Advanced View Concepts

Matthias Tretter, @myell0w

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."

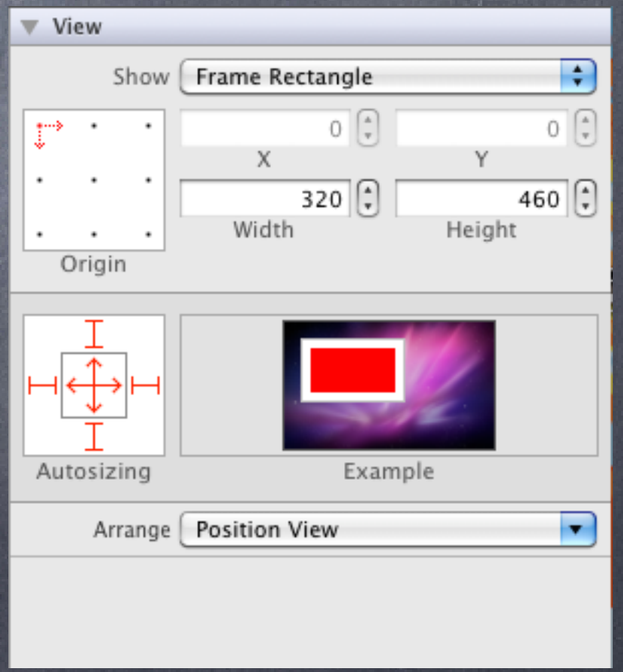
- `- (void)sizeToFit;`

  „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

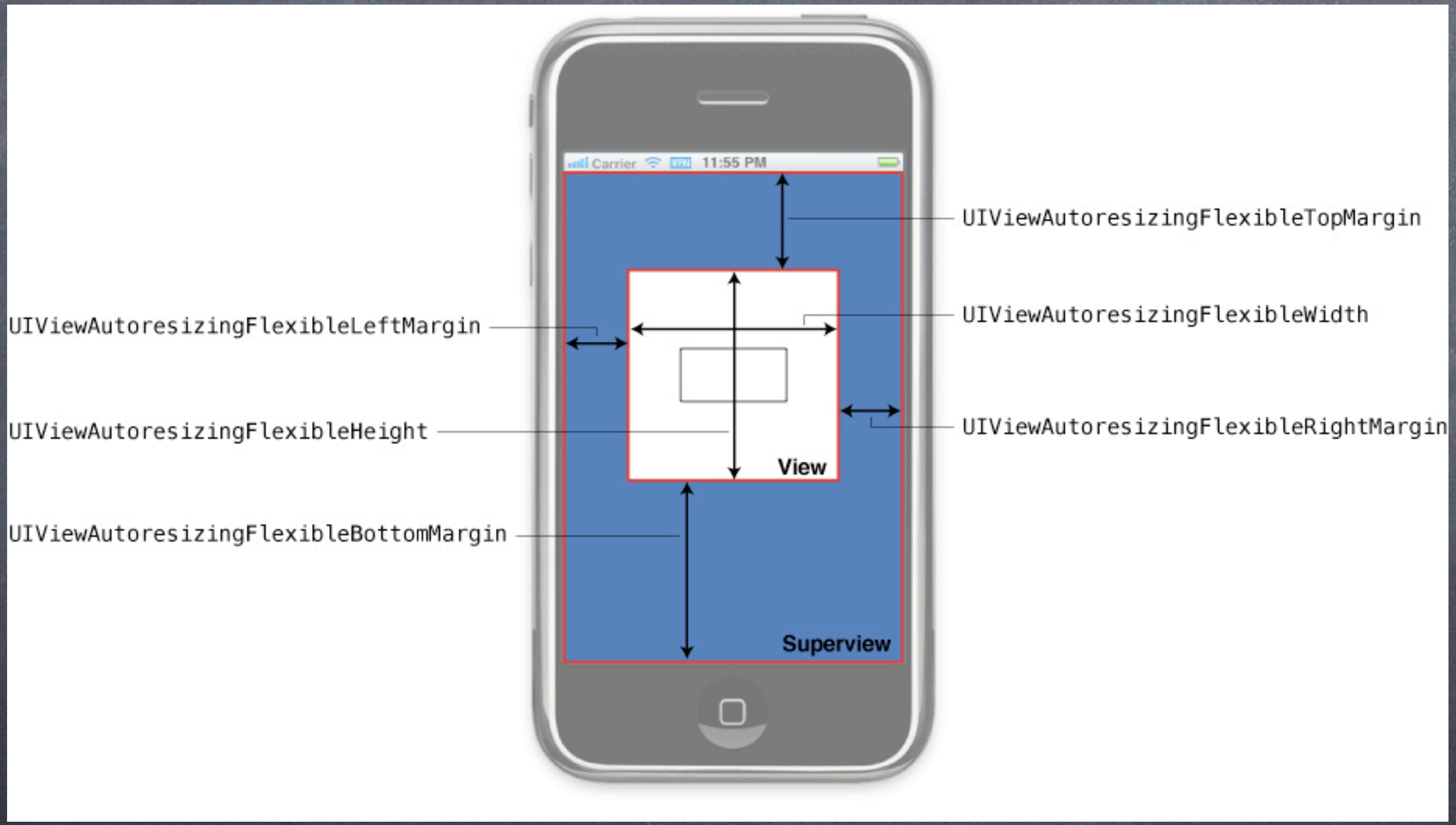
```
typedef NS_OPTIONS(NSUInteger, UIViewAutoresizing) {
  UIViewAutoresizingNone                 = 0,
  UIViewAutoresizingFlexibleLeftMargin   = 1 << 0,
  UIViewAutoresizingFlexibleWidth        = 1 << 1,
  UIViewAutoresizingFlexibleRightMargin  = 1 << 2,
  UIViewAutoresizingFlexibleTopMargin    = 1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



- Xcode Interface Builder visual support

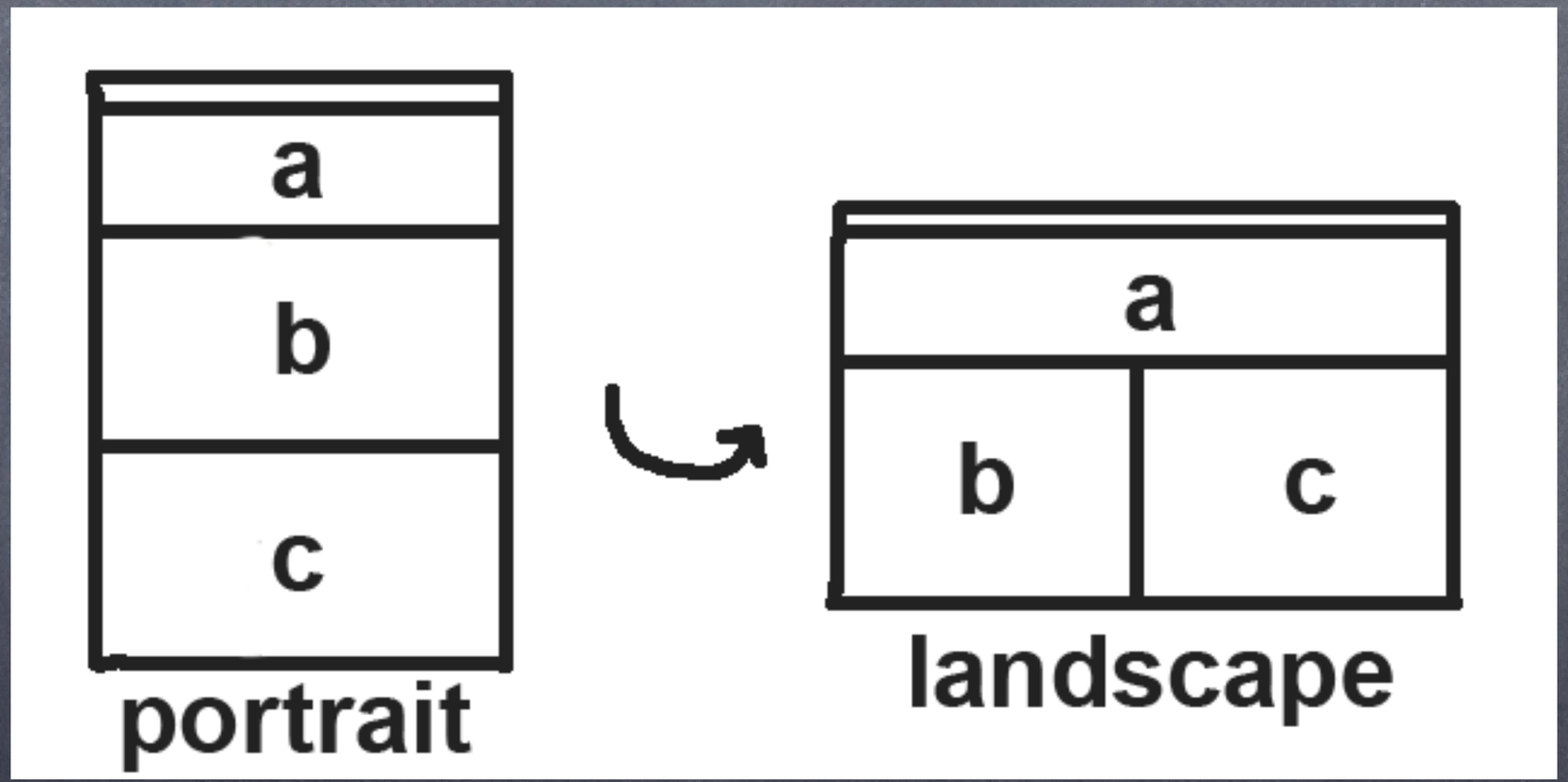
Autoresizing

What happens, when the bounds of the superview change?



Layouting

What if autoresizing isn't sufficient?



Layouting

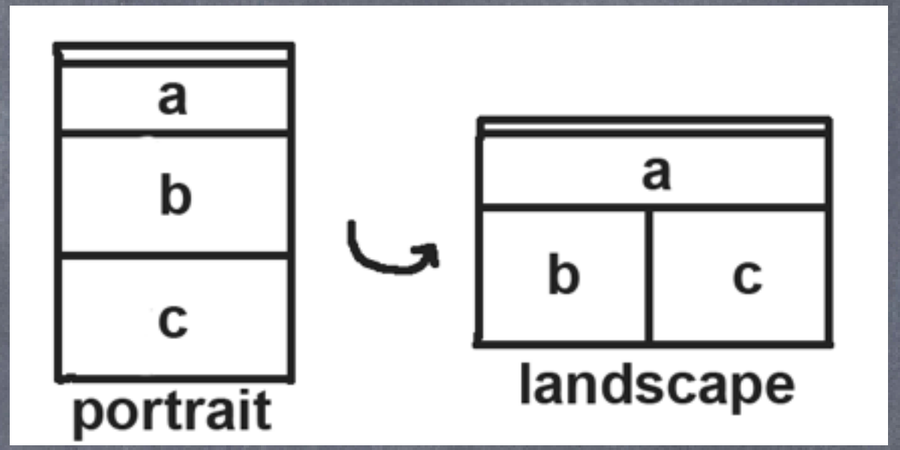
- You can manually layout views, if autoresizing doesn't work
- How?

  in View subclass: `- (void)layoutSubviews`
  in Controller: `- (void)viewWillLayoutSubviews`

- Indicate that a view needs relayout?

```
[view setNeedsLayout]
```

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);
```

Further Literature

- View Programming Guide for iOS, Apple
  http://developer.apple.com/library/ios/#documentation/windowsviews/conceptual/viewpg_iphoneos/Introduction/Introduction.html