

# Core Animation

DevDay for iPhone  
Daniel H Steinberg

# On Using Animation

- Why
- When
- How

# CA is not today's blink tag

# Should we leave it out?

# Use Animation

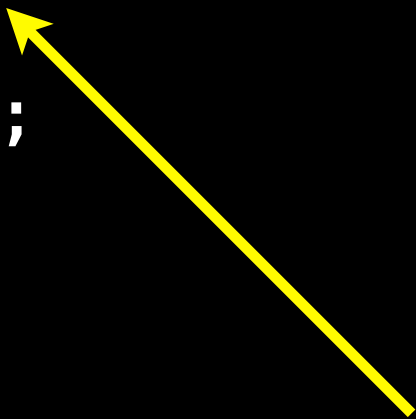
- To set context for the user
- To help the user make transitions
- To build a mental and physical model of the space in which they use your app
- To fit in with user expectation of apps that run on the platform

# Canonical Example

- Check out the various animations enabled in a standard Apple table view example.

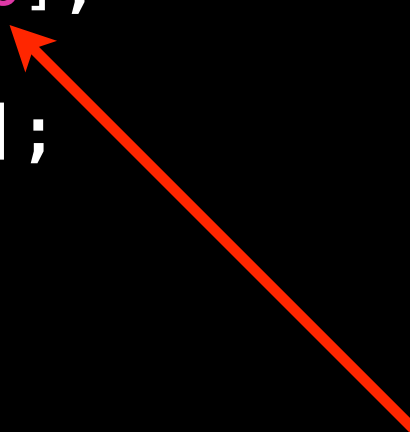
# Implementing the animation

```
- (void) tableView:(UITableView *)tableView  
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
  
    MethodController *detailViewController =  
        [[MethodController alloc] init];  
  
    [self.navigationController  
        pushViewController:detailViewController  
        animated:YES];  
  
    [detailViewController release];  
}
```



# Not Implementing the animation

```
- (void) tableView:(UITableView *)tableView  
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {  
  
    MethodController *detailViewController =  
        [[MethodController alloc] init];  
  
    [self.navigationController  
        pushViewController:detailViewController  
        animated:NO];  
  
    [detailViewController release];  
}
```





# Split-View Template

# On Using Animation

- Why
- When
- How

# Working with Layers

- iPhone views automatically have layers

# Displaying a layer


- Create LayerDemo - a new view based iPad project
- The VC's `imageLayer` will hold an image
- Set the size, position, and contents of the layer
- Add the `imageLayer` to our view's layer

# The Layer Property

```
#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>

@interface LayerDemoViewController : UIViewController
{
    CALayer *imageLayer;
}
@property(n nonatomic, retain) CALayer *imageLayer;

@end
```



- Foundation.framework
- UIKit.framework
- CoreGraphics.framework
- QuartzCore.framework

# Adding the sublayer

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    [self.view.layer addSublayer:self.imageLayer];  
}
```

# Configuring the layer

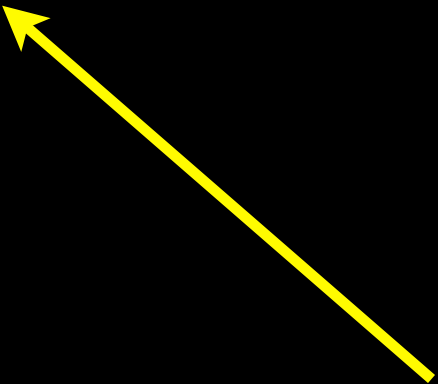
```
-(CALayer *) imageLayer {
    if (nil == imageLayer) {
        self.imageLayer = [CALayer layer];
        imageLayer.contents =
            (id)( [UIImage imageNamed:@"devday.png"].CGImage);
        imageLayer.position =
            CGPointMake(self.view.bounds.size.width/2,
                        self.view.bounds.size.height/2);
        imageLayer.bounds = CGRectMake(0.0, 0.0, 400.0, 200.0);
    }
    return imageLayer;
}
```

Something looks wrong



# Configuring the layer

```
-(CALayer *) imageLayer {  
    if (nil == imageLayer) {  
        self.imageLayer = [CALayer layer];  
        imageLayer.contents =  
            (id)( [UIImage imageNamed:@"devday.png"].CGImage);  
        imageLayer.position =  
            CGPointMake(self.view.bounds.size.width/2,  
                        self.view.bounds.size.height/2);  
        imageLayer.bounds = CGRectMake(0.0, 0.0, 400.0, 200.0);  
        imageLayer.contentsGravity = kCAGravityResizeAspect;  
    }  
    return imageLayer;  
}
```



That's better

# Add an animation

- For effect made initial position lower and smaller and start with another image.
- When screen is tapped move the image up, swap images, and grow it to full width

# Really?

```
- (void)touchesBegan:(NSSet *)touches
    withEvent:(UIEvent *)event {
    self.imageLayer.contents =
        (id)( [UIImage imageNamed:@"devday.png"].CGImage);
    self.imageLayer.frame =
        CGRectMake(0.0, 0.0, self.view.bounds.size.width,
                    self.view.bounds.size.height/6);
}
```

What do you get for  
free?

# Explicit Animation

```
-(CABasicAnimation *) moveAnimation {  
    if (nil == moveAnimation) {  
        self.moveAnimation =  
            [CABasicAnimation animationWithKeyPath:@"position"];  
        self.moveAnimation.duration = 3.0;  
        self.moveAnimation.timingFunction = [CAMediaTimingFunction  
            functionWithName:kCAMediaTimingFunctionEaseIn];  
    }  
    return moveAnimation;  
}
```

# Explicit Animation

```
if (nil == imageLayer) {  
    self.imageLayer = [CALayer layer];  
    imageLayer.contents =  
        (id)( [UIImage imageNamed:@"ready.png"].CGImage);  
    imageLayer.position = CGPointMake(self.view.bounds.size.width/2,  
                                     9*self.view.bounds.size.height/10);  
    imageLayer.bounds = CGRectMake(0.0, 0.0, 300.0, 200.0);  
    imageLayer.contentsGravity = kCAGravityResizeAspect;  
    imageLayer.actions =  
        [NSDictionary dictionaryWithObject: self.moveAnimation  
                                     forKey:@"position"];  
}
```

# Now look!

- Changed the timing
- Changed the timing function
- Can customize the animation



# Other Paths

# Calling Animations

- We'll add a rotation animation
- We'll invoke this animation at startup

# Rotation Animation

```
-(CABasicAnimation *) rotationAnimation {
    if (nil == rotationAnimation) {
        self.rotationAnimation = [CABasicAnimation
                                   animationWithKeyPath:@"transform.rotation"];
        rotationAnimation.repeatCount = 6;
        rotationAnimation.fromValue =
            [NSNumber numberWithFloat:0.0];
        rotationAnimation.toValue =
            [NSNumber numberWithFloat:2.0 * M_PI];
    }
    return rotationAnimation;
}
```

# Adding the Animation

```
if (nil == imageLayer) {
    self.imageLayer = [CALayer layer];
    imageLayer.contents =
        (id)( [UIImage imageNamed:@"ready.png"].CGImage);
    imageLayer.position = CGPointMake(self.view.bounds.size.width/2,
                                     9*self.view.bounds.size.height/10);
    imageLayer.bounds = CGRectMake(0.0, 0.0, 300.0, 200.0);
    imageLayer.contentsGravity = kCAGravityResizeAspect;
    [imageLayer addAnimation:self.rotationAnimation
                      forKey:@"rotationAnimation"];

    imageLayer.actions =
        [NSDictionary dictionaryWithObject: self.moveAnimation
                                   forKey:@"position"];
}
```

# Location matters

- Run the app
- Move the `addAnimation:forKey:` call down to `touchesBegan` and rerun

# And beyond ...

- Code and slides:
- <http://dimsumthinking.com/services/code/>