

# Advanced View Concepts

Matthias Tretter, @myell0w

# Initializing a UIView

- ⦿ Yes, you can override initWithFrame:

Use previously explained `self = [super initWithFrame:aRect]` syntax.

- ⦿ But you will also want to set up stuff in `awakeFromNib`

This is because initWithFrame: is NOT called for a UIView coming out of a storyboard!

But awakeFromNib is.

It's called "awakeFromNib" for historical reasons.

- ⦿ Typical code ...

```
- (void)setup { ... }

- (void)awakeFromNib { [self setup]; }

- (id)initWithFrame:(CGRect)aRect
{
    self = [super initWithFrame:aRect];
    [self setup];
    return self;
}
```

# Sizing

- Manually sizing views

`@property (nonatomic, assign) CGRect frame`

- autosizing to fit needed size (built-in views)

`- (CGSize)sizeThatFits:(CGSize)rect;`

“Calculates and returns a size that best fits the receiver’s subviews.”

- `- (void)sizeToFit;`

„Call this method when you want to resize the current view so that it uses the most appropriate amount of space.“

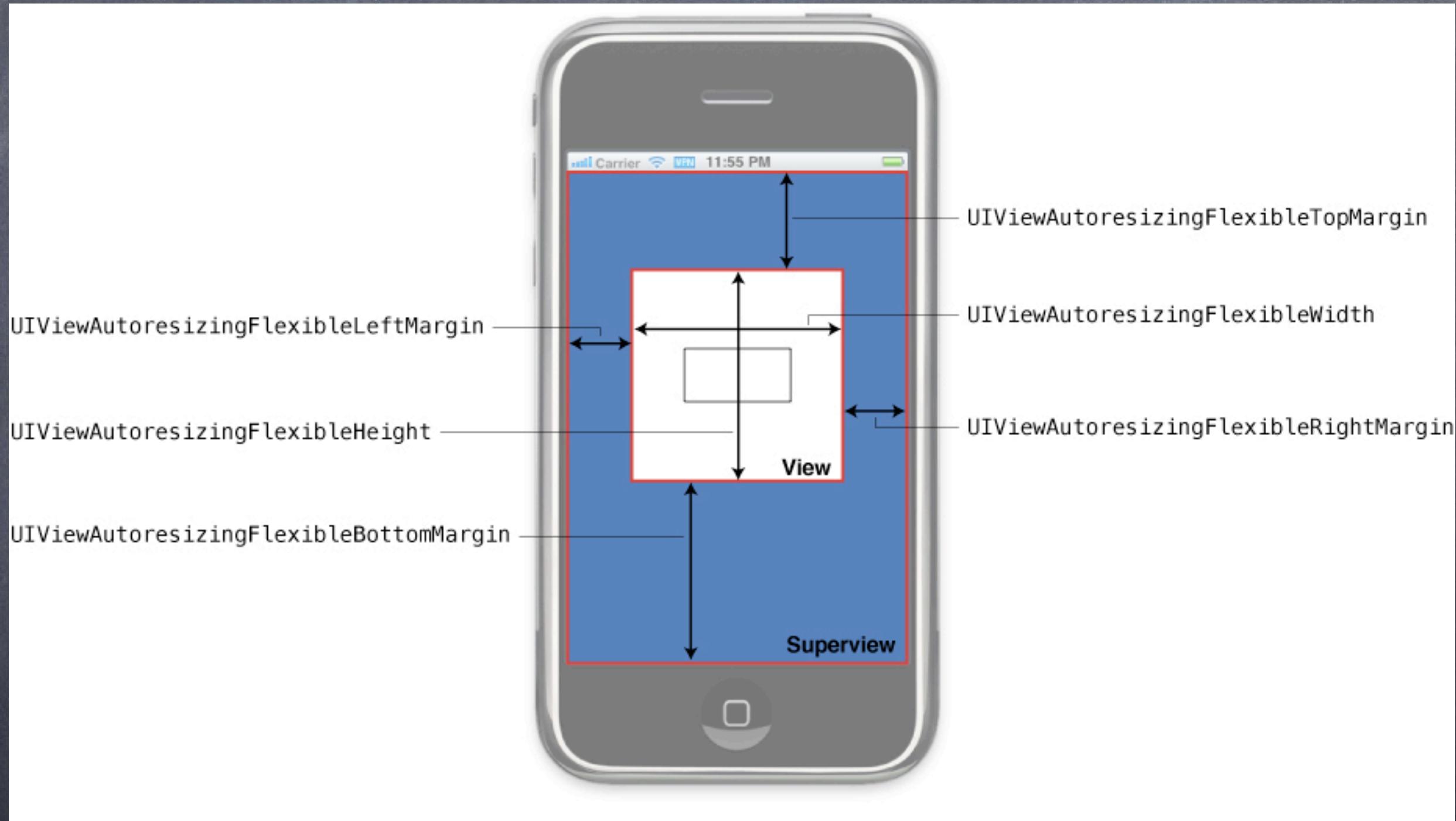
# Autoresizing

- ⦿ Most often: rotation. Device orientation changes, therefore the bounds of the Window/Controller's view. Subviews need to be repositioned.
- ⦿ **@property (nonatomic) UIViewAutoresizing autoresizingMask**

```
typedef NS_OPTIONS(NSUInteger, UIViewAutoresizing) {  
    UIViewAutoresizingNone          = 0,  
    UIViewAutoresizingFlexibleLeftMargin = 1 << 0,  
    UIViewAutoresizingFlexibleWidth   = 1 << 1,  
    UIViewAutoresizingFlexibleRightMargin = 1 << 2,  
    UIViewAutoresizingFlexibleTopMargin = 1 << 3,  
    UIViewAutoresizingFlexibleHeight  = 1 << 4,  
    UIViewAutoresizingFlexibleBottomMargin = 1 << 5  
};
```

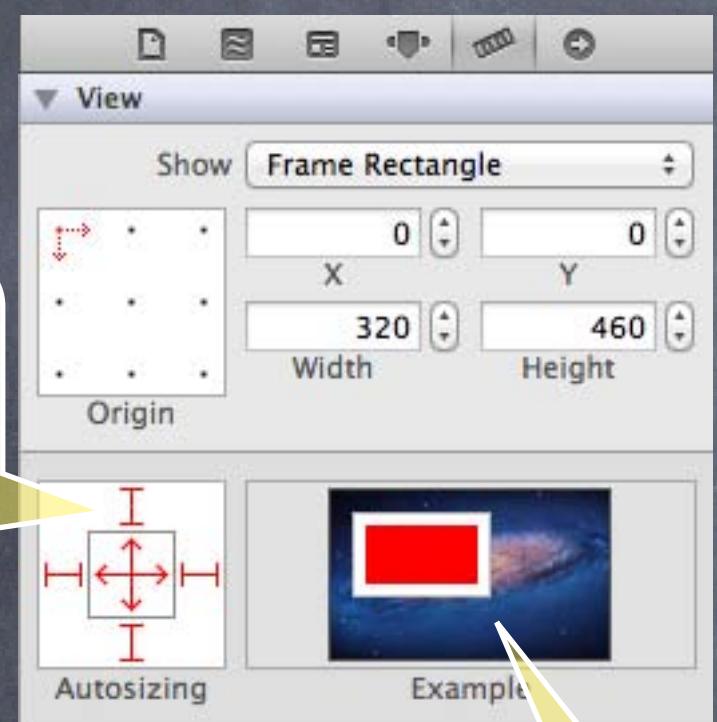
# Autoresizing

What happens, when the bounds of the superview change?



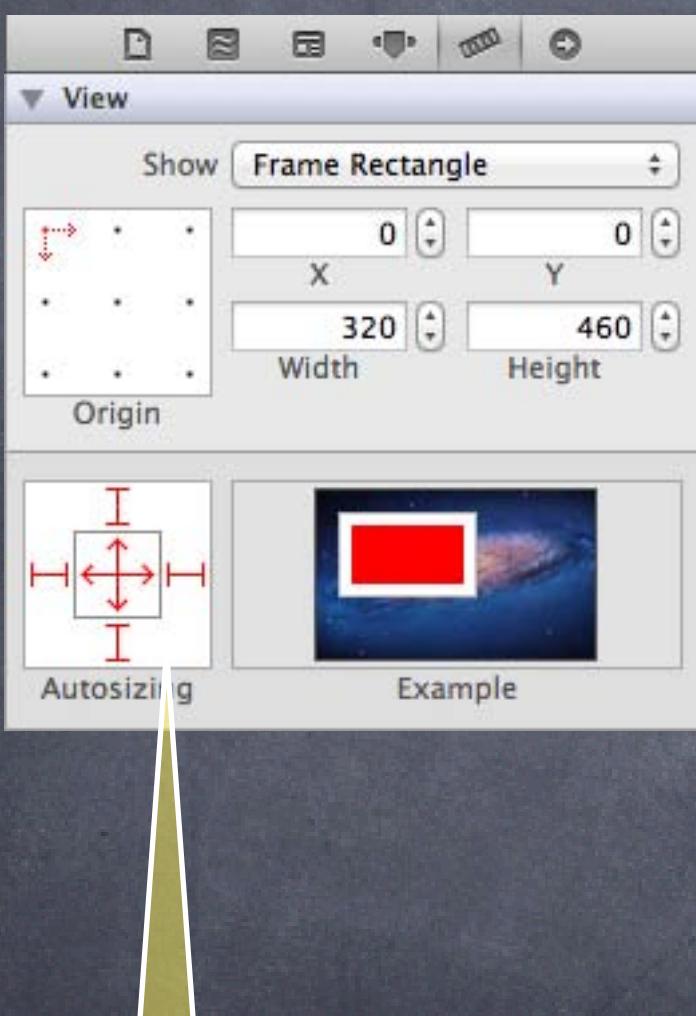
# Struts and Springs

- Set a view's struts and springs in size inspector in Xcode



# Struts and Springs

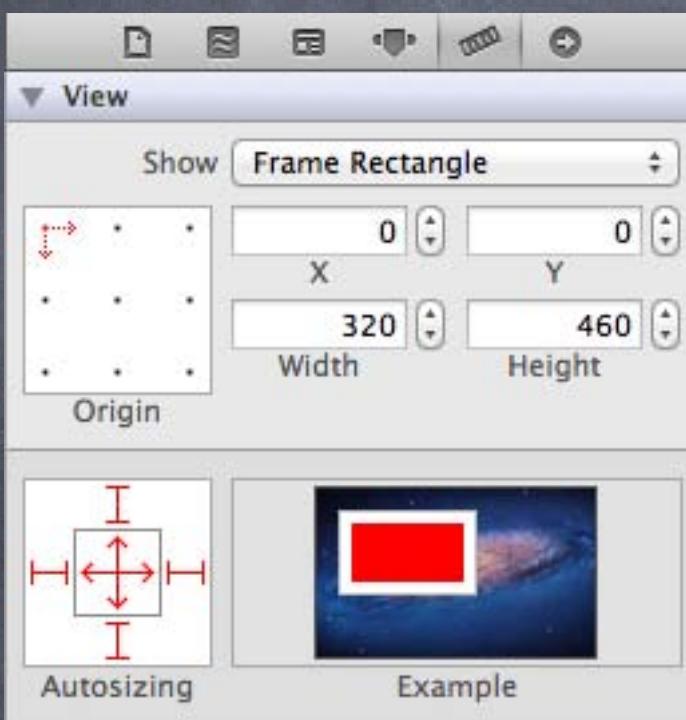
- Set a view's struts and springs in size inspector in Xcode



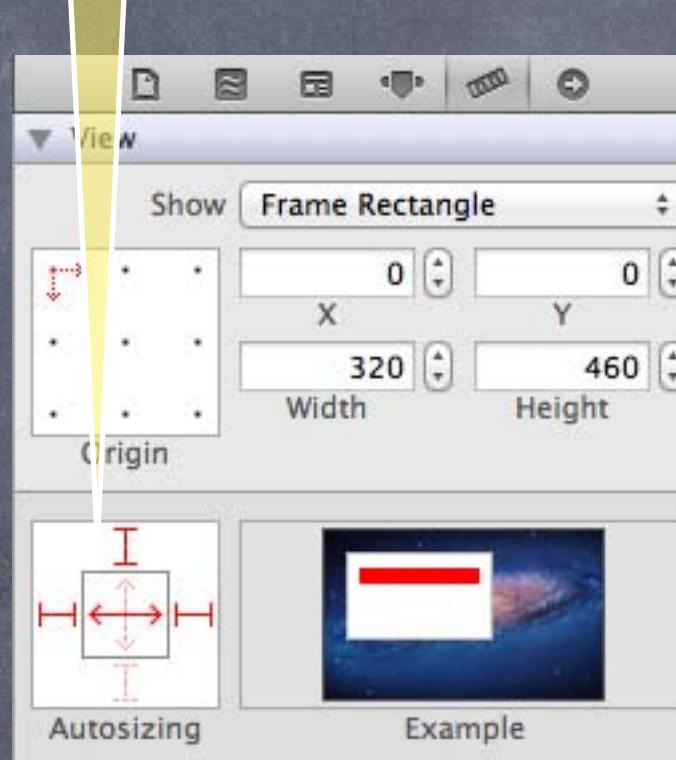
Grows and shrinks as its superview's bounds grow and shrink because struts fixed to all sides and both springs allow expansion.

# Struts and Springs

- Set a view's struts and springs in size inspector in Xcode

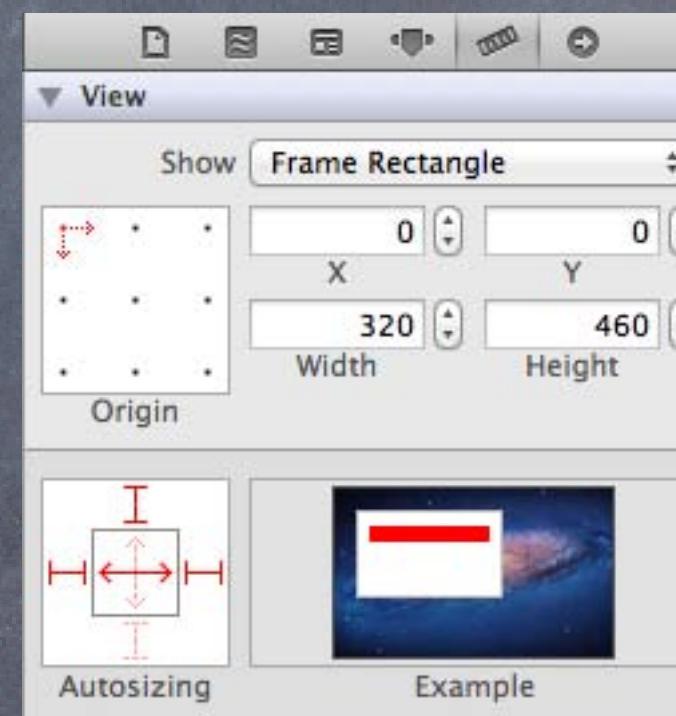
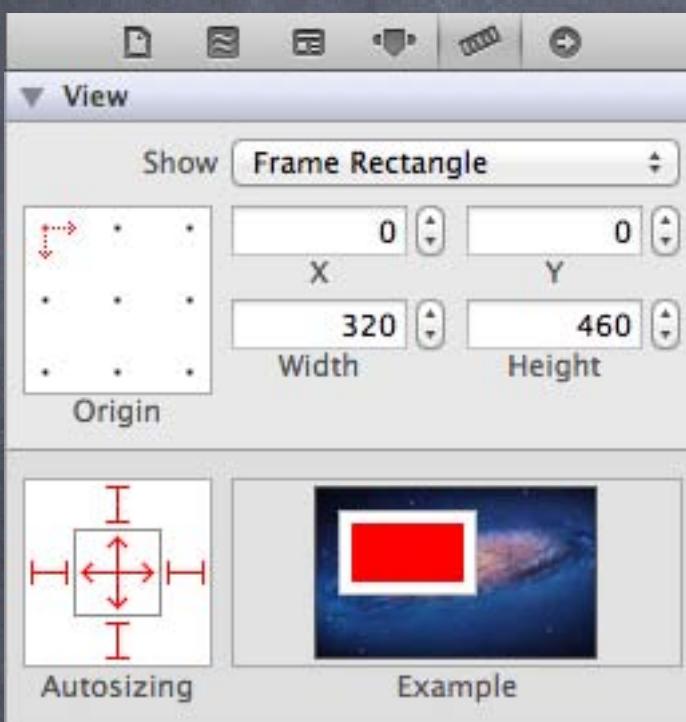


Grows and shrinks only horizontally as its superview's bounds grow and shrink and sticks to the top in its superview.

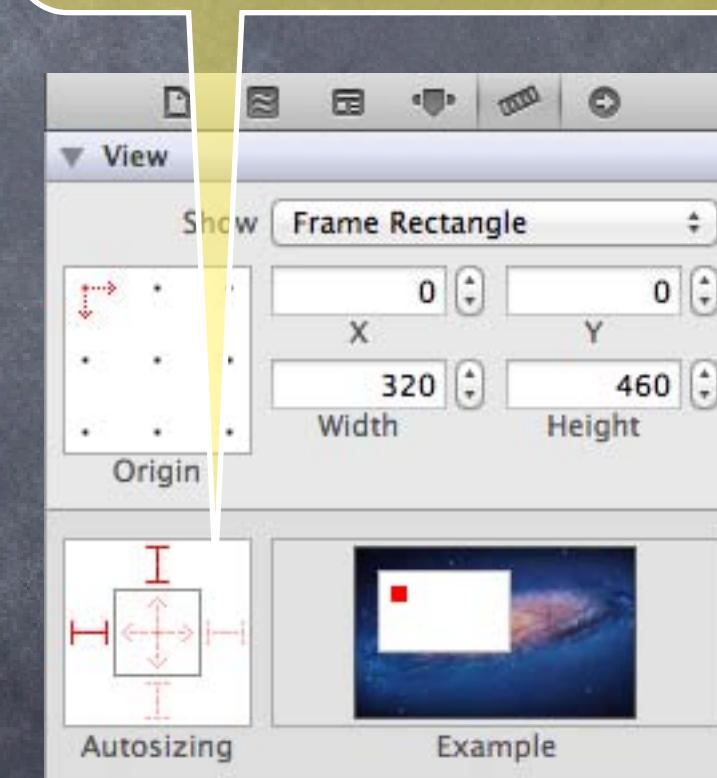


# Struts and Springs

- Set a view's struts and springs in size inspector in Xcode

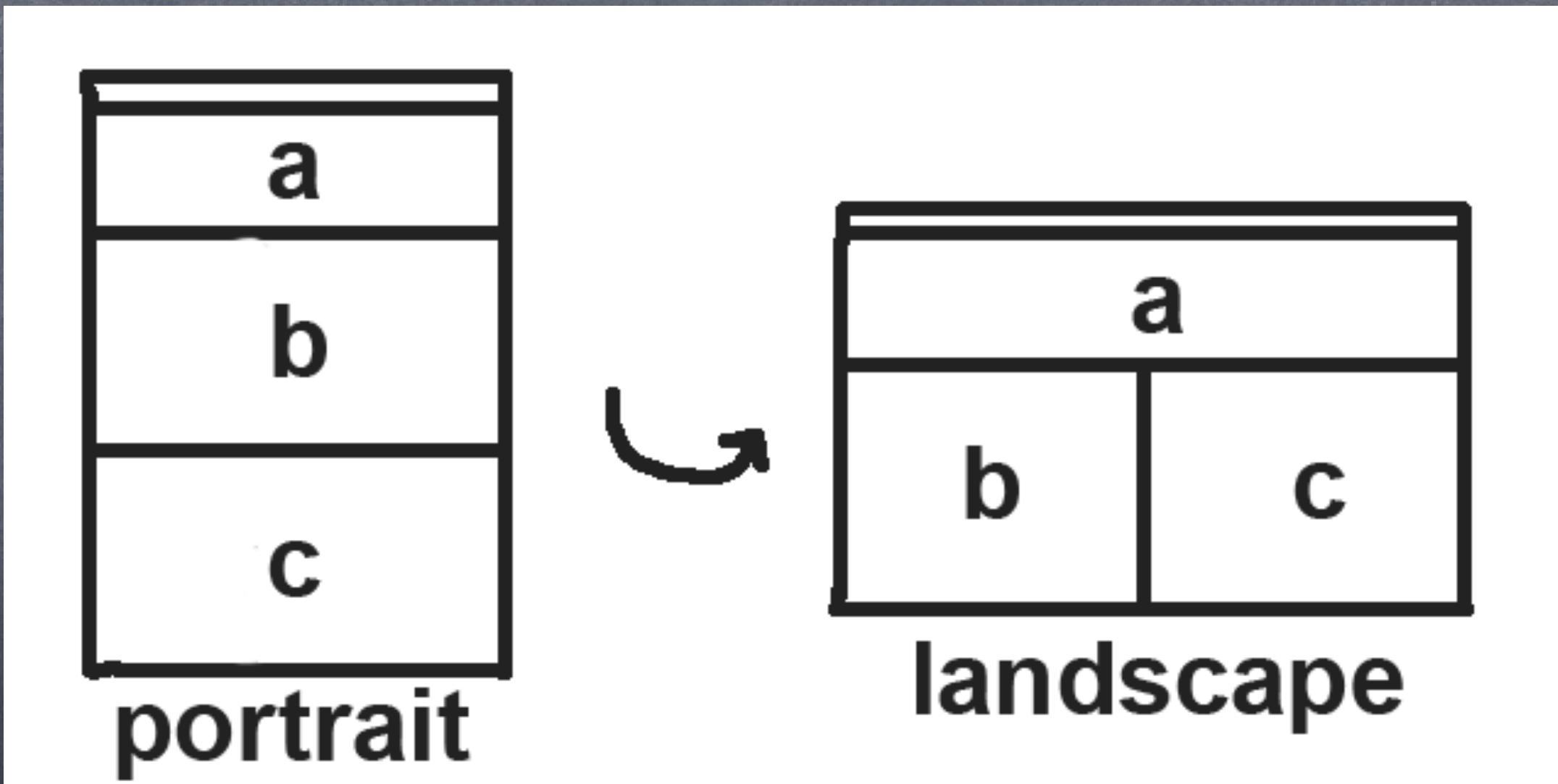


Sticks to upper left corner  
(fixed size).



# Layouting

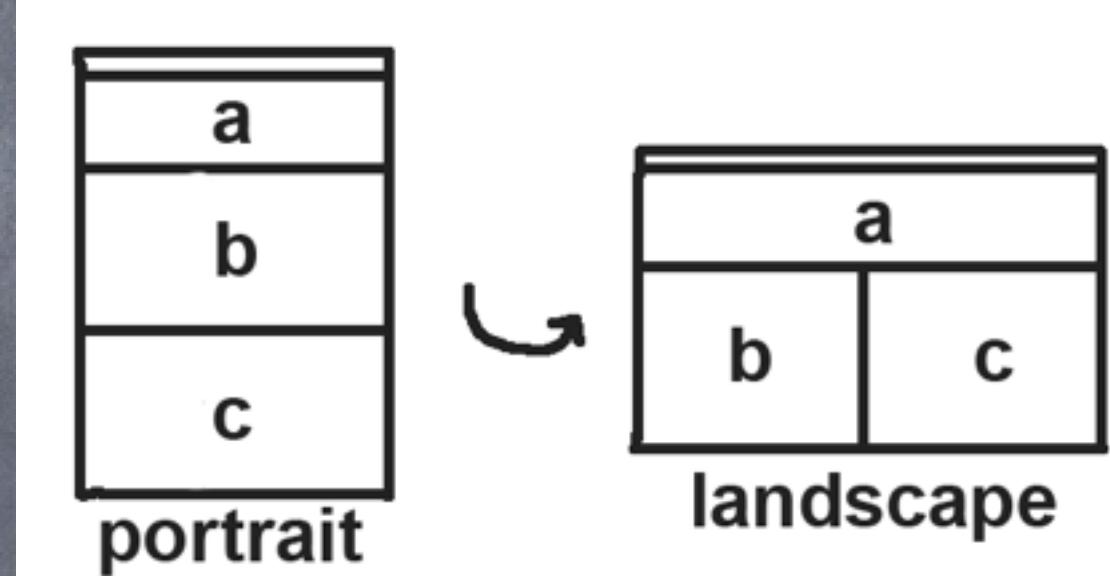
What if autoresizing isn't sufficient?



# Layouting

- ⦿ You can manually layout views, if autoresizing doesn't work
- ⦿ How?
  - in View subclass:
    - `(void)layoutSubviews`
    - `(void)viewWillLayoutSubviews`
  - in Controller:
    - `[view setNeedsLayout]`
- ⦿ Indicate that a view needs relayout?

# Layouting



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

    if (UIInterfaceOrientationIsPortrait(self.interfaceOrientation)) {
        a.frame = CGRectMake(0,0,320,150);
        b.frame = CGRectMake(0,150,320,150);
        c.frame = CGRectMake(0,300,320,150);
    } else {
        a.frame = CGRectMake(0,0,480,100);
        b.frame = CGRectMake(0,100,480,100);
        c.frame = CGRectMake(0,200,480,100);
    }
}
```

# Animation

- Very easy to animate views
- The following properties of the UIView class are animatable:

```
@property frame  
@property bounds  
@property center  
@property transform           // affine transform (scale, rotate, ...)  
@property alpha  
@property backgroundColor  
@property contentStretch      // deprecated since iOS 6
```

# Animation

```
+ (void)animateWithDuration:(NSTimeInterval)duration  
    animations:(void (^)(void))animations  
  
+ (void)animateWithDuration:(NSTimeInterval)duration  
    delay:(NSTimeInterval)delay  
    options:(UIViewAnimationOptions)options  
    animations:(void (^)(void))animations  
    completion:(void (^)(BOOL finished))completion
```

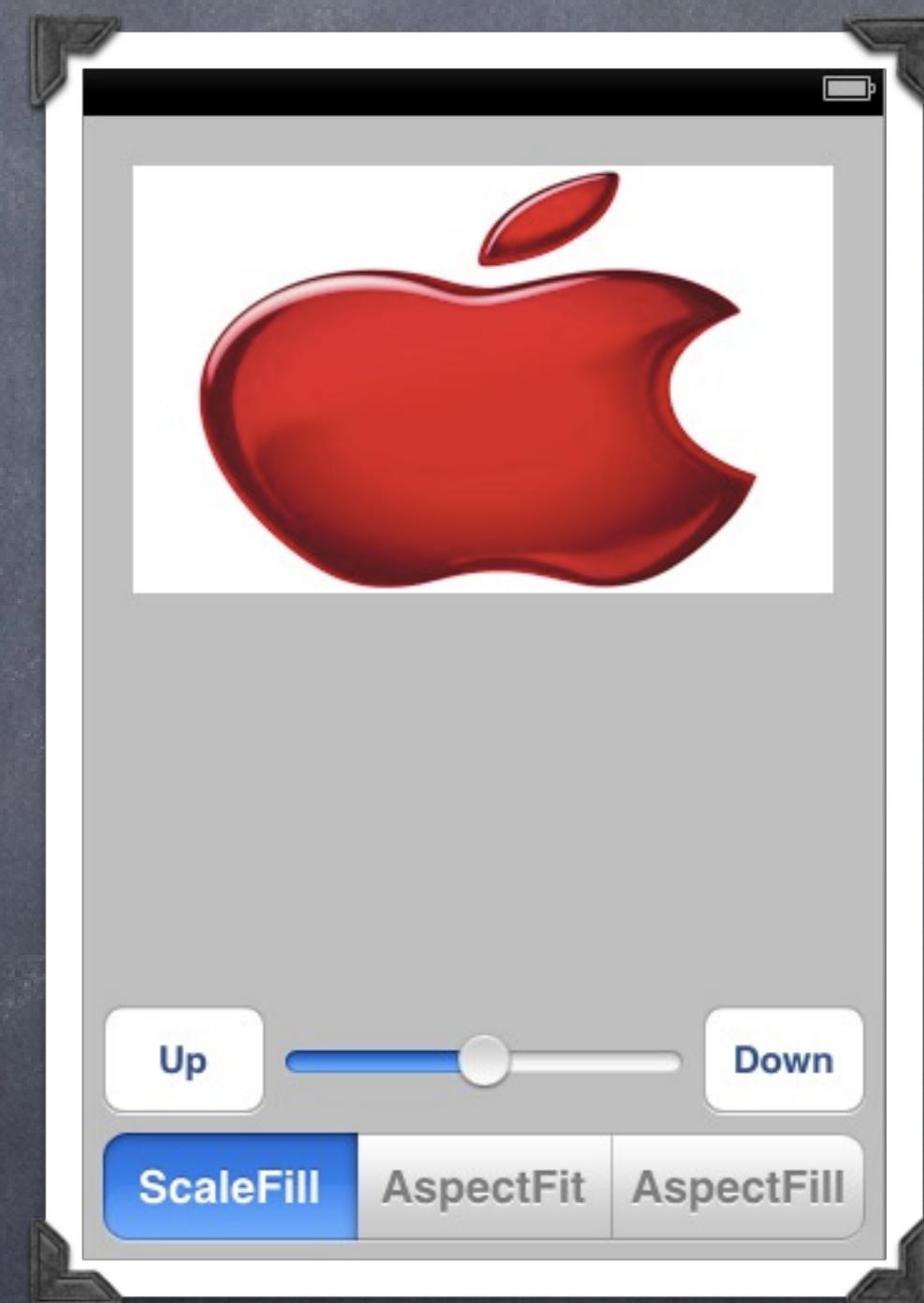
# Animation Example

```
[UIView animateWithDuration:1.5 // seconds
    animations:^{
        // animate position of view
        view.center = CGPointMake(50.f,300.f);

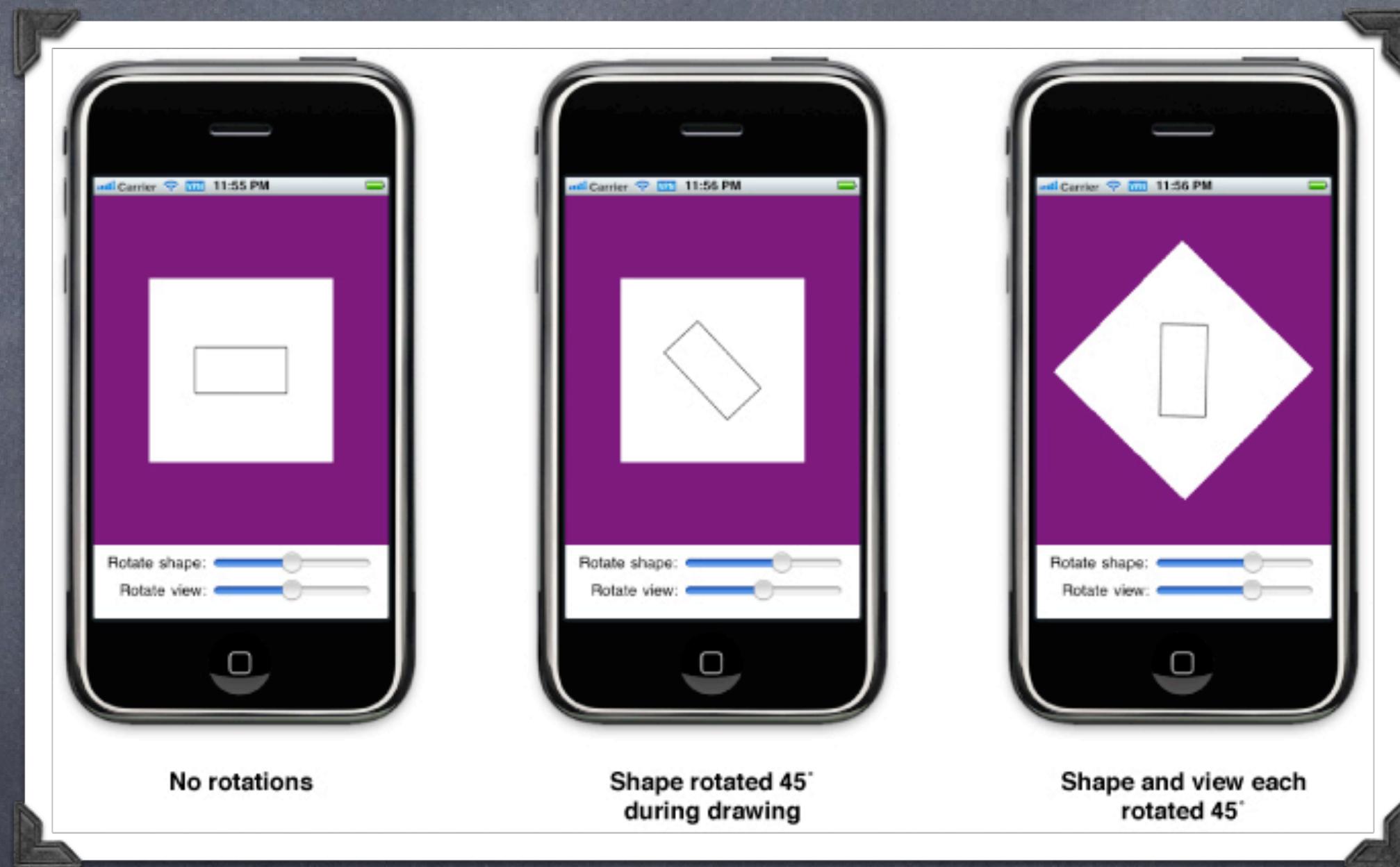
        // scale view to doubled size
        view.transform = CGAffineTransformMakeScale(2.f,2.f);
    }];

    // prints 50/300
    NSLog(@"Center: %f/%f", view.center.x, view.center.y);
```

# Demo App: View Fun



# Demo App: Rotate Views



# Further Literature

- ⦿ View Programming Guide for iOS, Apple  
[http://developer.apple.com/library/ios/#documentation/windowsviews/conceptual/viewpg\\_iphoneos/Introduction/Introduction.html](http://developer.apple.com/library/ios/#documentation/windowsviews/conceptual/viewpg_iphoneos/Introduction/Introduction.html)