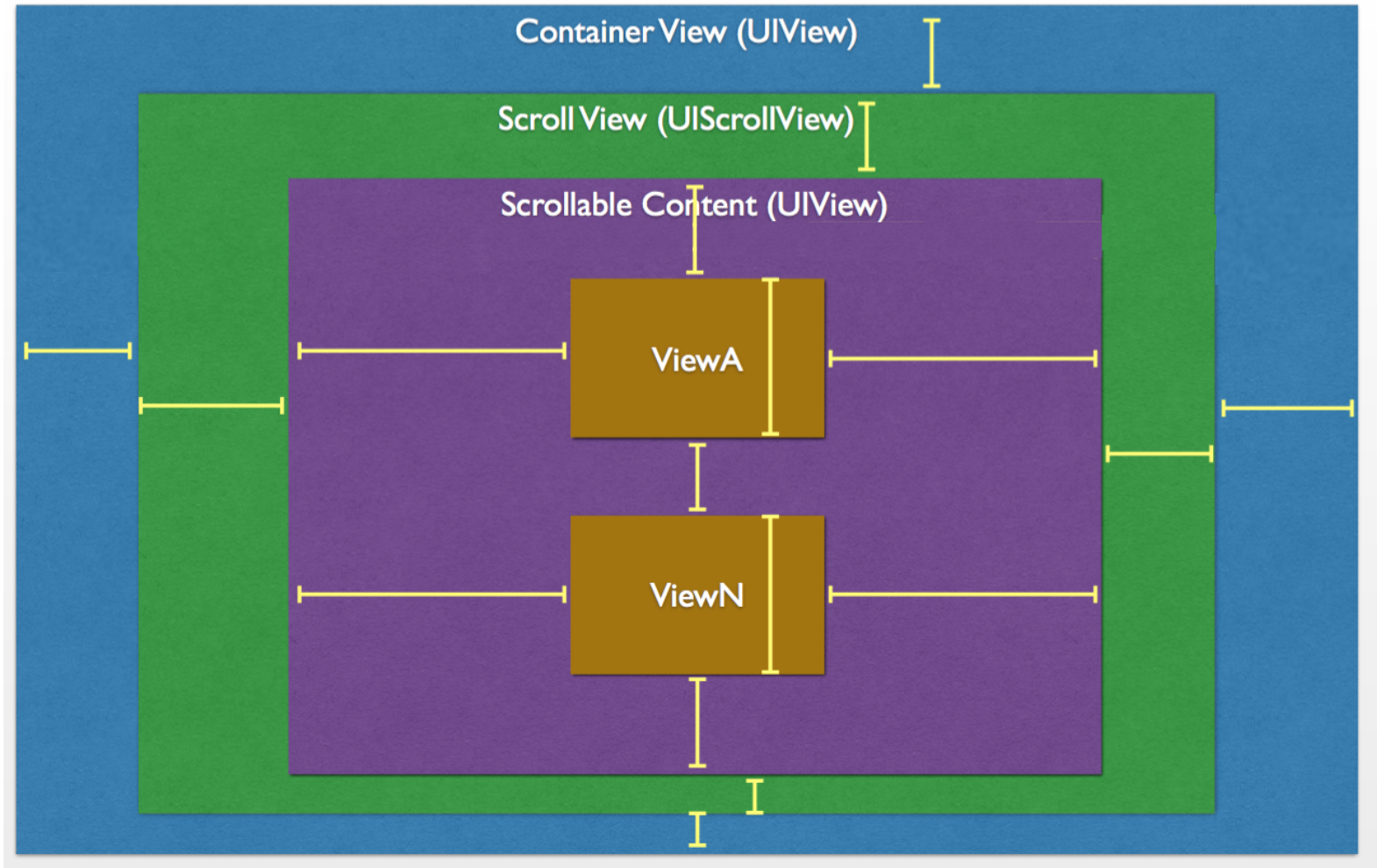
Problems



ScrollViews



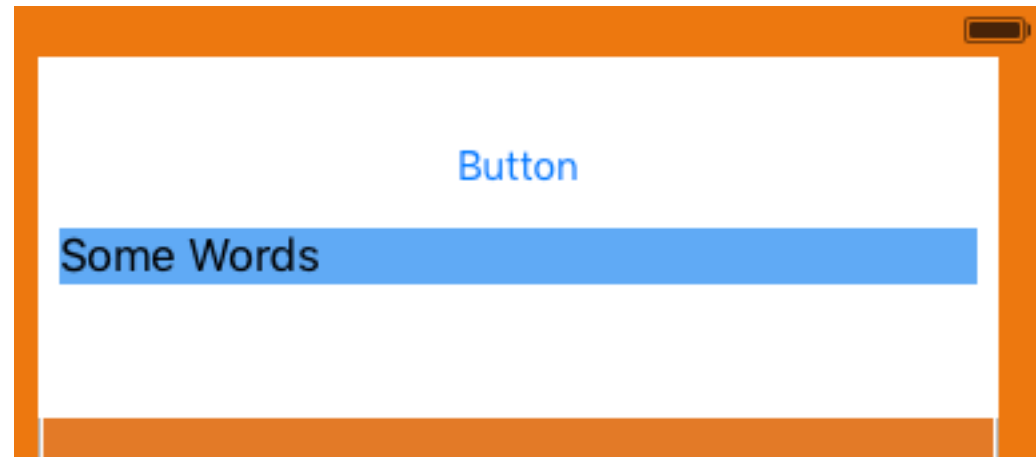
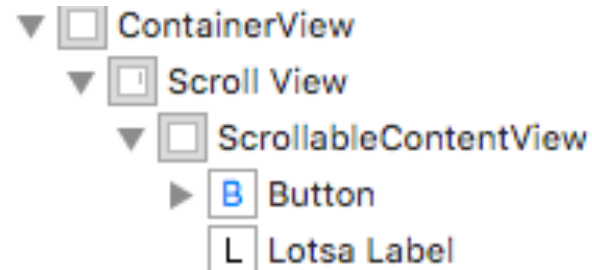
16ScrollView – view hierarchy



17 Auto Layout in ScrollViews

Goals

- Components resize
- No manual calculation of `contentSize`
- No conflicts of ambiguity



18Scroll – uh oh



19 ScrollView – possible issues

ScrollViews don't have

- Height
- Width
- Intrinsic size

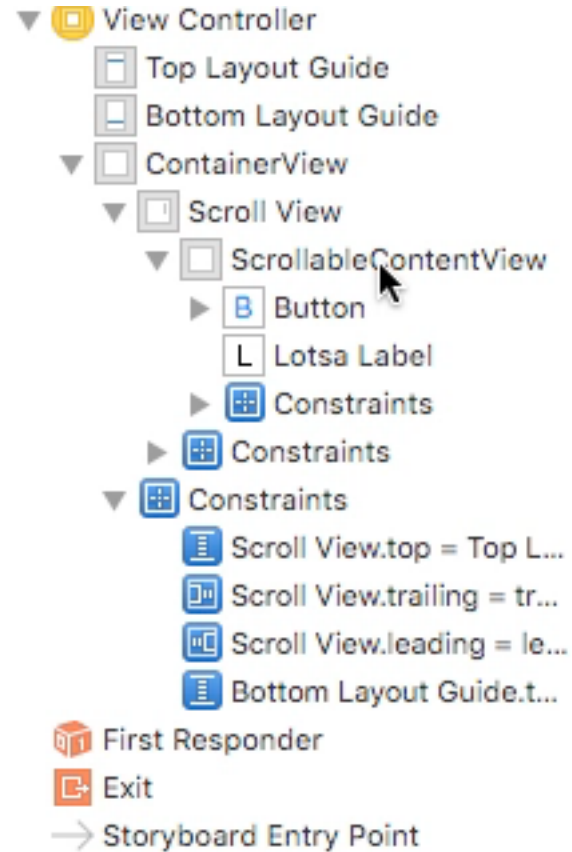
ScrollViews do have

- Content hugging
- Compression resistance

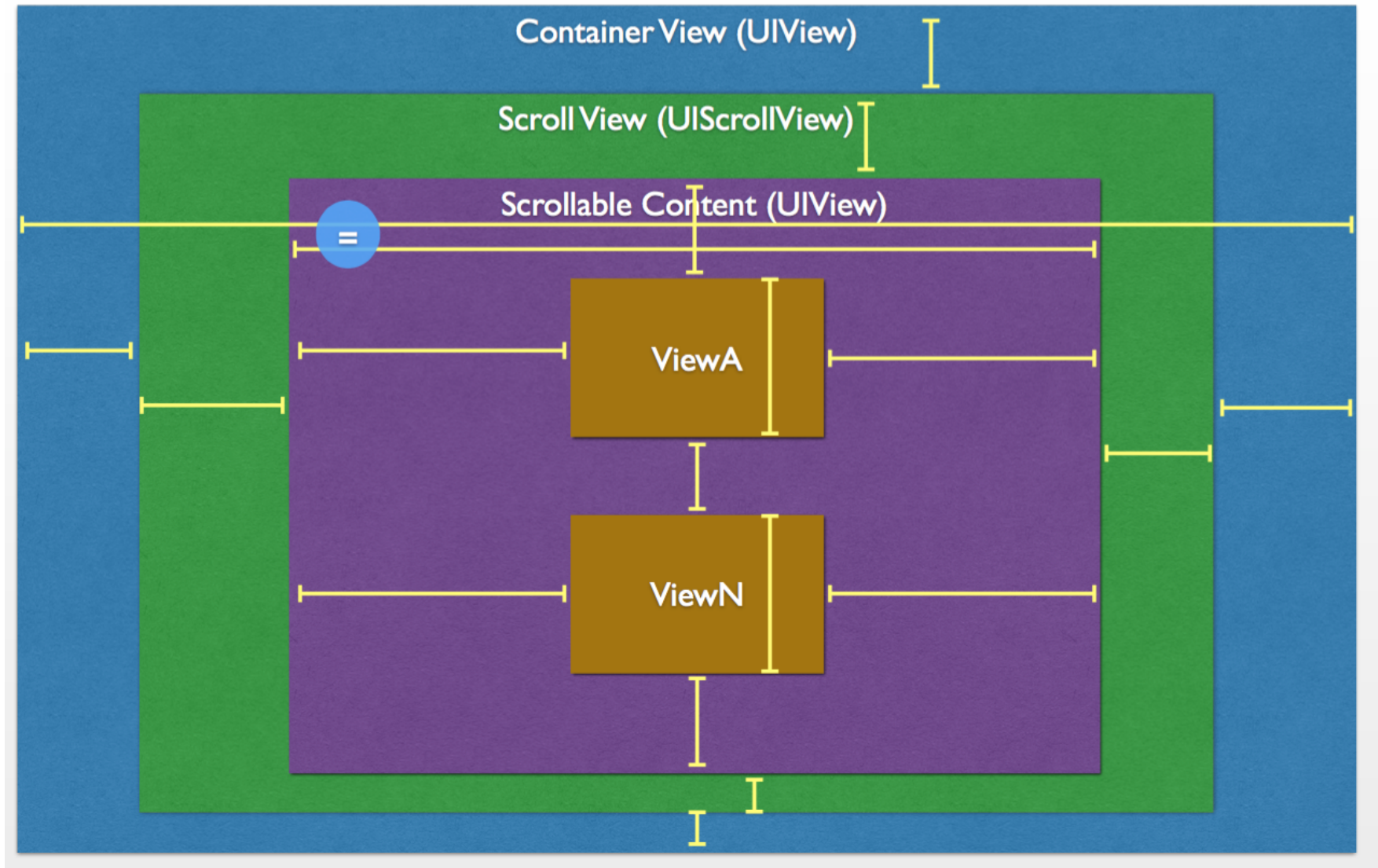
20 ScrollView – fix

Since the scroll view doesn't have a width, we set the `contentWidth = containerWidth`

Hold CTL and drag mouse



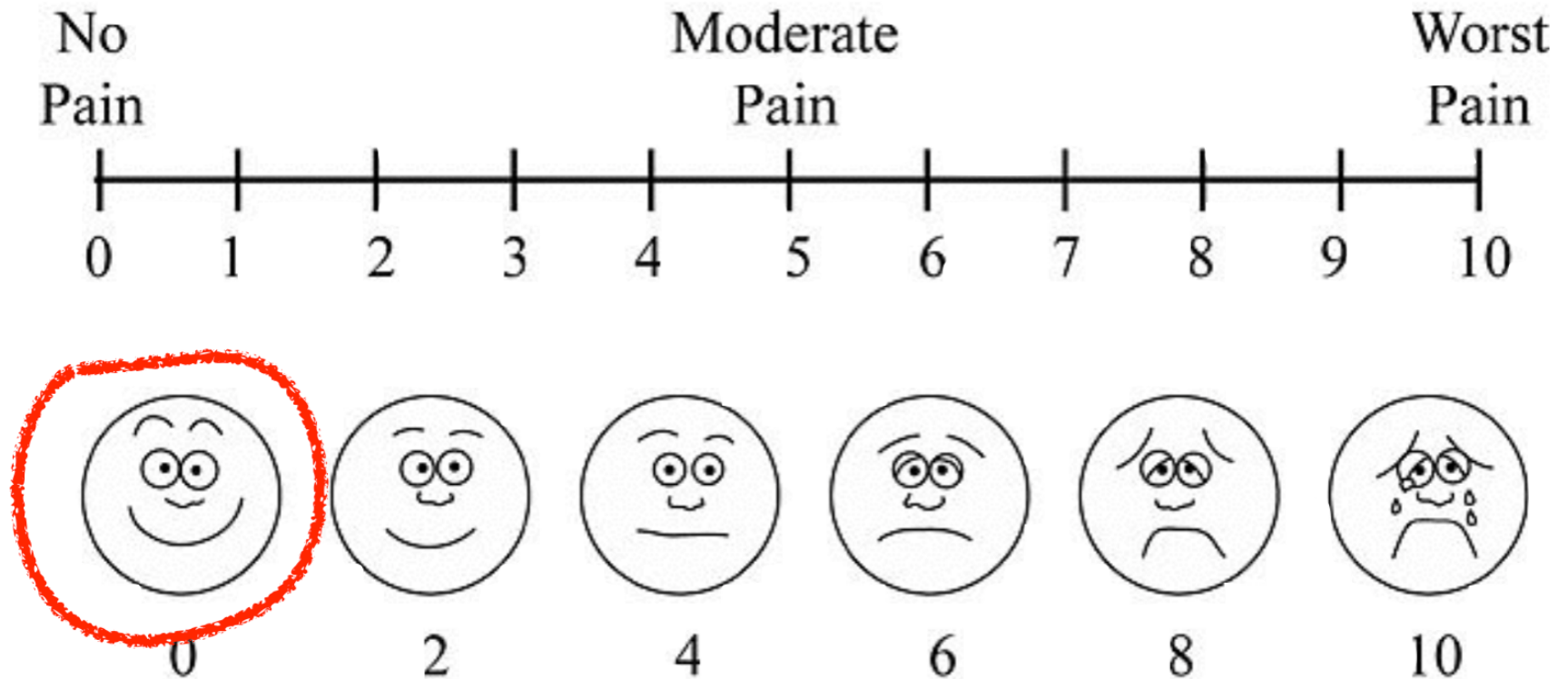
21ScrollView – solution



22Scroll – corrected



iOS 9 StackViews



- New iOS9
- Support Interesting & Complex Layouts
- No New Magic
- Horizontal or Vertical
- Easy to Add-Subtract Views from the Stack

Object Library



Horizontal Stack View - Arranges views linearly.



Vertical Stack View - Arranges views linearly.

25 UIStackView Example

Label 1

Label 2

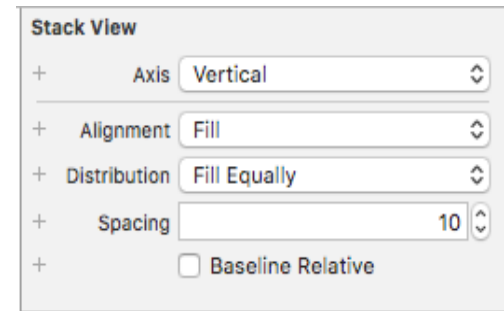
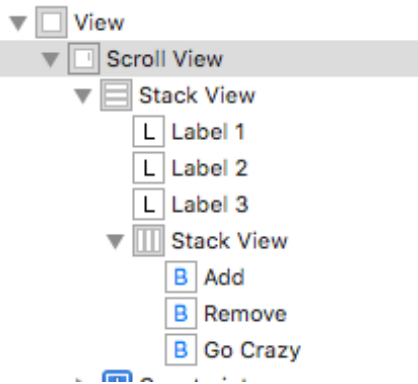
Label 3

Label 4

Add

Remove

Go Crazy



26UIStackView Example Code

```
@IBAction func addItem(sender: AnyObject) {
    let items = stackView.arrangedSubviews.count
    let label = UILabel()
    label.text = "Label \(items)"
    self.stackView.insertArrangedSubview(label, atIndex: items-1)
}

@IBAction func removeItem(sender: AnyObject) {
    let items = stackView.arrangedSubviews.count

    if (items > 1) { // the last item is the buttons stackView don't remove
        let viewToRemove = stackView.arrangedSubviews[items-2]
        self.stackView.removeArrangedSubview(viewToRemove)
        viewToRemove.removeFromSuperview()
    }
}

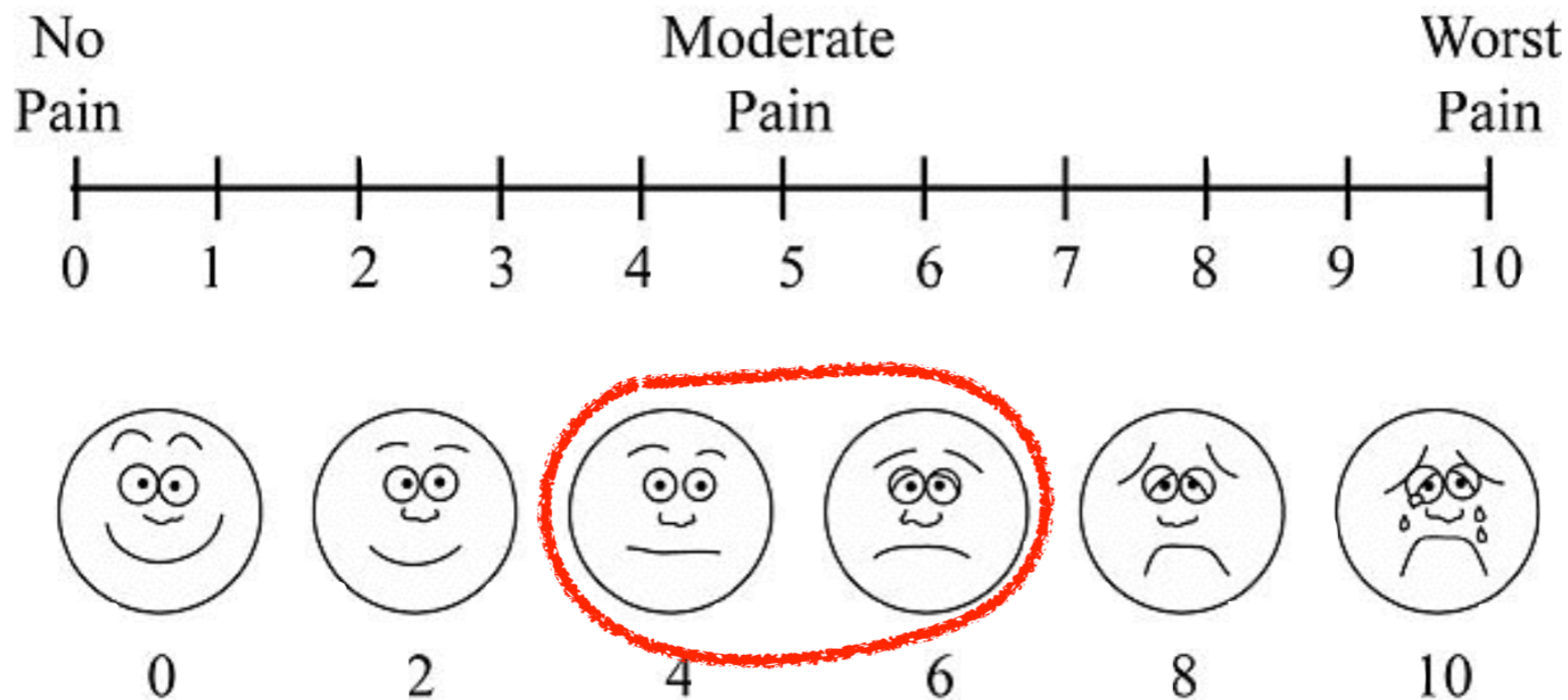
@IBAction func goCrazy(sender: AnyObject) {
    for _ in 1...8 {
        self.addItem(sender)
    }
}
```

27 `UIStackView` Warnings

- No Intrinsic Size
- Doesn't Replace TableViews
- Beware of Adding Constraints
- Removing Arranged Views



TableViews



29 Auto Layout in TableViews

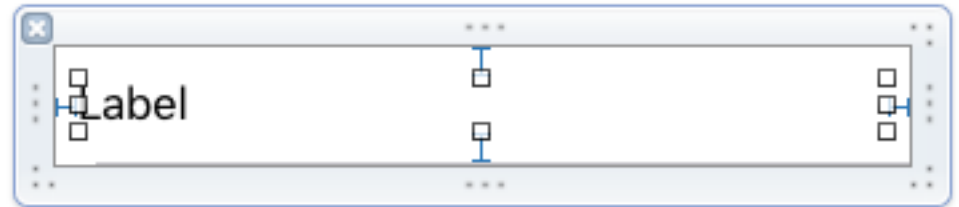
- Rows change size based on content
- Avoid pre-render cell contents to calculate size
- Build cell that is bi-directional
- In code, specify estimated row height
- In code, specify automatic dimension

```
tableView.estimatedRowHeight = 50.0  
tableView.rowHeight = UITableViewAutomaticDimension
```

30 Self-Sizing Cells

Basic Auto Layout TableViewCell

- UILabel
numberOfLines
property to 0

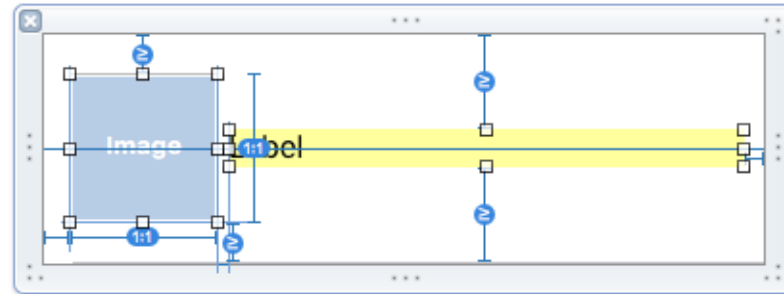


The quick brown fox jumps over the lazy dog,
catches the tiger by his tail and never lets it
go.

31Self-Sizing Cells

Complex Auto Layout TableViewCell

- Use \geq for constraints



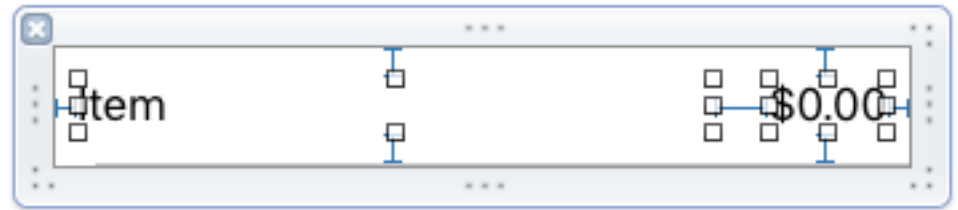
plum. sweet candy sweet
bears gummi biscuit
Gummi tart. candy candy
tart. pudding tart cotton
bears pudding lollipop.



tart candy cotton cotton biscuit
sugar Cake Gummi candy sugar
sweet sweet candy wafer wafer tart
candy cotton gummi cotton gummi
plum. sweet lollipop. candy gummi

Using Priorities Auto Layout TableViewCell

- Content Hugging Priority
- Compression Resistance Priority



SpongeBob Toy

\$19.99

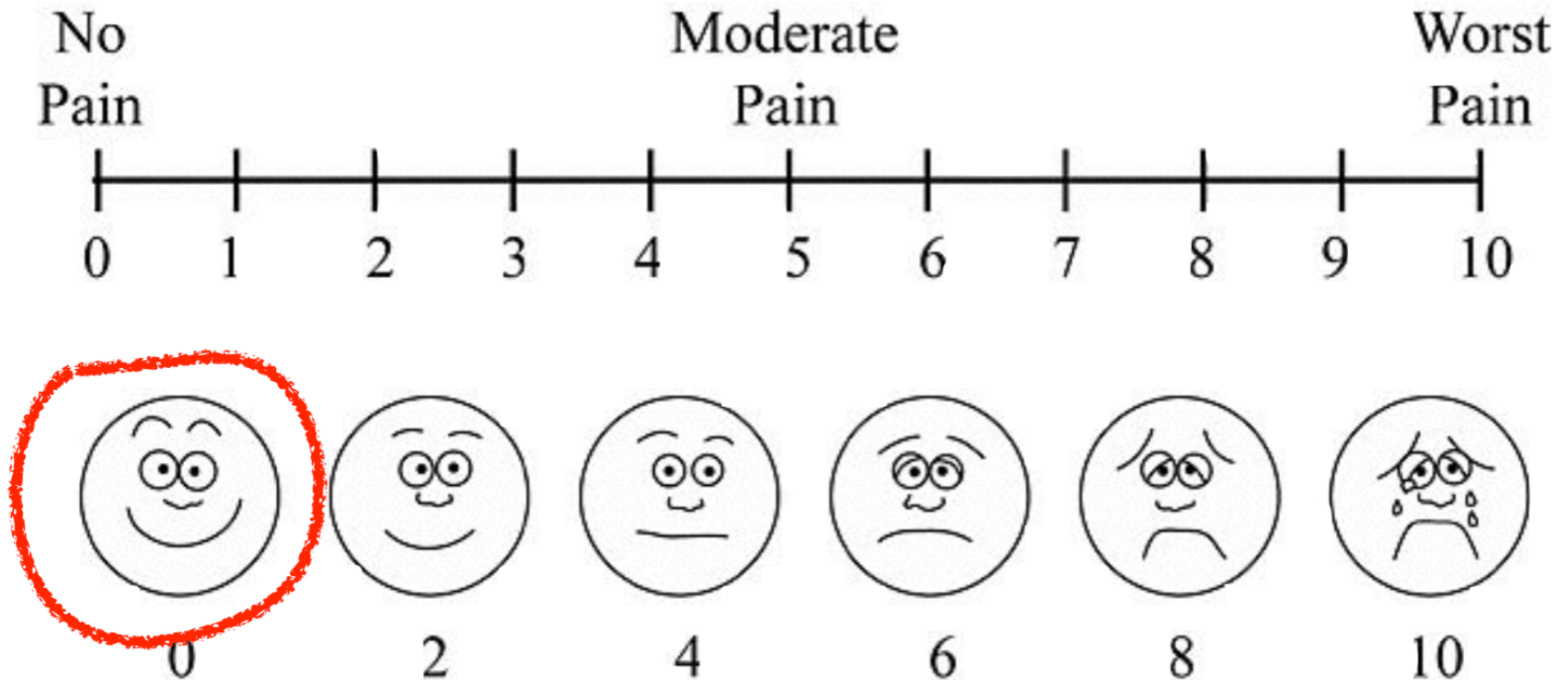
Microwave Oven, Chrome
1200W

\$649.99

1967 Ferrari Testarossa
330 P4

\$12,798,900

Animations With AutoLayout



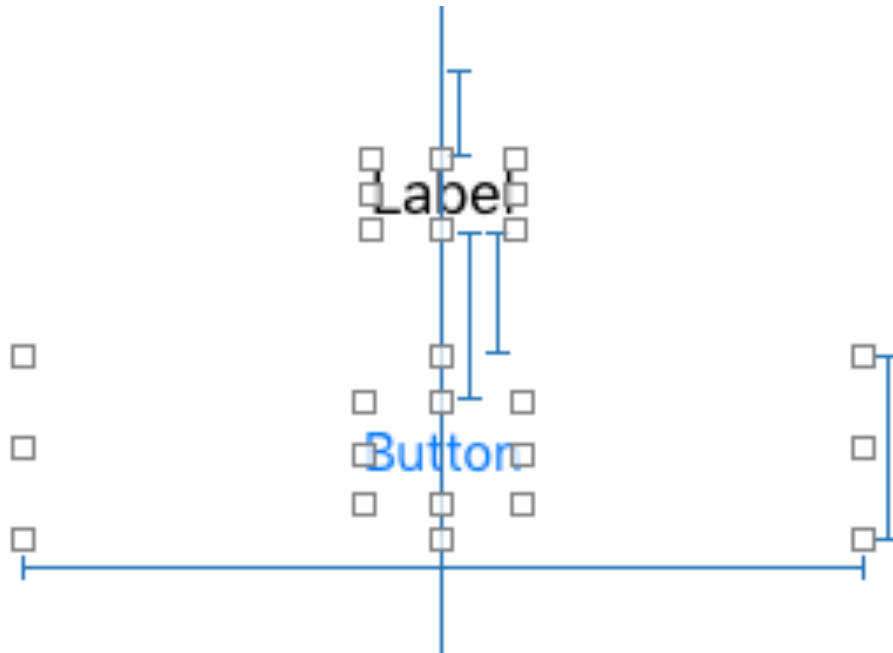
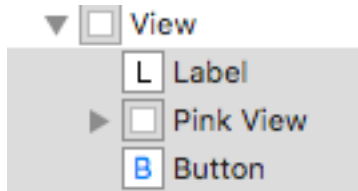
34 Animation Example

Label

Button



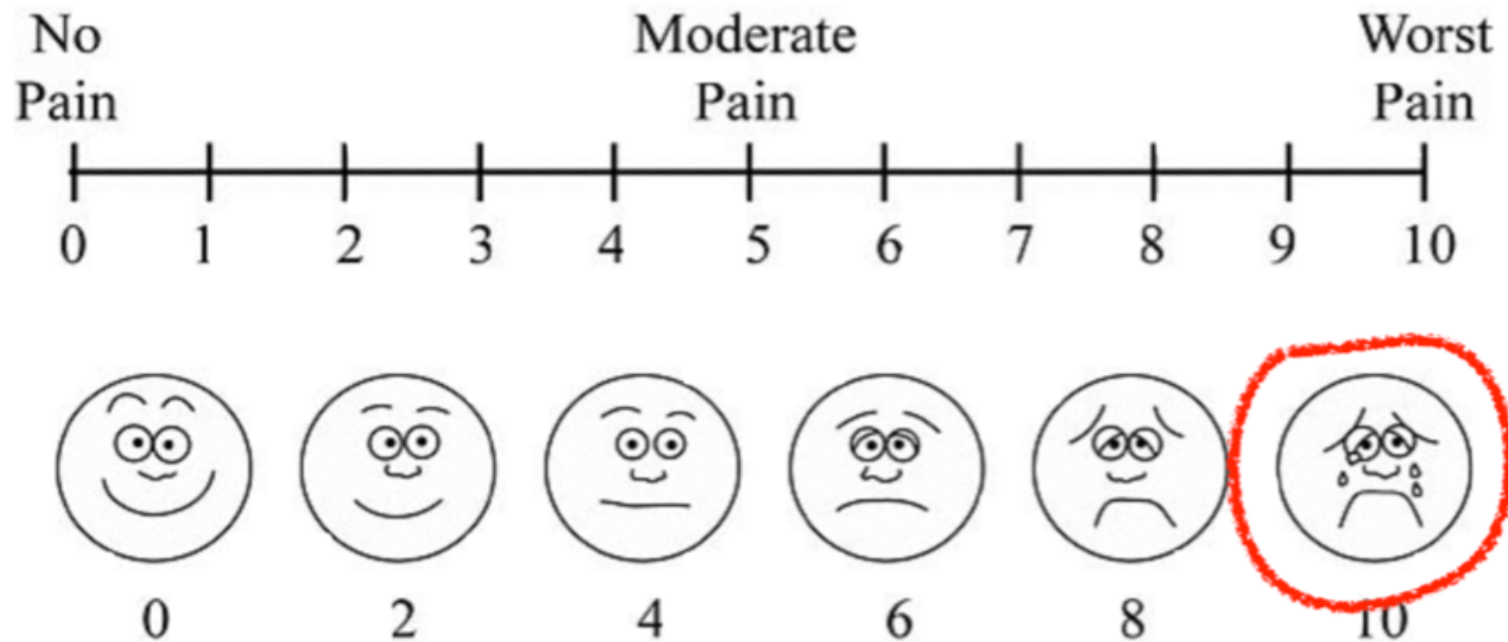
35 Setting up the ViewController



36Code for Animation

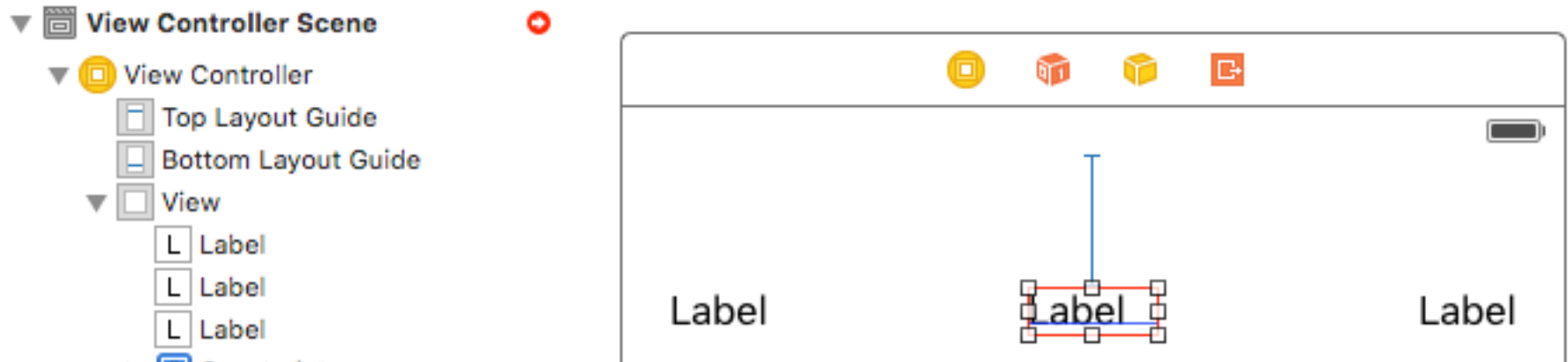
```
@IBAction func buttonPressed(sender: AnyObject) {
    dispatch_async(dispatch_get_main_queue(), {[weak self] () -> Void in
        UIView.animateWithDuration(0.75, animations: { () -> Void in
            self?.pinkView.alpha = 1.0
            self?.buttonTopConstraint.constant = 126
            //the next line allow the constraint to animate
            self?.button.layoutIfNeeded()
        }, completion: { (Bool) -> Void in
        })
    })
}
```

Debugging Auto Layout



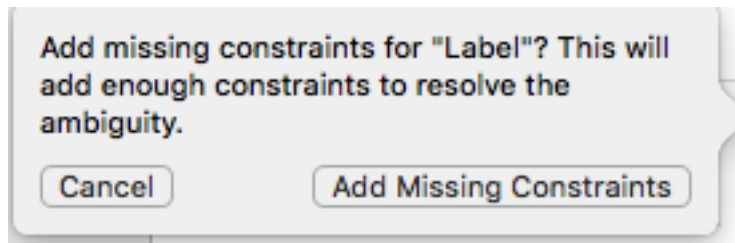
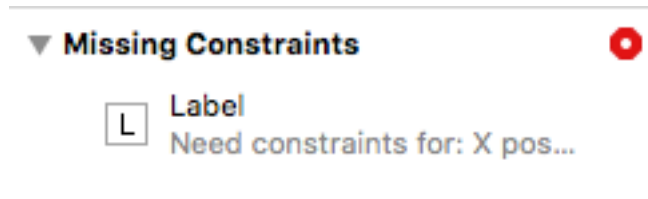
38 Deal with Ambiguity First

Under specification of constraints



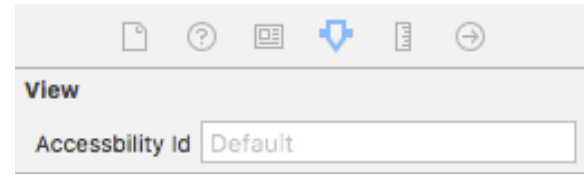
39 Auto-fixing Constraints

Recommendation is to fix your own constraint.



40 Steps to Debugging Ambiguity

- Give each view Accessibility Id
- Detect in XIB first
- Detect during run of app second
- 4 required constraints
 - X
 - Y
 - Height
 - Width



41 Debugging Ambiguity during Run of App

Label

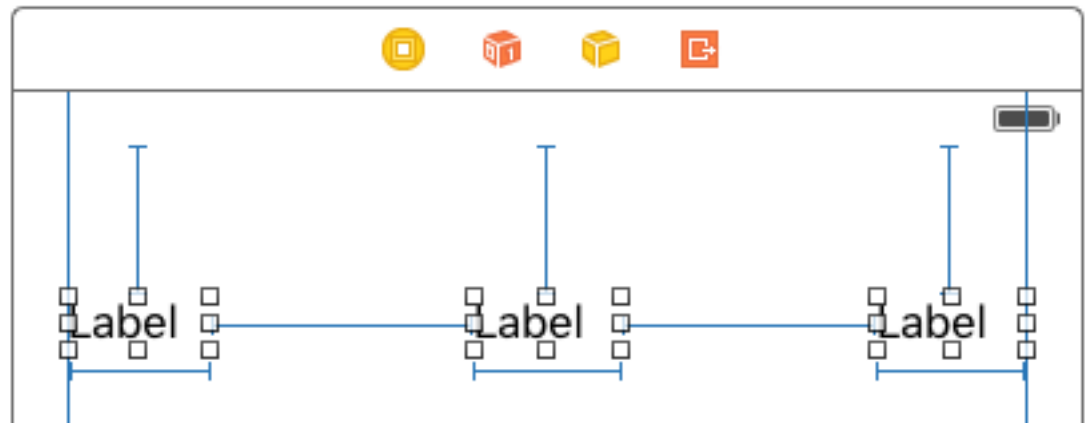
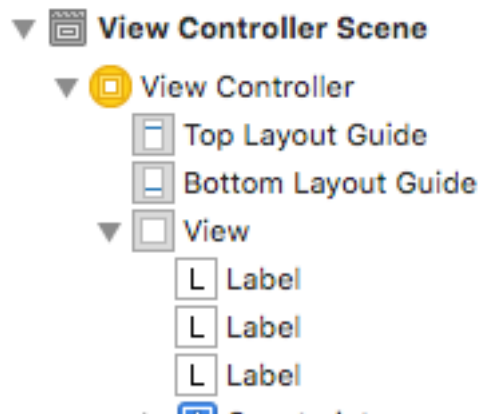
Label

```
(lldb) po [[UIWindow keyWindow] _autolayoutTrace]
UIWindow:0x78f63160
|   UIViewLayoutContainerView:0x7a4df230
|   |   UITransitionView:0x7a4e27a0
|   |   |   UIViewControllerWrapperView:0x78e60d50
|   |   |   |   UIViewLayoutContainerView:0x7a4dc600
|   |   |   |   |   UINavigationControllerTransitionView:0x7a2697f0
|   |   |   |   |   |   UIViewControllerWrapperView:0x78e69890
|   |   |   |   |   |   |   *UIView:0x78f5c290
|   |   |   |   |   |   |   |   *Label 1:0x78f73f50
|   |   |   |   |   |   |   |   *Label 3:0x78f740f0
|   |   |   |   |   |   |   |   *Label 2:0x78f74200- AMBIGUOUS LAYOUT for Label 2.Width{id: 166},
|   |   |   |   |   |   |   |   Label 2.Height{id: 169}
```

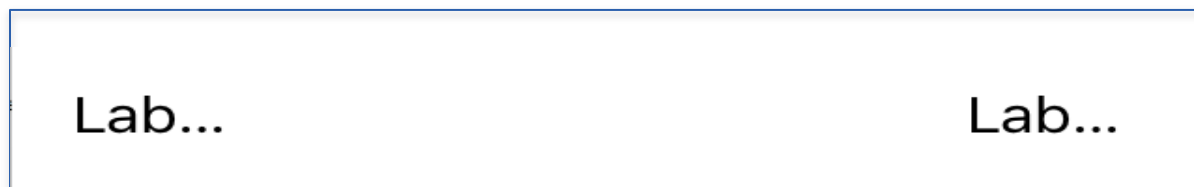
If there is more than one ambiguous constraint, start debugging with the highest ancestor.

42 Dealing with Conflicts

Over specification of constraints



Over specification of constraints



44 Dealing with Conflicts

In the console window we see:

Make a symbolic breakpoint at `UIViewAlertForUnsatisfiableConstraints` to catch this in the debugger. The methods in the `UIConstraintBasedLayoutDebugging` category on `UIView` listed in `<UIKit/UIView.h>` may also be helpful.

2016-03-22 17:16:21.927 ALProblems[12815:2204706] Unable to simultaneously satisfy constraints.

Probably at least one of the constraints in the following list is one you don't want.

Try this:

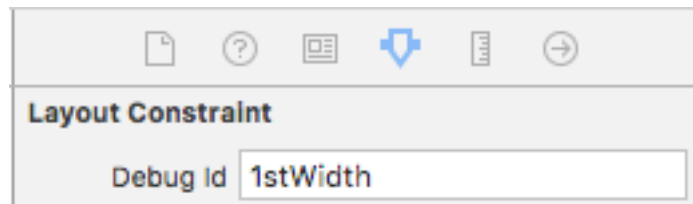
- (1) look at each constraint and try to figure out which you don't expect;
- (2) find the code that added the unwanted constraint or constraints and fix it.

```
(
    "<NSLayoutConstraint:0x7afb4c00 UILabel:0x7afb4140'Label'.leading == UIView:
0x7afb4080.leadingMargin>",
    "<NSLayoutConstraint:0x7afb4dc0 H:[UILabel:0x7afb4140'Label']-(110)-[UILabel:
0x7afb4440'Label']>",
    "<NSLayoutConstraint:0x7afb4df0 H:[UILabel:0x7afb4440'Label']-(110)-[UILabel:
0x7afb4330'Label']>",
    "<NSLayoutConstraint:0x7afb4e20 H:[UILabel:0x7afb4140'Label' (42)]>",
    "<NSLayoutConstraint:0x7afb4e80 H:[UILabel:0x7afb4440'Label' (56)]>",
    "<NSLayoutConstraint:0x7afb4c90 'TrailingMargin' UIView:0x7afb4080.trailingMargin == UILabel:
0x7afb4330'Label'.trailing>",
    "<NSLayoutConstraint:0x7c242a00 'UIView-Encapsulated-Layout-Width' H:[UIView:0x7afb4080(320)]>"
)
```

Will attempt to recover by breaking constraint

```
<NSLayoutConstraint:0x7afb4e80 H:[UILabel:0x7afb4440'Label' (56)]>
```

45 Dealing with Conflicts



By giving each view an Accessibility Id and each constraint a Debug Id we have:

Make a symbolic breakpoint at `UIViewAlertForUnsatisfiableConstraints` to catch this in the debugger. The methods in the `UIConstraintBasedLayoutDebugging` category on `UIView` listed in `<UIKit/UIView.h>` may also be helpful.

2016-03-22 17:13:51.664 ALProblems[12752:2184079] Unable to simultaneously satisfy constraints.
Probably at least one of the constraints in the following list is one you don't want.

Try this:

- (1) look at each constraint and try to figure out which you don't expect;
- (2) find the code that added the unwanted constraint or constraints and fix it.

```
(
    "<NSLayoutConstraint:0x7aa75540 '1stWidth' H:[Label 1(42)] (Names: Label 1:0x7aa5a510 )>",
    "<NSLayoutConstraint:0x7aa4c410 '2ndWidth' H:[Label 2(56)] (Names: Label 2:0x7aa74710 )>",
    "<NSLayoutConstraint:0x7aa72ef0 'LeadingMargin' Label 1.leading == UIView:0x7aa68040.leadingMargin (Names: Label 1:0x7aa5a510 )>",
    "<NSLayoutConstraint:0x7aa754e0 'spaceBetween1&2' H:[Label 1]-(110)-[Label 2] (Names: Label 2:0x7aa74710, Label 1:0x7aa5a510 )>",
    "<NSLayoutConstraint:0x7aa75510 'spaceBetween2&3' H:[Label 2]-(110)-[Label 3] (Names: Label 3:0x7aa73040, Label 2:0x7aa74710 )>",
    "<NSLayoutConstraint:0x7aa75030 'TrailingMargin' UIView:0x7aa68040.trailingMargin == Label 3.trailing (Names: Label 3:0x7aa73040 )>",
    "<NSLayoutConstraint:0x7a665ce0 'UIView-Encapsulated-Layout-Width' H:[UIView:0x7aa68040(320)]>"
)
```

Will attempt to recover by breaking constraint

```
<NSLayoutConstraint:0x7aa4c410 '2ndWidth' H:[Label 2(56)] (Names: Label 2:0x7aa74710 )>
```

- Identify static size vs. unconstrained view to allow stretching
- Some views have intrinsic size based on their content
- Make use of priorities, hugging, and compression resistance
- ScrollViews require extra care
- StackViews can simplify spacing numerous views
- Use UITableViewCell auto layout by specifying in the code the estimated height and automatic dimension
- Be systematic in debugging both ambiguity and conflicting constraints

47 Additional Resources

Auto Layout

<https://www.capttechconsulting.com/blogs/ios-7-tutorial-series-auto-layout-in-xcode-5>

Multi-tasking

<https://www.capttechconsulting.com/blogs/ios-9-tutorial-series-multi-tasking-with-adaptive-ui>

Size-Classes

<http://www.capttechconsulting.com/blogs/ios-9-tutorial-series---size-classes-preparing-your-apps-with-adaptive-ui>

CapTech Consulting: Web: CapTechConsulting.com

Twitter: [#CapTechListens](https://twitter.com/CapTechListens)

Contact: Eric Stroh Email: [strohtennis @ gmail](mailto:strohtennis@gmail.com)

Twitter: [#strohtennis](https://twitter.com/strohtennis)

Web: strohtennis.com

Github: github.com/strohtennis/ESAutoLayout