



Building Animation Driven Interfaces

André Boulé
iOS Software Engineer





Building Animation Driven Interfaces

André Boulé
iOS Software Engineer

animated: YES



UINavigationController:

```
- (void)pushViewController:  
(UIViewController *)viewController  
animated:(BOOL)animated;
```



UITableView:

```
- (void)deleteRowsAtIndexPaths:(NSArray *)  
indexPaths withRowAnimation:  
(UITableViewRowAnimation)animation;
```

Built-in Animations

- Context
- Consistency
- Less work for you

Going Beyond...

Agenda

- UIView animations
- Transitions
- Window rotation animations
- Performance
- CoreAnimation

UIView Animations

UIView Animations

Comparing to timer-based animations



setAlpha:0.0

setAlpha:0.2

setAlpha:0.4

setAlpha:0.6

setAlpha:0.8

setAlpha:1.0

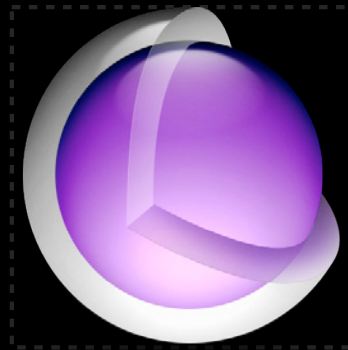


UIView Animations

Using UIView and CoreAnimation

setAlpha:0.0

setAlpha:1.0



Hardware
Accelerated!

UIView Animations

```
// myView alpha is currently 0.0  
[UIView beginAnimations:@"fadeIn" context:nil]  
[UIView setAnimationDuration:2.0];  
[myView setAlpha:1.0];  
[UIView commitAnimations];
```



UIView Animations

New in iOS 4: Blocks API

```
// myView alpha is currently 0.0  
[UIView animateWithDuration:2.0 animations:^(  
    [myView setAlpha:1.0];  
)];
```



Moving a View

```
CGRect viewFrame = [myView frame];  
viewFrame.origin.x += 400;  
[UIView animateWithDuration:2.0 animations:^(  
    [myView setFrame:viewFrame];  
)];
```



Rotating a View

```
[UIView animateWithDuration:2.0 animations:^(  
    [myView setTransform:  
        CGAffineTransformMakeRotation(M_PI)];  
)];
```



Combining Animations

```
[UIView animateWithDuration:2.0 animations:^(  
    [myView setFrame:viewFrame];  
    [myView setAlpha:0];  
)];
```



Completion Block

```
[UIView animateWithDuration:2.0 animations:^(
    [myView setFrame:viewFrame];
)
completion:^(BOOL finished){
    [UIView animateWithDuration:2.0 animations:^(
        [myView setAlpha:0];
    )];
}];
```



Non-Animatable Properties

- BOOL clipsToBounds
- BOOL hidden
 - But you can use alpha!
- BOOL opaque

UIView Animations

- UIView animations API is:
 - Simple
 - Works with subviews
 - Gets you 90% there...

Beyond the Basics

Advanced UIView animations

Canceling an Animation

- No explicit cancel
- Set property while animating
 - Jumps to “set” value
 - Finished parameter is NO

Animation Options

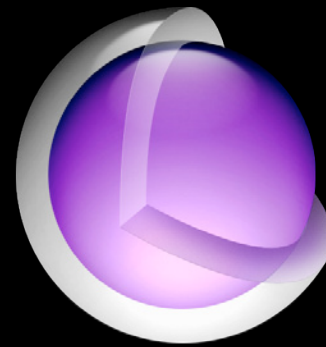
- Setter methods
- Using with blocks:

```
[UIView animateWithDuration:DURATION  
    delay:DELAY  
    options:(UIViewAnimationOptions)options  
    animations:^(  
    completion:^(BOOL finished){ }];
```

1. Repeating an Animation



Repeat



Repeat
Autoreverse

2. Animations and Interaction

- App-wide interaction disabled
 - Blocks API only
- Override with `AllowUserInteraction`

3. Animation Curves

- **CurveEaseInOut** : default
 - Ramps animation up and down



3. Animation Curves

- CurveLinear
 - No ramp up or down



3. Animation Curves



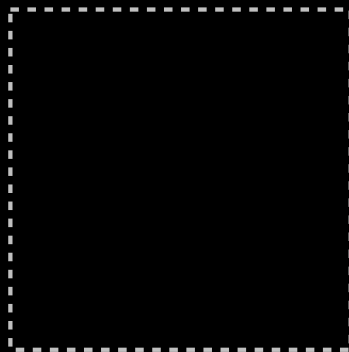
CurveLinear
→



CurveEaseInOut
→



4. Begin from Current State Option



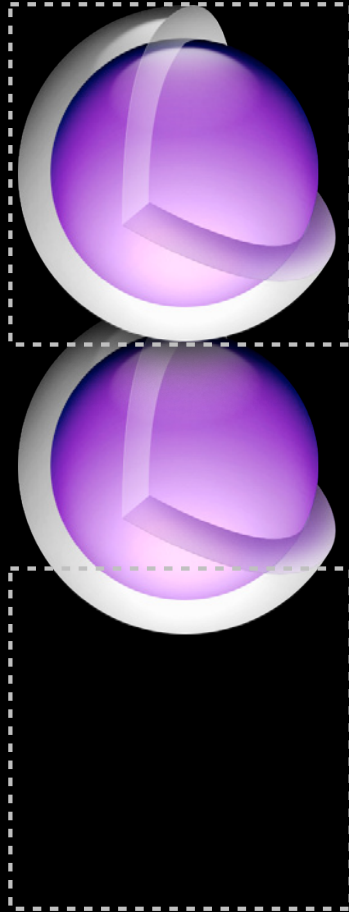
4. Begin from Current State Option



- Second animation starts mid-way through first
- Without BeginFromCurrentState



4. Begin from Current State Option



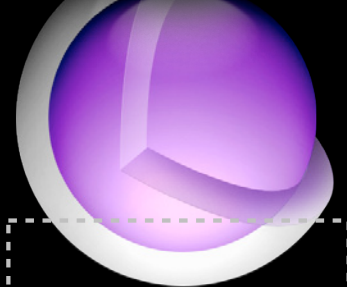
- With `BeginFromCurrentState`



4. Begin from Current State Option



- With `BeginFromCurrentState`
- With `CurveEaseOut`

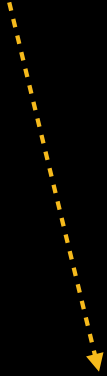


5. Animation Property Inheritance

```
[UIView animateWithDuration:2.0 animations:^(
    [myView setAlpha:0.0];
    [myView shrinkFrame];
)];

...

@implementation MyView
- (void)shrinkFrame
{
    [UIView animateWithDuration:1.02.0 animations:^(
        [self setFrame:CGRectZero];
    )];
}
```



OverrideInheritedDuration

OverrideInheritedCurve

Animation Options

1. Repeat/Autoreverse
2. AllowUserInteraction
3. Curves
4. BeginFromCurrentState
5. OverrideInheritedDuration/Curve

Transitions





transitionWithView:

- Animates state changes to a view
 - You make the change(s) to the view
 - System animates between before and after

```
+ (void)transitionWithView:(UIView *)view
    duration:(NSTimeInterval)duration
    options:(UIViewAnimationOptions)options
    animations:(void (^)(void))animations
    completion:(void (^)(BOOL finished))completion;
```

transitionWithView Example

```
- (void)flipToNextPage
{
    NSString *nextPageText = [notepadPages textForPage:c+1];
    [noteView setText:nextPageText];
    [UIView transitionWithView:noteView duration:1.0
        options:UIViewAnimationOptionTransitionCurlUp
        animations:nil
        completion:nil];
}
```

transitionFromView:toView:

- Animates between two views
 - You provide both views
 - System manages them

```
+ (void)transitionFromView:(UIView *)fromView  
    toView:(UIView *)toView  
    duration:(NSTimeInterval)duration  
    options:(UIViewAnimationOptions)options  
    completion:(void (^)(BOOL finished))completion;
```

transitionFromView:toView:example

```
-(void)flipToWidgetBackside
{
    // contentView has frontView as a child.

    [UIView transitionFromView:frontView
                     toView:backView
                     duration:1.0
                     options:UIViewAnimationOptionTransitionFlipFromLeft
                     completion:nil];

    // contentView will now have backView as a child.
}
```


Transitions API

- Single view:
 - `transitionWithView:`
- Two view:
 - `transitionFromView:toView:`

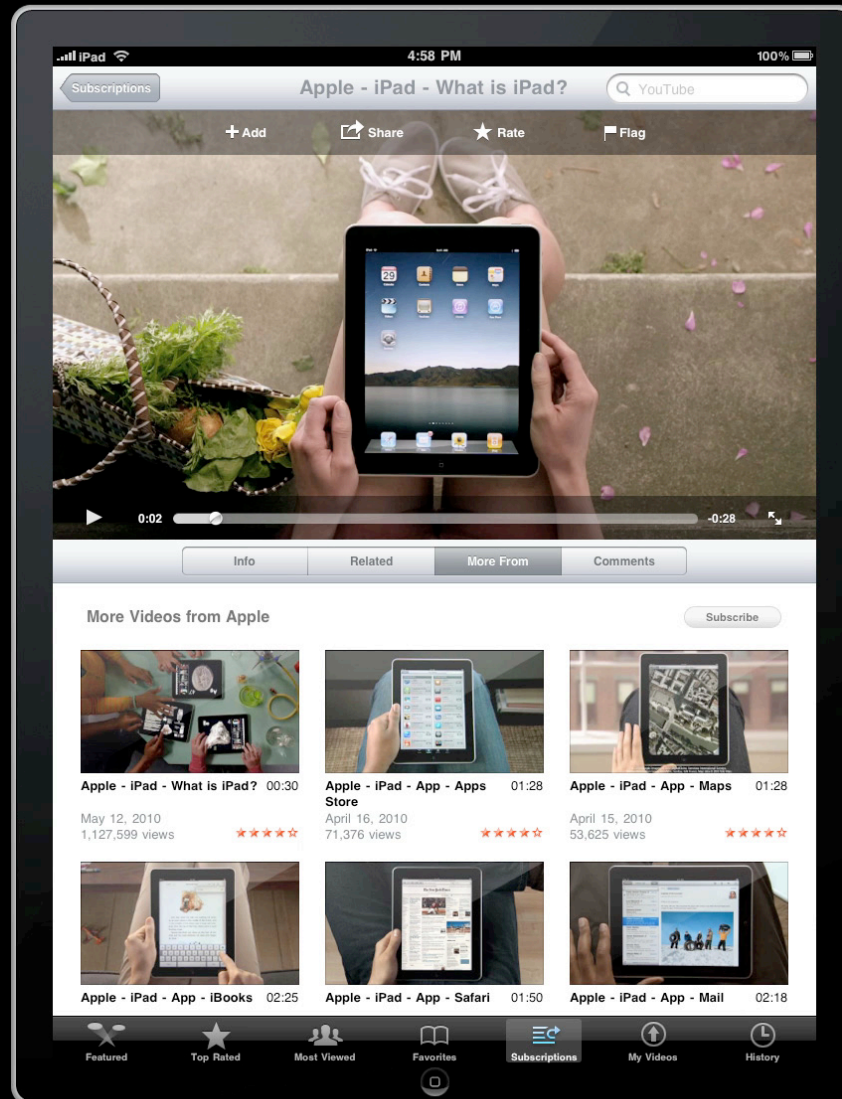
Demo

Tyler Hawkins

Window Orientation Animations

Window Orientation

- Support both portrait and landscape orientations
 - Users want and expect it
 - Keyboard is larger in landscape
 - Consider UI layout



Window Orientation

Animating using UIViewController

- Override point to enable rotation:
 - `(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)orientation;`
- Automatically manages:
 - Rotation transform
 - View frame
 - Animation
 - Status bar
 - User interaction

Window Orientation Animation

```
-(void)layoutSubviews
{
    BOOL isLandscape = UIInterfaceOrientationIsLandscape([[UIApplication
sharedApplication] statusBarOrientation]);
    if (isLandscape) {
        // Landscape layout here!
    } else {
        // Portrait layout here!
    }
}
```

Window Orientation Animation

Override points in UIViewController

`willRotate`

- Before rotation



`willAnimateRotation`

- Within animation



`didRotate`

- After rotation

Window Orientation Animation

Override points in UIViewController

– (void)**willRotate**ToInterfaceOrientation:(UIInterfaceOrientation)
toInterfaceOrientation duration:(NSTimeInterval)duration

– (void)**willAnimateRotation**ToInterfaceOrientation:(UIInterfaceOrientation)
toInterfaceOrientation duration:(NSTimeInterval)duration

– (void)**didRotate**FromInterfaceOrientation:(UIInterfaceOrientation)
fromInterfaceOrientation

Window Orientation

Without UIViewController

- Listen for device orientation changed notification
 - `UIDeviceOrientationDidChangeNotification`
- Manually manage:
 - Rotation transform
 - View frame
 - Animation
 - Status bar
 - Disabling of user interaction

Window Orientation

- UINavigationController makes it easy!

Maximize Animation Performance

Measuring Performance

- Test your app on real devices
 - Avoid simulator
 - Different device generations
- Use instruments

Performance Tips

- Avoid burdening processor
- Remove non-visible views
- Don't drawRect animations
 - Use UIView animations

Performance Tips

View hierarchy depth

Strawberry 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Rosales Family: Rosaceae Genus: Fragaria
Orange 	Kingdom: Plantae Angiosperms Eudicots	Order: Sapindales Family: Rutaceae Genus: Citrus
Apple 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Rosales Family: Rosaceae Genus: Malus
Banana 	Kingdom: Plantae Angiosperms Monocots	Order: Zingiberales Family: Musaceae Genus: Musa
Grape 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Vitales Family: Vitaceae Genus: Vitis
Kiwi 	Kingdom: Plantae Angiosperms Eudicots	Order: Ericales Family: Actinidiaceae Genus: Actinidia

Performance Tips

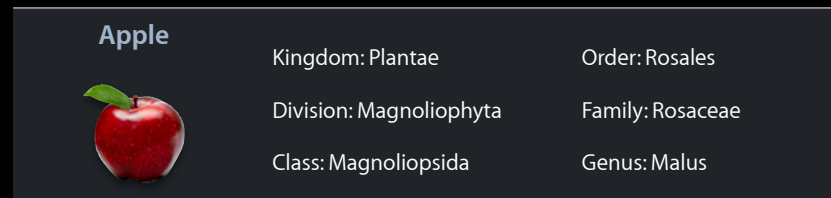
View hierarchy depth

- Reduce depth
- Less to composite, less memory

Strawberry 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Rosales Family: Rosaceae Genus: Fragaria
Orange 	Kingdom: Plantae Angiosperms Eudicots	Order: Sapindales Family: Rutaceae Genus: Citrus
Apple 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Rosales Family: Rosaceae Genus: Malus
Banana 	Kingdom: Plantae Angiosperms Monocots	Order: Zingiberales Family: Musaceae Genus: Musa
Grape 	Kingdom: Plantae Division: Magnoliophyta Class: Magnoliopsida	Order: Vitales Family: Vitaceae Genus: Vitis
Kiwi 	Kingdom: Plantae Angiosperms Eudicots	Order: Ericales Family: Actinidiaceae Genus: Actinidia

Performance Tips

- Sometimes a drawRect implementation is not an option
 - Animated subviews
 - UI control subviews
 - Complex view hierarchy



Performance Tips

shouldRasterize

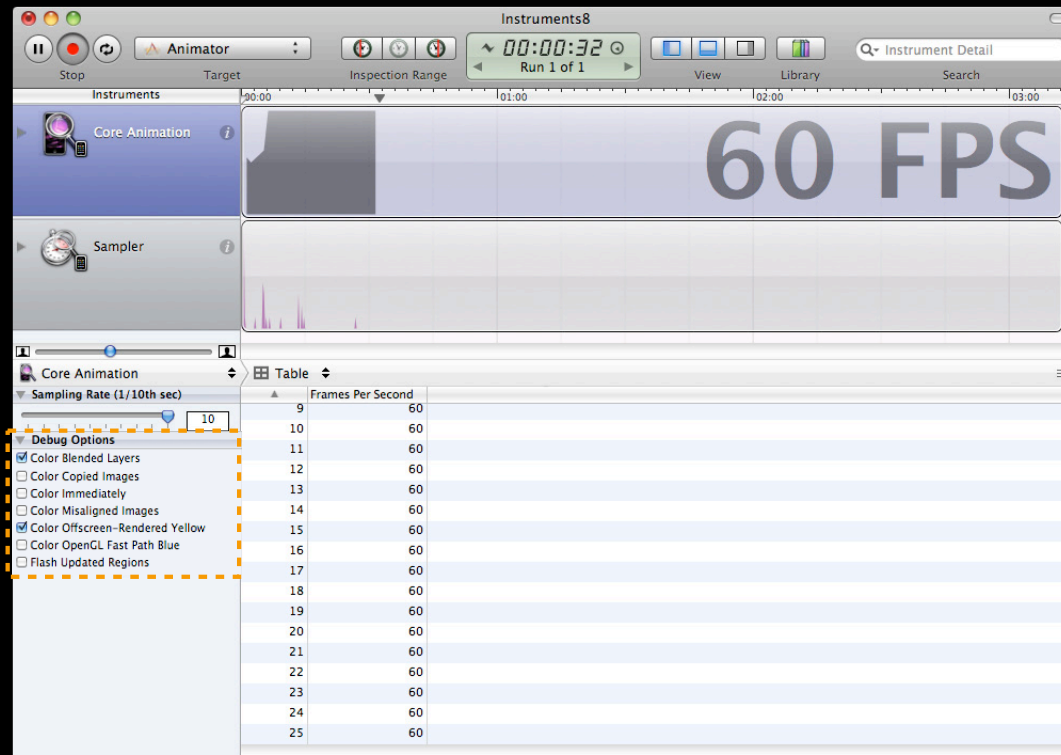
- Render offscreen and cache
- Can hurt performance
 - Animated subviews
 - Limited cache
- Judicial use can improve scroll animations
- Use it as little as possible
 - `_myView.layer.shouldRasterize = YES;`

Performance Tips

- Performance pitfalls:
 - Non-opaque layers
 - Layers requiring offscreen rendering

Measuring performance

Run with Performance Tool: CoreAnimation



Related Sessions

Advanced Performance Optimization on iPhone OS, Part 1

Mission
Thursday 3:15PM

Core Animation in Practice, Part 2

Nob Hill
Thursday 2:00PM

Advanced Animations Using CoreAnimation

Using CoreAnimation

- CoreAnimation Animatable Properties:
 - ex: zPosition, cornerRadius, borderWidth, borderColor
- Types of animations:
 - Implicit animations
 - Explicit animations using **CABasicAnimation**

Creating a CABasicAnimation

```
CABasicAnimation *myAnimation = [CABasicAnimation  
    animationWithKeyPath:@"opacity"];  
  
myAnimation.toValue = [NSNumber numberWithFloat:1];  
myAnimation.duration = 3.0;  
myAnimation.timingFunction = [CAMediaTimingFunction  
    functionName:kCAMediaTimingFunctionEaseInEaseOut];  
  
[_myView.layer addAnimation:myAnimation forKey:@"myAnimation"];
```



Creating a CABasicAnimation

```
CABasicAnimation *myAnimation = [CABasicAnimation  
    animationWithKeyPath:@"opacity"];  
  
myAnimation.toValue = [NSNumber numberWithFloat:1];  
myAnimation.duration = 3.0;  
myAnimation.timingFunction = [CAMediaTimingFunction  
    functionName:kCAMediaTimingFunctionEaseInEaseOut];  
  
[_myView.layer setOpacity:1];  
[_myView.layer addAnimation:myAnimation forKey:@"myAnimation"];
```



Motion Along a Path

- CAKeyFrameAnimation
- @property CGPathRef path

Demo

Tyler Hawkins

Related Sessions

CoreAnimation in Practice, Part 1

Nob Hill
Thursday 11:30AM

Core Animation in Practice, Part 2

Nob Hill
Thursday 2:00PM

Conclusion



- UIView Animations



- Blocks API



- CoreAnimation

More Information

Bill Dudney

Application Frameworks Evangelist
dudney@apple.com

Apple Developer Forums

<http://devforums.apple.com>



