



# *Core Animation*

*Scott Stevenson*

*Essentials*

# *Why Core Animation*

Simpler graphics model

True layers

Animation without timers

Right thing by default

Very fast

Excellent Cocoa citizen

Multi-tiered approach



*It's the*

**FUTURE**

# *Core Animation Provides*

Display

Layout

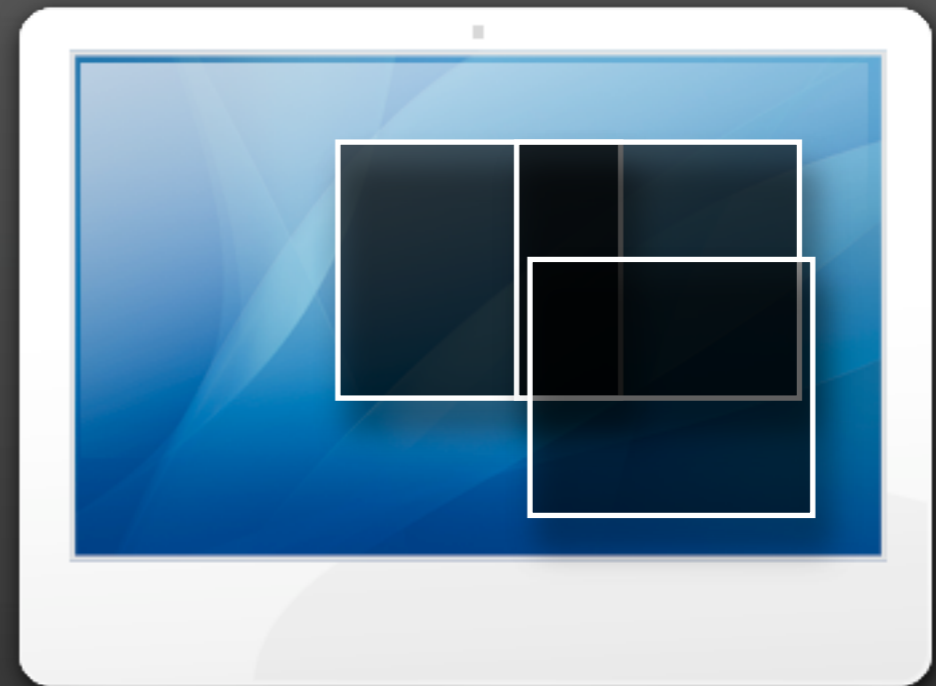
Filters + Transitions

3D Space

Animation

Caching

Basic Styling



*Works With*

Quartz 2D

Core Image

ImageIO

OpenGL

UIKit

Quartz Composer

*Layers*

# *CALayer*



Basic data structure

Real onscreen object

Content

Styling

Subclassing not required

*Similar* to `NSView`

# *CALayer vs NSView*



**NSView**

NSResponder

AppKit (NS) data types

Some overhead

High-level event support

Timer-based animation



**CALayer**

NSObject

CoreGraphics (CG) types

Lightweight

Content on video card

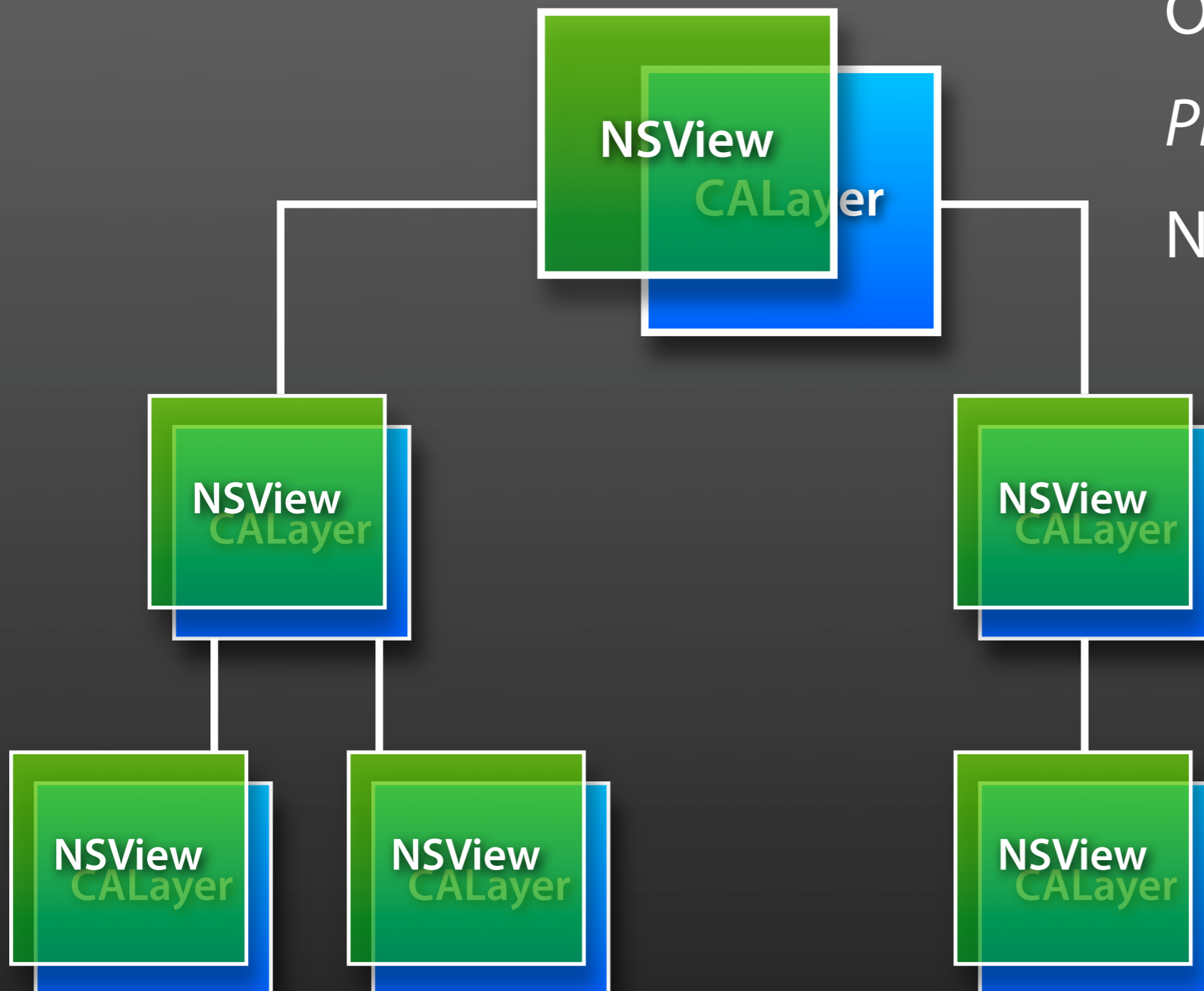
Animations + Transitions

# Setup

```
@implementation MyView
-(void) awakeFromNib {
    self.wantsLayer = YES;
    self.layer.frame = self.frame;
}
}
```

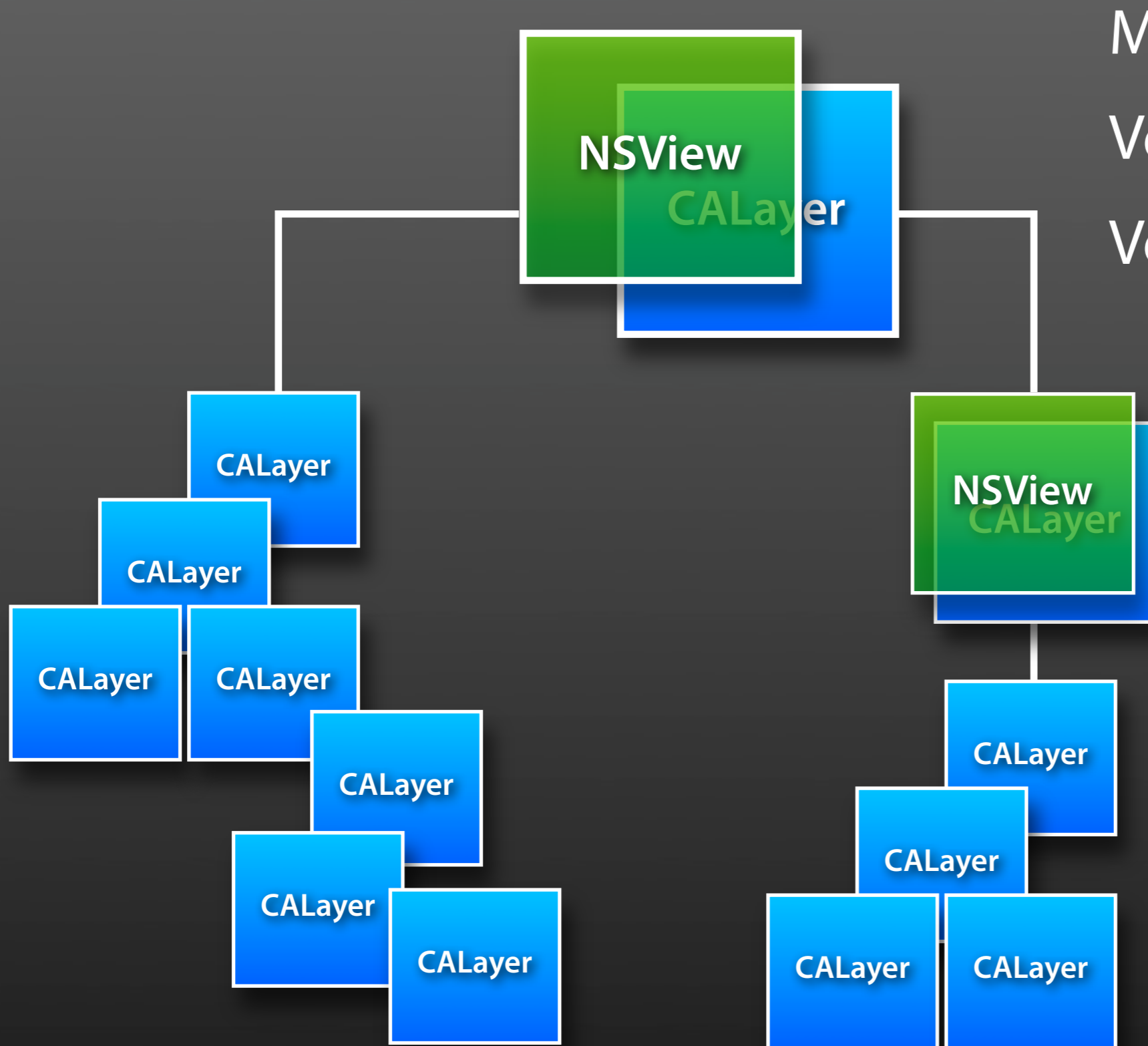


# Trees of Views



OK for dozens  
*Probably* OK for 100s  
Not great for 1000s

# Trees of Layers



Mix and match  
Very scalable  
Very easy

# *CALayer Properties*

CGRect **frame**;

CGRect **bounds**;

CGPoint **position**;

CGPoint **anchorPoint**;

CGFloat **zPosition**;

CATransform3D **transform**;

CALayer **\*superlayer**;

NSArray **\*sublayers**;



# *CALayer Properties*

BOOL **hidden**;

float **opacity**;

CGColorRef **shadowColor**;

float **shadowOpacity**;

CGSize **shadowOffset**;

CGFloat **shadowRadius**;

CGFloat **cornerRadius**;

CGFloat **borderWidth**;

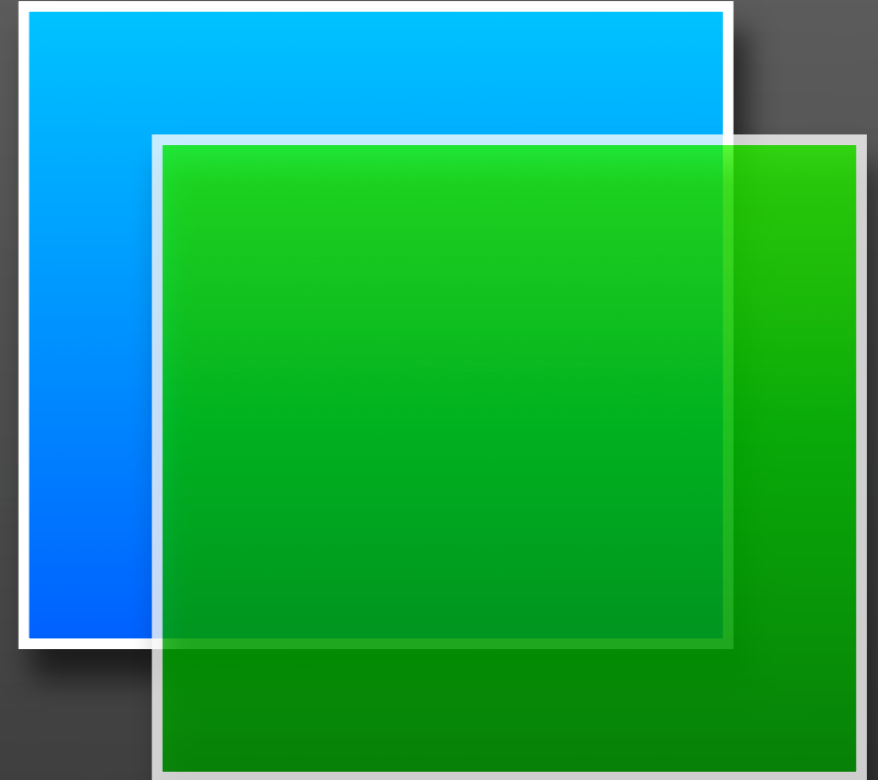
CGColorRef **borderColor**;

CGColorRef **backgroundColor**;



# *CALayer Filters*

```
id compositingFilter;  
NSArray *filters;  
NSArray *backgroundFilters;
```



# CALayer

```
CGColorRef black;  
CGColorRef white;  
  
CALayer* myLayer = [CALayer layer];  
myLayer.frame = CGRectMake (0,0,20,20);  
myLayer.backgroundColor = black;  
myLayer.borderColor = white;  
myLayer.borderWidth = 2.5;  
myLayer.cornerRadius = 4.0;  
NSDictionary* attributes;  
myLayer.style = attributes;
```

# Sublayers

```
CALayer* containerLayer = [CALayer layer];

CALayer* imageLayer = [CALayer layer];
CATextLayer* textLayer = [CATextLayer layer];
imageLayer.name = @"imageContent";
textLayer.name = @"textContent";
[containerLayer addSublayer: imageLayer];
[containerLayer addSublayer: textLayer];

for ( CALayer* layer in containerLayer.sublayers )
    NSLog(@"Layer: %@", layer.name);
```

***Animation***

# ***CAAnimation***

CAPropertyAnimation

*CABasicAnimation*

*CAKeyframeAnimation*

CATransition

CAAnimationGroup



# *CABasicAnimation*



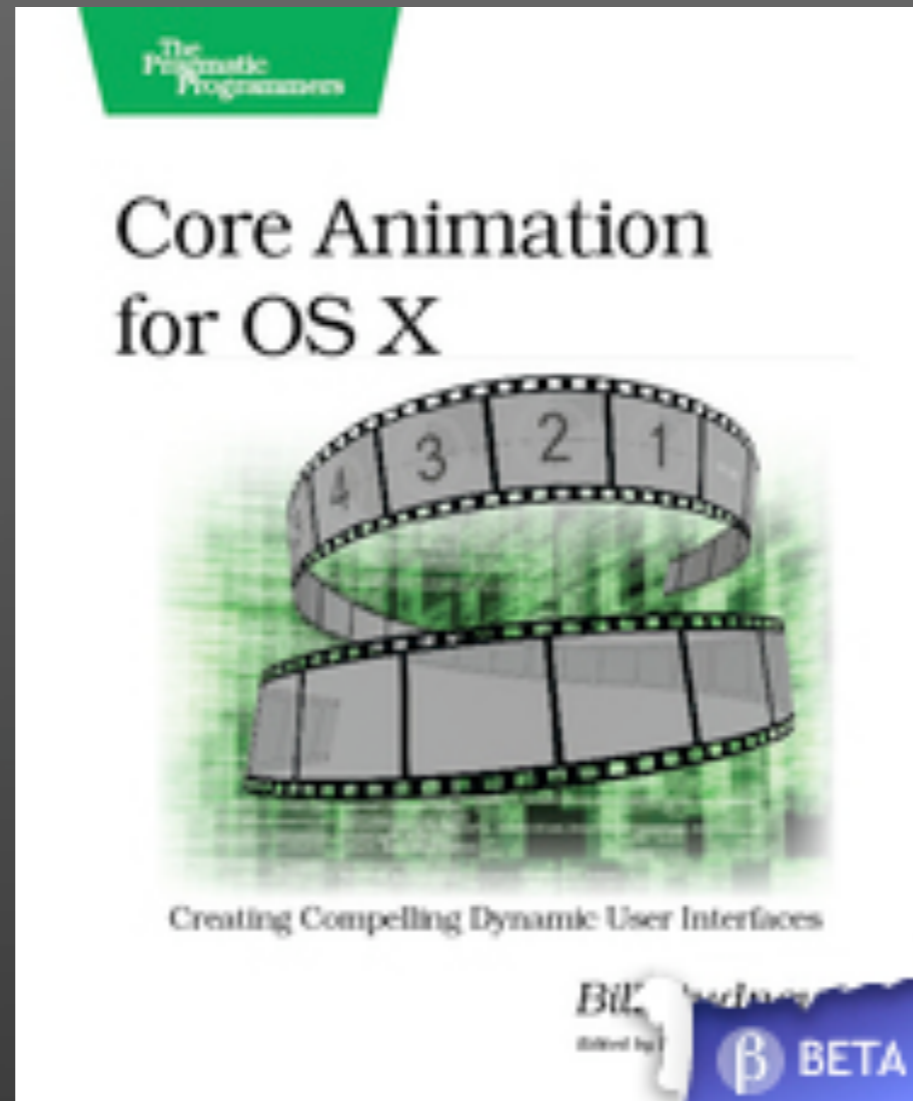
Point A to Point B

# *CAKeyframeAnimation*



Multiple Stop Points

*Wrap Up*



# Core Animation for OS X

# *Wrap-up*

<http://theocacao.com/>