AUTO LAYOUT IS A UNICORN…

…WILD AND FREE.

MAGIK AWAITS, COME AND SEE.

Hi there folks!

-Art credit: Elinor, 5 years old.

ABOUT ME

SousChef

YOUR DIGITAL COOKING ASSISTANT

* 100K+ Users
* Mac, iPhone, and iPad versions
* 300K+ User Contributed Recipes
* Recently acquired by a UK Company

NICE MOHAWK

Ita

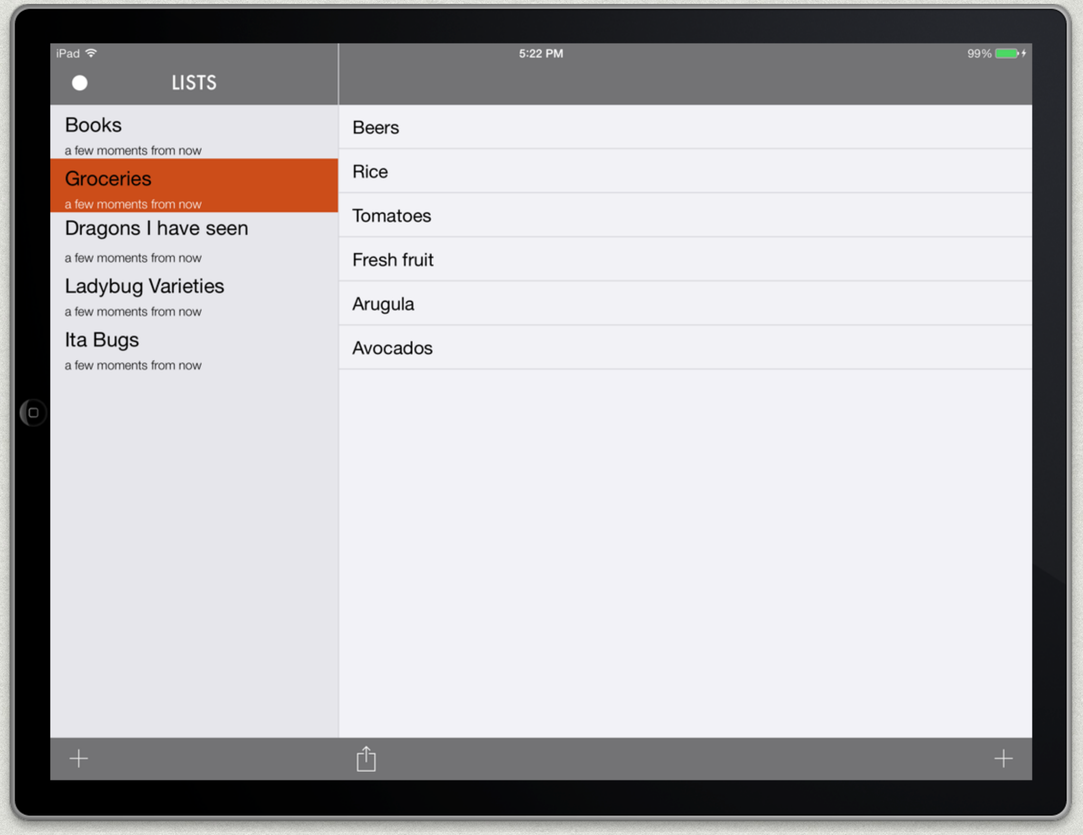
A FINE LIST-MAKING APP

* Universal iOS App
* iCloud Documents and Data
* Hit Top 100 in USA
* Recently had a 75K d/l day during promotion

Ita 2

A FINER LIST-MAKING APP

* iOS 7 re-envisioning
* Slick inline editing mode
* Built on Text Kit
* In-app settings
* UIDynamics



Reverb

A COMPLETELY NEW WAY TO DISCOVER NEWS

* SF Bay Area Client
* Interesting move into the News discovery market
* NML hired to do animations and social integrations, push notifications
* iPad only

THE FINE EDGE

nicemohawk.com/blog

AUTO LAYOUT

This is how it’s going to go.

I'm not really going to talk about the wiz-bang stuff.

We're mostly going to talk about the pain in the ass, kind of esoteric stuff.

- Auto Layout Basics
- Animation
- Debugging
- Advanced Stuff

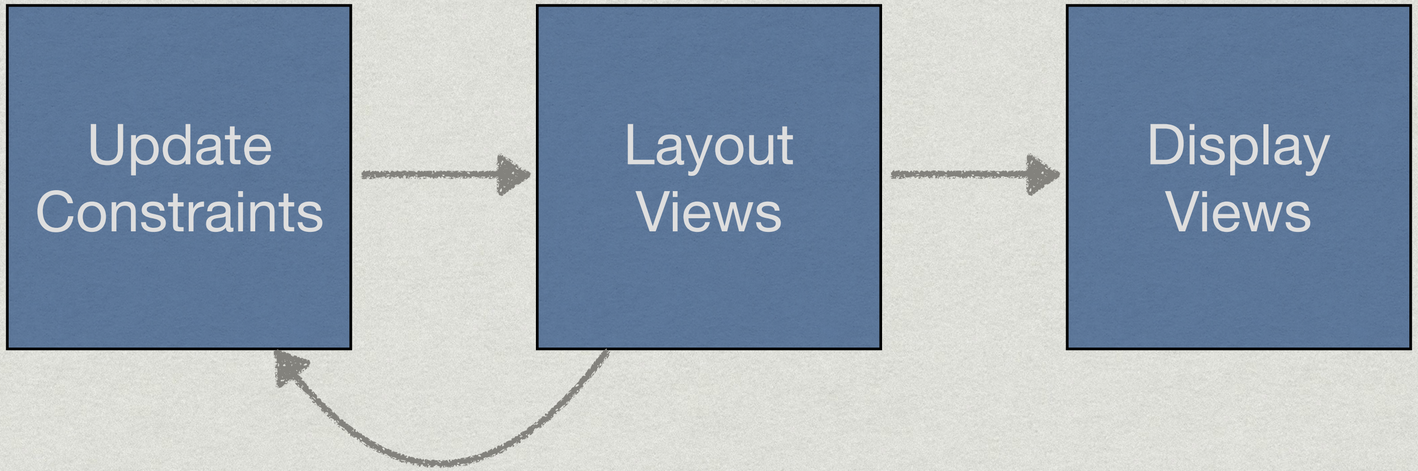
DISCLAIMER:
AUTO LAYOUT IS SOMETIMES HARD AND I’M IN IT WITH YOU, I'VE REVERTED AT LEAST A FEW PROJECTS TO FRAMES, SPRINGS AND STRUTS.

Auto Layout

- Auto Layout is sometimes referred to as constraint based layout system.
- You create a set on constraints and the Auto Layout Engine **solves** them.
  - This means you need an *unambiguous* set of constraints.
    - Which usually means two **or more** constraints for each axis.

DISCLAIMER:
AUTO LAYOUT IS SOMETIMES HARD AND I'M IN IT WITH YOU, I'VE REVERTED AT LEAST A FEW PROJECTS TO FRAMES, SPRINGS AND STRUTS.

The Architecture



The Architecture

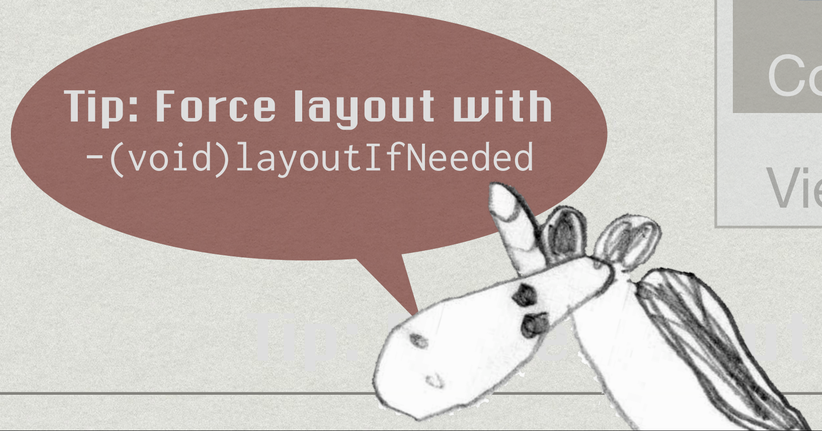
- Bottom up
  - Subview Constraints
  - Container View Constraints
  - View Controller Level Constraints
- Trigger with `-(void)setNeedsUpdateConstraints;`
- Override point:

```
- (void)updateViewConstraints;
// Overrides must call super or send -updateConstraints
   to the view.
```

The Architecture

- Top down
  - Layout View Controller's view
  - Layout Container View
  - Layout Subview
- Trigger with `-(void)setNeedsLayout;`
- Override point:

```
- (void)layoutSubviews;
// Overrides must call super
```



The Architecture

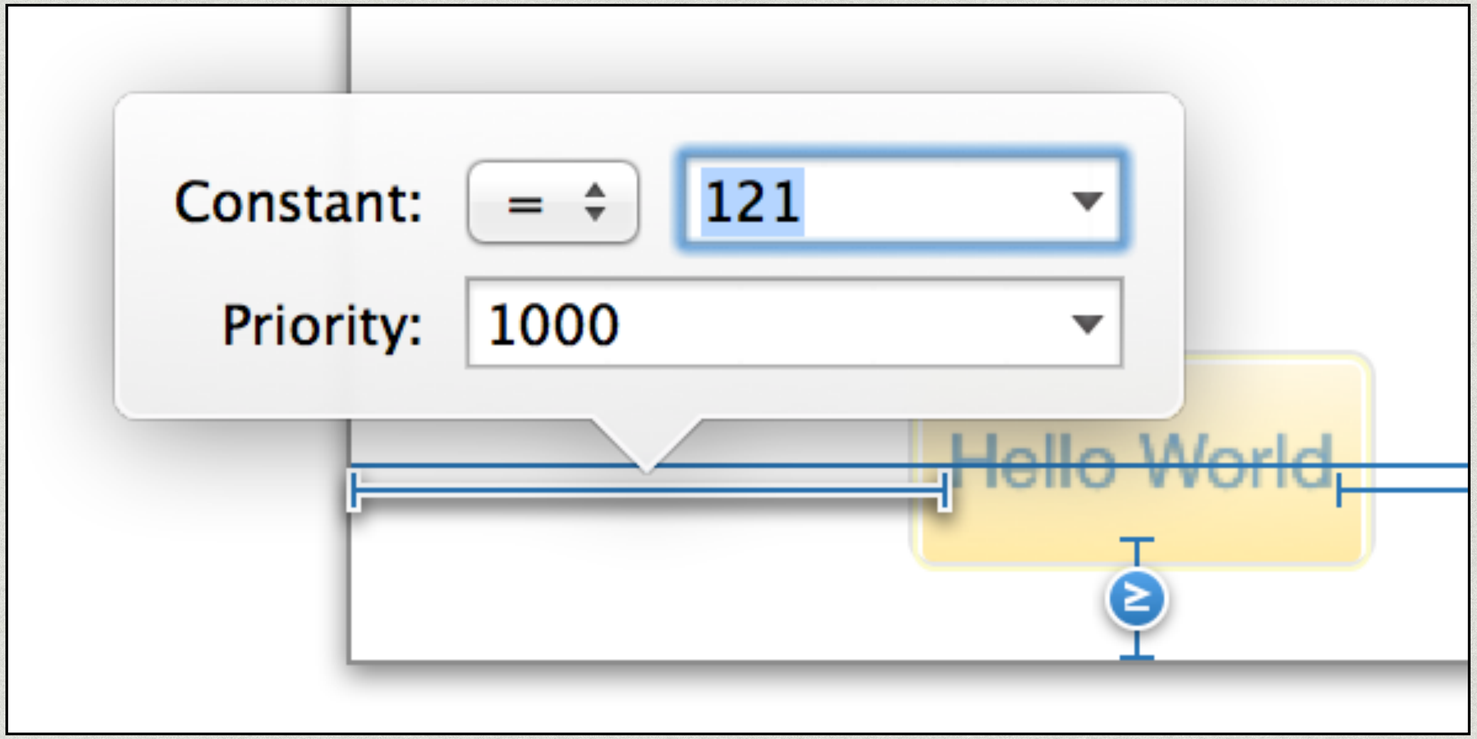
- Top down
  - Draw View Controller's view
  - Draw Container View
  - Draw Subview
- Trigger with `-(void)setNeedsDisplay;`
- Override point:

```
- (void)drawRect:(CGRect)rect;
// Overrides *may* call super.
```

THINGS YOU SHOULD KNOW

…IF YOU'RE A BABY UNICORN

Constraints!



Constraints!

```
[NSLayoutConstraint constraintsWithVisualFormat:@“H:|-[view]-|”
    options:0
    metrics:nil
    views:NSDictionaryOfVariableBindings(view)];
```

Constraint Maths

```
attribute1 = multiplier * attribute2 + constant
```

```
view.x = superview.leading + 20
```

Constraints also support >=, <= and priorities

Constraints not frames!

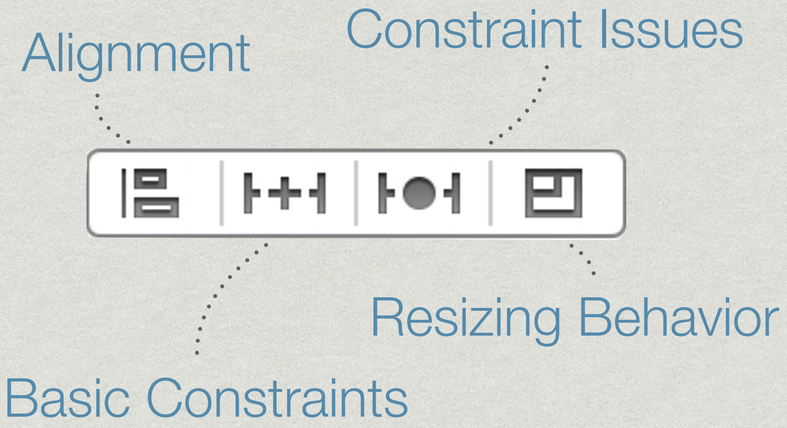
- **Do not** edit `view.frame` manually
  - Unless you want a migraine.
  - Reading from `view.frame` is fine, just do it after `-layoutSubviews`

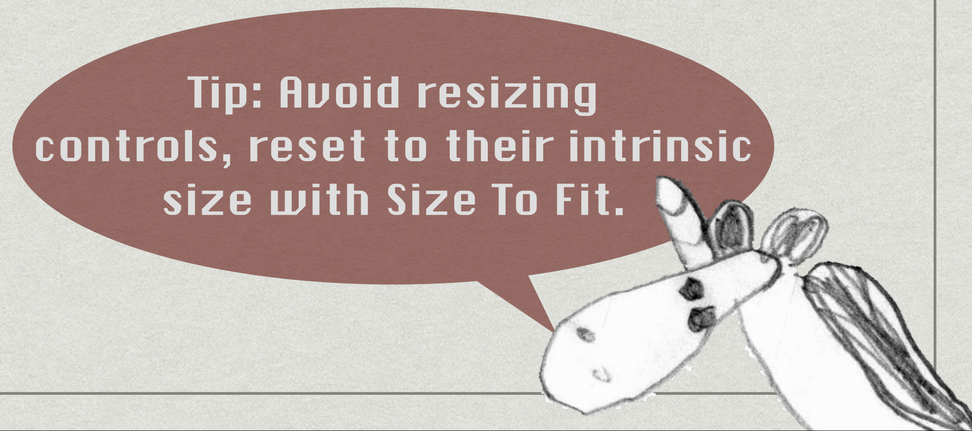
Alignment Rects not frames!

- Alignment rects give you a lot of flexibility
  - `view.alignmentRectInsets`
  - `-alignmentRectForFrame:`
  - `-frameForAlignmentRect:`

Interface Builder

- Interface Builder in Xcode 5 is usable.
  - Auto Layout tools are ~~hidden~~ stored here:
  - Now supports ambiguous layouts,
    easy alignment and some nice shortcuts
  - Use the Auto Layout Issue pane to resolve issues.
    It looks like this:





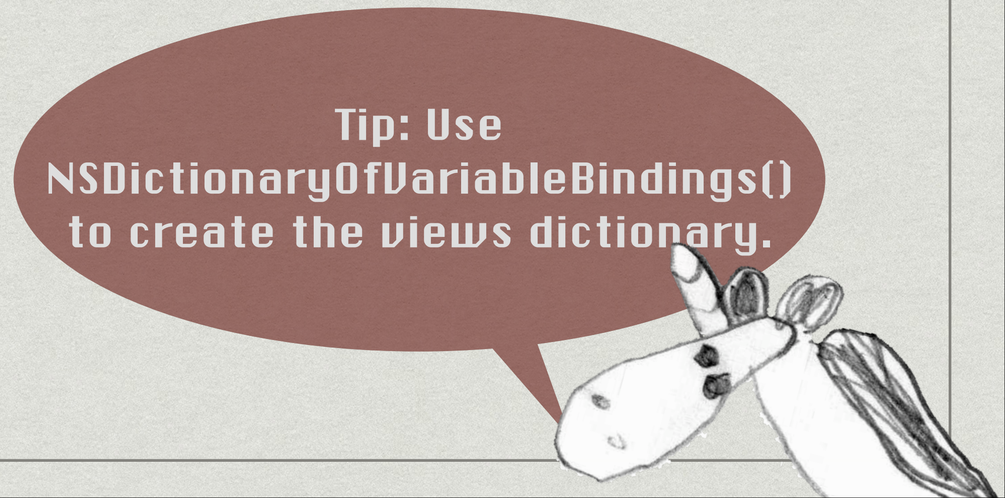
In code

- Make sure you set `view.translatesAutoresizingMaskIntoConstraints = NO;`
- Add constraints using either:

```
+ (id)constraintWithItem:(id)view1
  attribute:(NSLayoutAttribute)attr1
  relatedBy:(NSLayoutRelation)relation
  toItem:(id)view2 attribute:(NSLayoutAttribute)attr2
  multiplier:(CGFloat)multiplier
  constant:(CGFloat)c;
```

or

```
+ (NSArray *)constraintsWithVisualFormat:(NSString *)format
  options:(NSLayoutFormatOptions)opts
  metrics:(NSDictionary *)metrics
  views:(NSDictionary *)views;
```

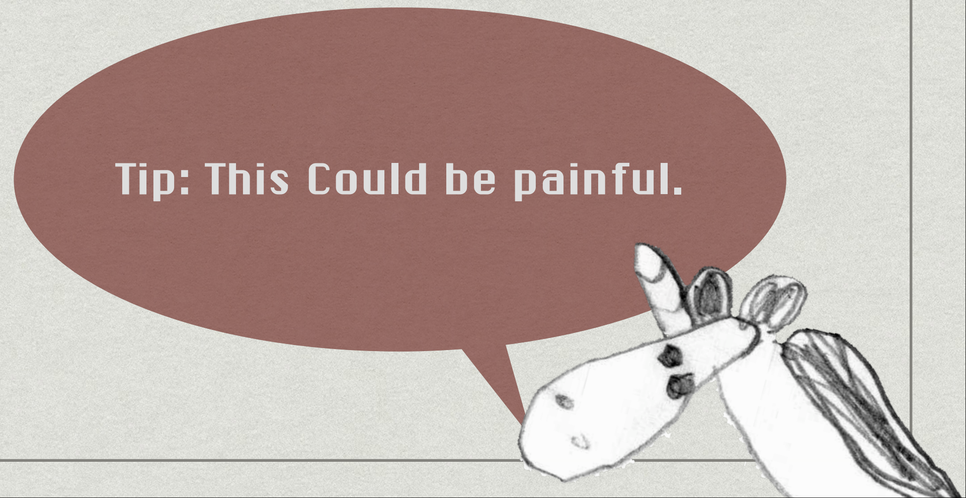


Starting a new project with Auto Layout

- New Xcode projects have Auto Layout turned on by default.
- Storyboards: create as many constraints as possible in IB
- Constraints are mutable so hook up IBOutlets to them
- Interface Builder turns off autoresizing automatically for views created there.
  - Feel free to still opt out of Auto Layout for some views. You’re not doing anything wrong.

Converting old projects to Auto Layout

- Set `translatesAutoresizingMaskIntoConstraints = NO;` on views you want to opt into Auto Layout.
- Only edit the frame on views that are not part of auto layout.
- Implement `+ (BOOL)requiresConstraintBasedLayout` for custom views.



Multiple Screen Size Support

- Rotation is something we already know how to deal with.
- Another thing we already do is support iPad and iPhone screen sizes.
- Multiple screen sizes are just a more general case.

DEMO

ANIMATING WITH AUTO LAYOUT

Two Options

- Animate your constraints
  - Some claim it’s too slow.
  - Limited on what you can do since it has to be defined as a constraint.
- Animate with CoreAnimation
  - CoreAnimation layer animations don’t invalidate view frames.
  - Feels a bit dirty, like you’re two timing Auto Layout.

Animating Constraints

- Simple once you find your target constraint
  - One easy way is to use outlets.

```
@property IBOutlet NSLayoutConstraint *animatedConstraint;

…

self.animatedConstraint.constant = change;

[UIView animateWithDuration:0.5f animations:^{
    [self.view layoutIfNeeded];
}];
```

Animating with CoreAnimation

- Seems best for temporary flouting of auto layout constraints

```
CABasicAnimation* animation = [CABasicAnimation
    animationWithKeyPath:@“translation”];

animation.duration = 0.5f;
animation.autoreverses = YES;
animation.toValue = [NSValue
    valueWithCATransform3D:CATransform3DMakeTranslation(100, 100, 0)];

[viewToAnimate.layer addAnimation:animation forKey:nil];
```

Animating Transforms

- Container View with minimal constraints
- Can set `translatesAutoresizingMaskIntoConstraints = YES;` for views inside this container. Then layout layout frames.

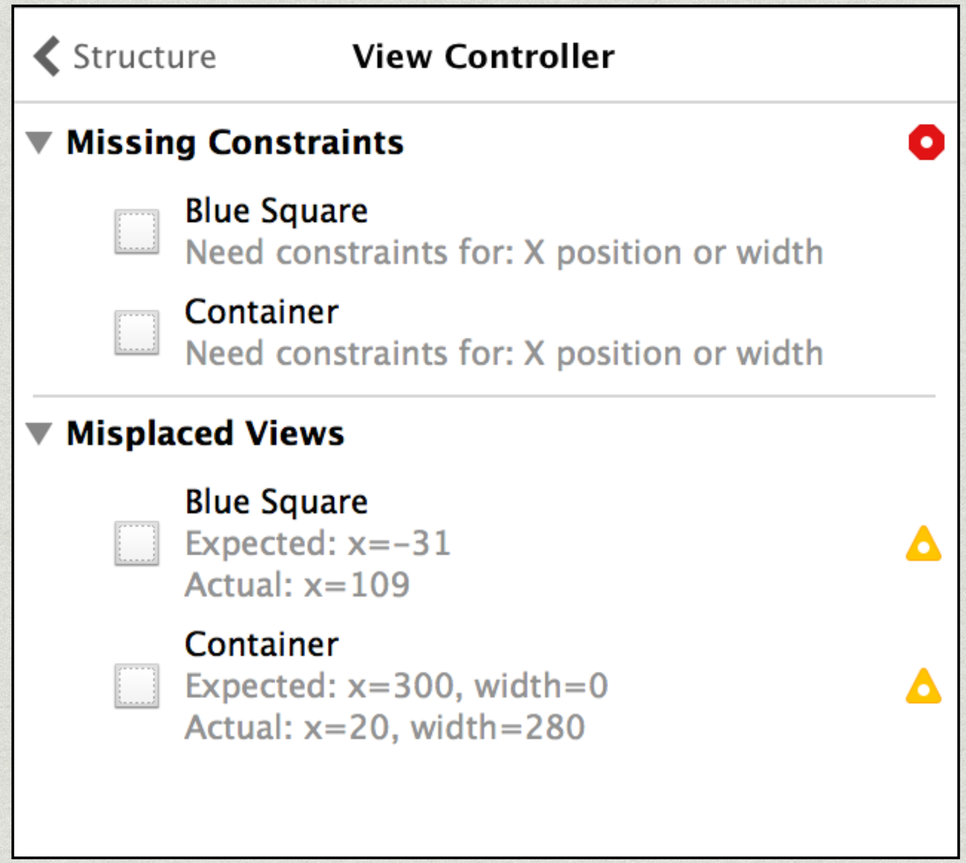
```
CGAffineTransform animationTransform =
    CGAffineTransformMakeScale(scaleFactor, scaleFactor));

[UIView animateWithDuration:0.5f animations:^{
    viewToTransform.transform =
        CGAffineTransformConcat(viewToTransform.transform, animationTransform);
}];
```

ANIMATION DEMO

DEBUGGING AUTO LAYOUT

What do I do with those little orange lines in Interface Builder?

* Auto Layout Issue Navigator
  * Took me forever to find this on my own.
  * It then took me forever to start using the icons on the right.
* Editor ❯ Canvas ❯ Show Involved Views For Selected Constraints
* Editor ❯ Canvas ❯ Show Intrinsic Size Constraints Contributing To Ambiguity.



Constraints

- If you see `NSLayoutResizingMaskConstraints` in the console you probably should turn `translatesAutoresizingMaskIntoConstraints` off.
- Reference Apple's "Resolving Issues" section of their Auto Layout Programming Guide and look for the problem you're having.
- Use UIView's constraint debugging methods:

```
- (NSArray *)constraintsAffectingLayoutForAxis:(UILayoutConstraintAxis)axis
- (BOOL)hasAmbiguousLayout
- (void)exerciseAmbiguityInLayout
```

NSLayoutConstraint

- Add category methods to improve debug output.
  - Justin Williams suggests something like this:

```
#ifdef DEBUG
- (NSString *)constraint_description
{
    NSString *description = [self description];

    return [description stringByAppendingFormat:@"(%@, %@)",
            [self.firstItem restorationIdentifier],
            [self.secondItem restorationIdentifier]];
}
#endif
```

DEBUGGING DEMO

THINGS YOU SHOULD KNOW

…IF YOU'RE A BAD-ASS UNICORN

Custom Views

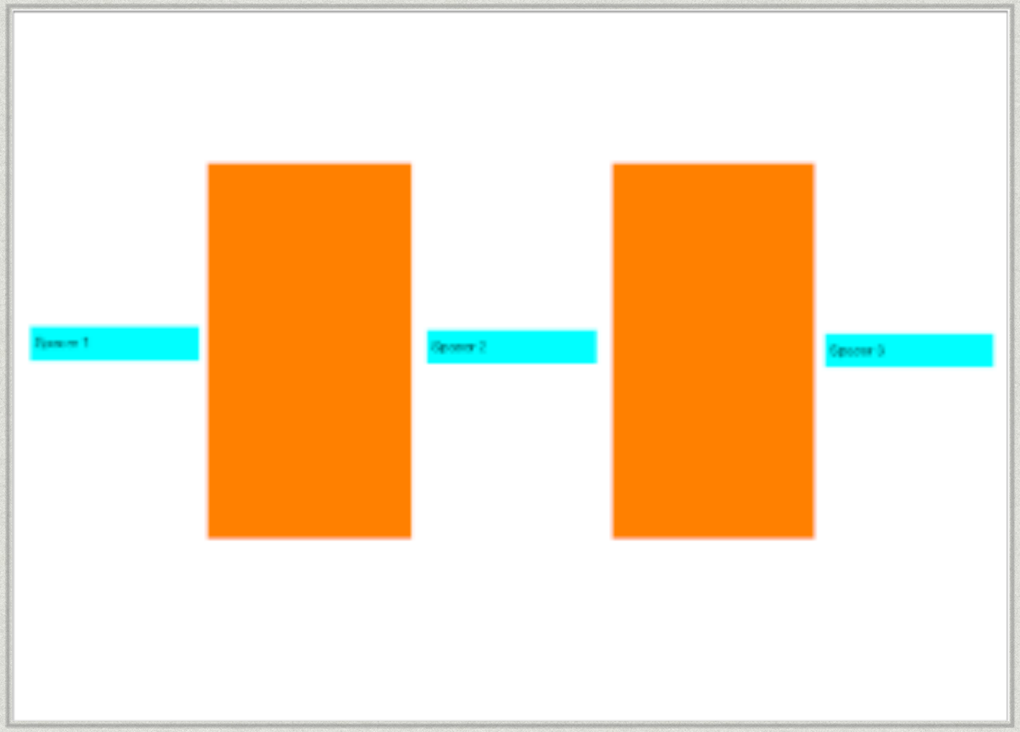
- Implement `-updateConstraints` to build constraints for your view's content or `- (void)updateViewConstraints` at the view controller level.
- Implement `-requiresConstraintBasedLayout` if you don't degrade gracefully.
- Make sure your alignment rect is correct given content ornamentation.
  - If you have text content: `- (UIView *)viewForBaselineLayout`

Content Size

- Define your content size with `-intrinsicContentSize`
  - You must notify Auto Layout if this changes by calling: `-invalidateIntrinsicContentSize`
- Content hugging
- Content compression resistance

Spacer Views

* This is the 1x1 transparent gif of Auto Layout
* Allows for lots of complex layouts.
* Read Apple's "Auto Layout By Example" Documentation.



1. Unicorns can be captured only by unfair means, and their single horn is said to neutralize poison.

2. One popular method of hunting unicorns involved entrapment by a virgin.

3. When Marco Polo saw a rhinoceros for the first time, he thought it was a unicorn.

-Art credit: Elinor, 5 years old.

Q&A

Talk notes: http://nicemohawk.com/talks/auto-layout/

Talk notes: http://nicemohawk.com/talks/auto-layout/

Ben Lachman
NICE MOHAWK LIMITED

@blach
ben@nicemohawk.com

Q&A

- WHO USES AUTO LAYOUT ALREADY?
- I WANT A MORE CONVENIENT WAY TO FIDDLE WITH AUTO LAYOUT. CATEGORIES ANYONE?
- HAS ANYONE INTEGRATED UIDYNAMICS AND AUTO LAYOUT?

Talk notes: http://nicemohawk.com/talks/auto-layout/

Talk notes: http://nicemohawk.com/talks/auto-layout/

Ben Lachman
NICE MOHAWK LIMITED

@blach
ben@nicemohawk.com