

CS 696 Mobile Phone Application Development
Fall Semester, 2010
Doc 23 Animation
Nov 23, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

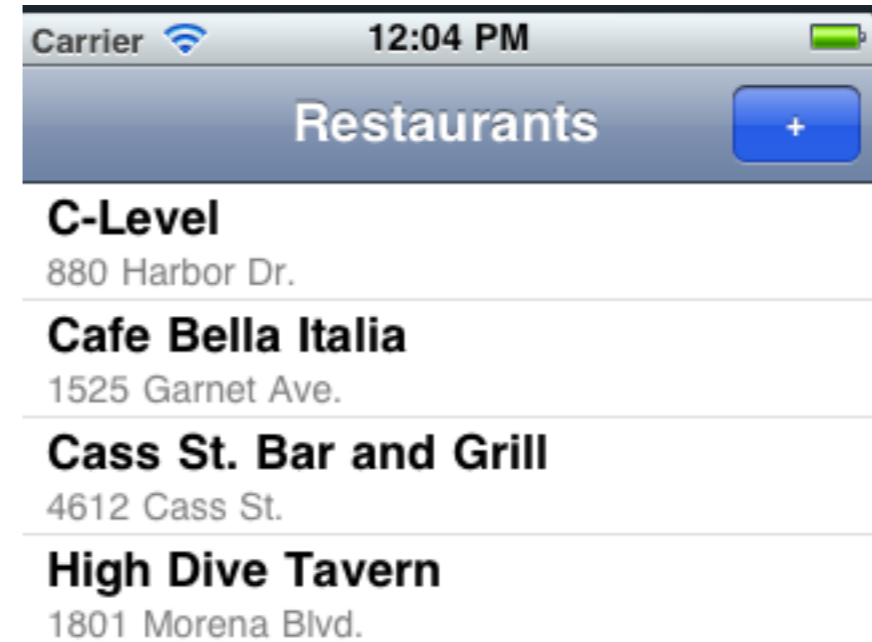
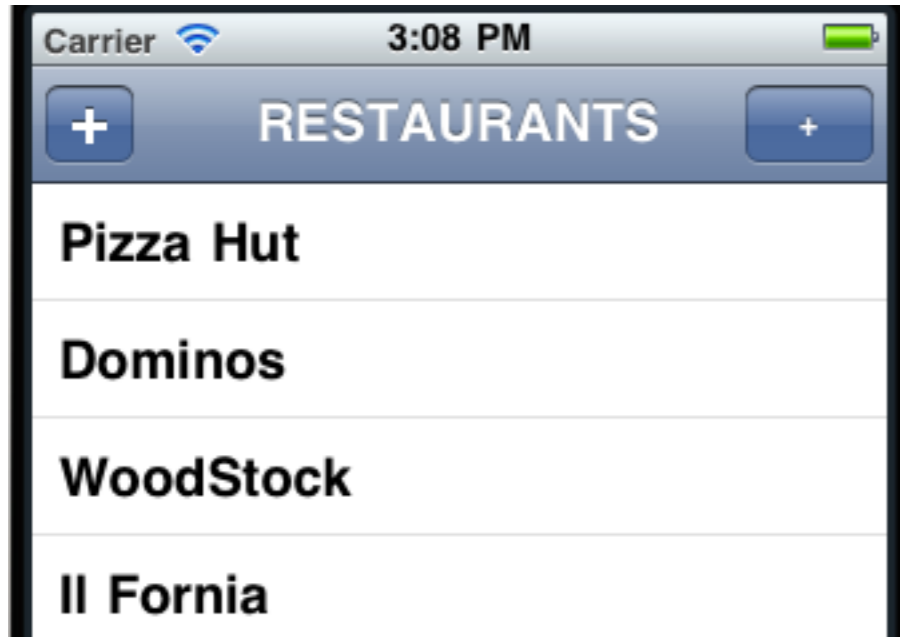
iPhone Programming: The Big Nerd Ranch Guide, Conway & Hillgass, Big Nerd Ranch, 2010
Chapters 18 & 19

Core Animation Programming Guide, Apple Documentation

Assignment 4 Comments



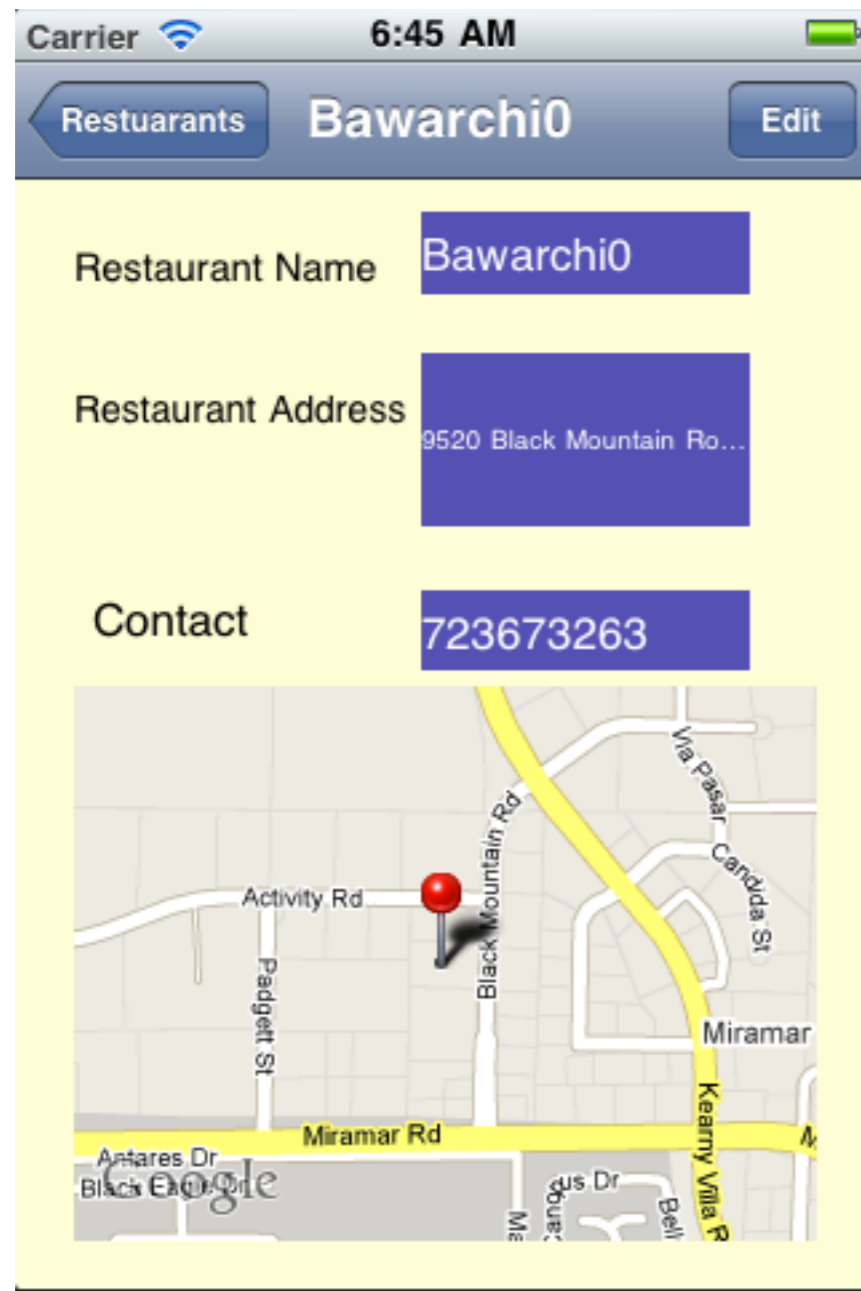
Add



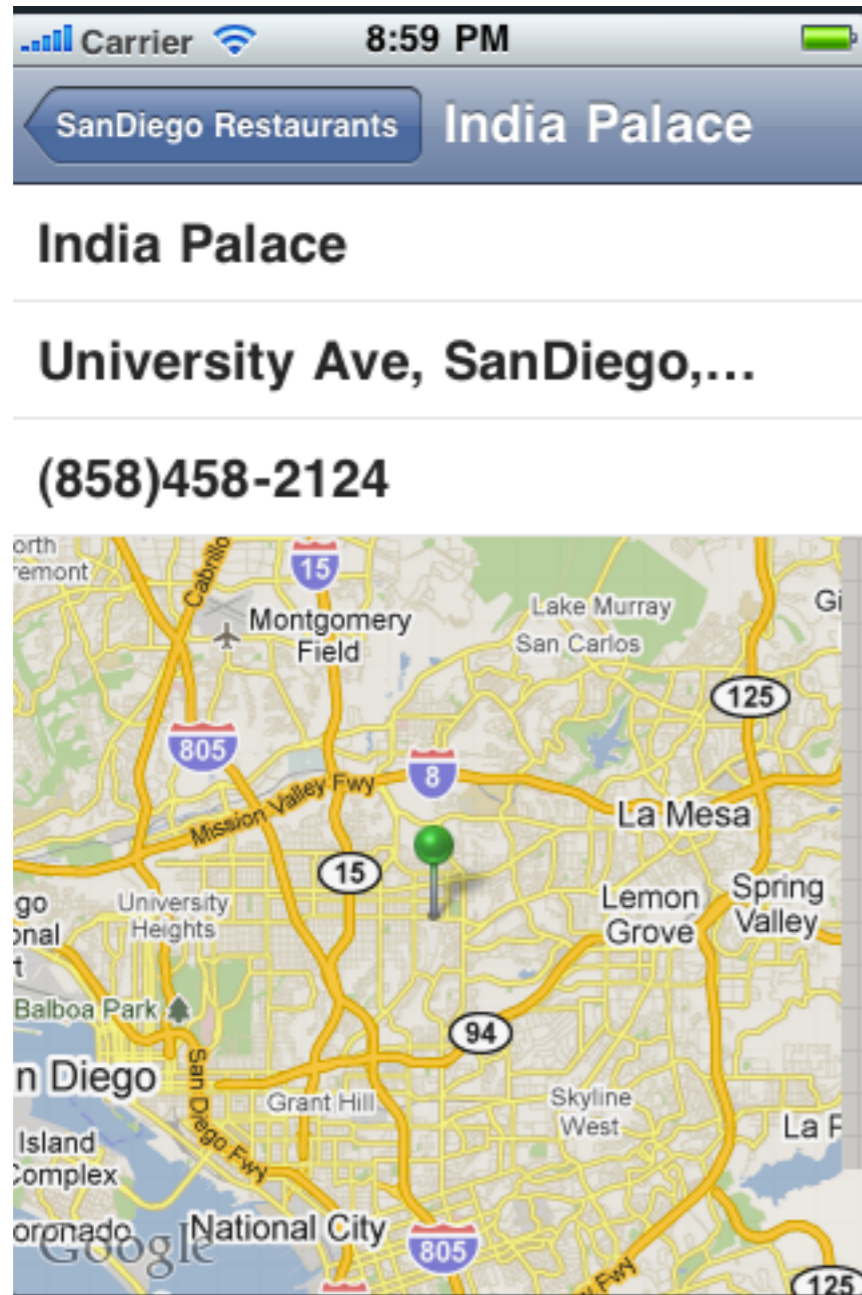
Order

The image shows a screenshot of an iPhone application interface for adding a restaurant. At the top, the status bar displays 'Carrier', a Wi-Fi signal icon, the time '2:37 PM', and a battery level icon. Below the status bar is a header bar with a 'Cancel' button on the left, the title 'Add Restaurant' in the center, and a 'Save' button on the right. The main content area contains several text input fields: 'Restaurant Name:' with a placeholder 'Restaurant Name', 'Street:' with a placeholder 'Street', 'Phone:' with a placeholder 'Phone', 'Zipcode:' with a placeholder 'Zipcode', 'City:' with a placeholder 'City', and 'State:' with a placeholder 'State'. The form is styled with a light gray background and rounded rectangular input fields.

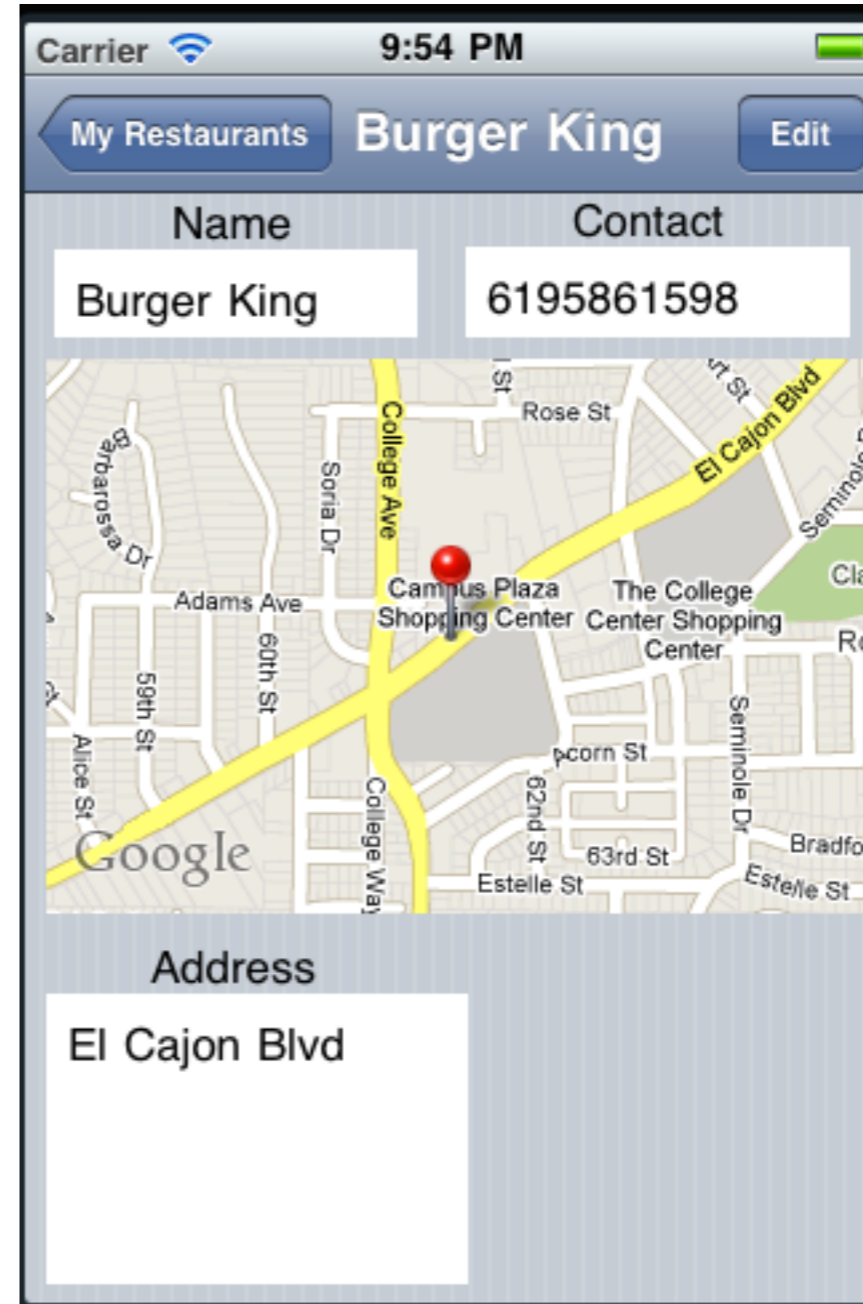
Nice Map, What the address?



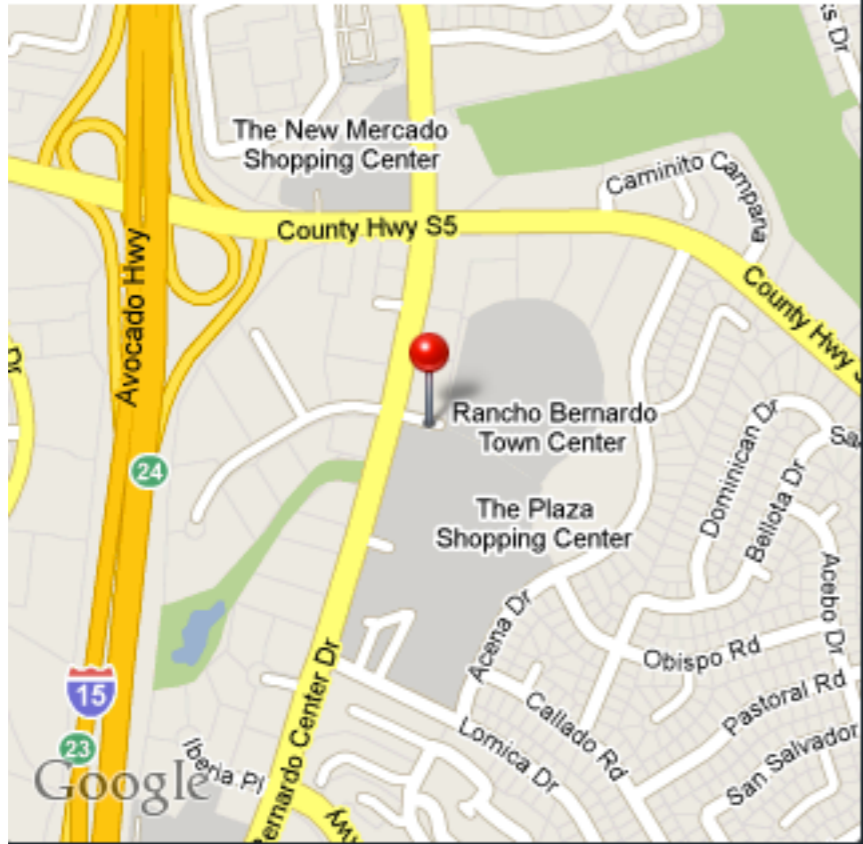
Is that a phone number?



...

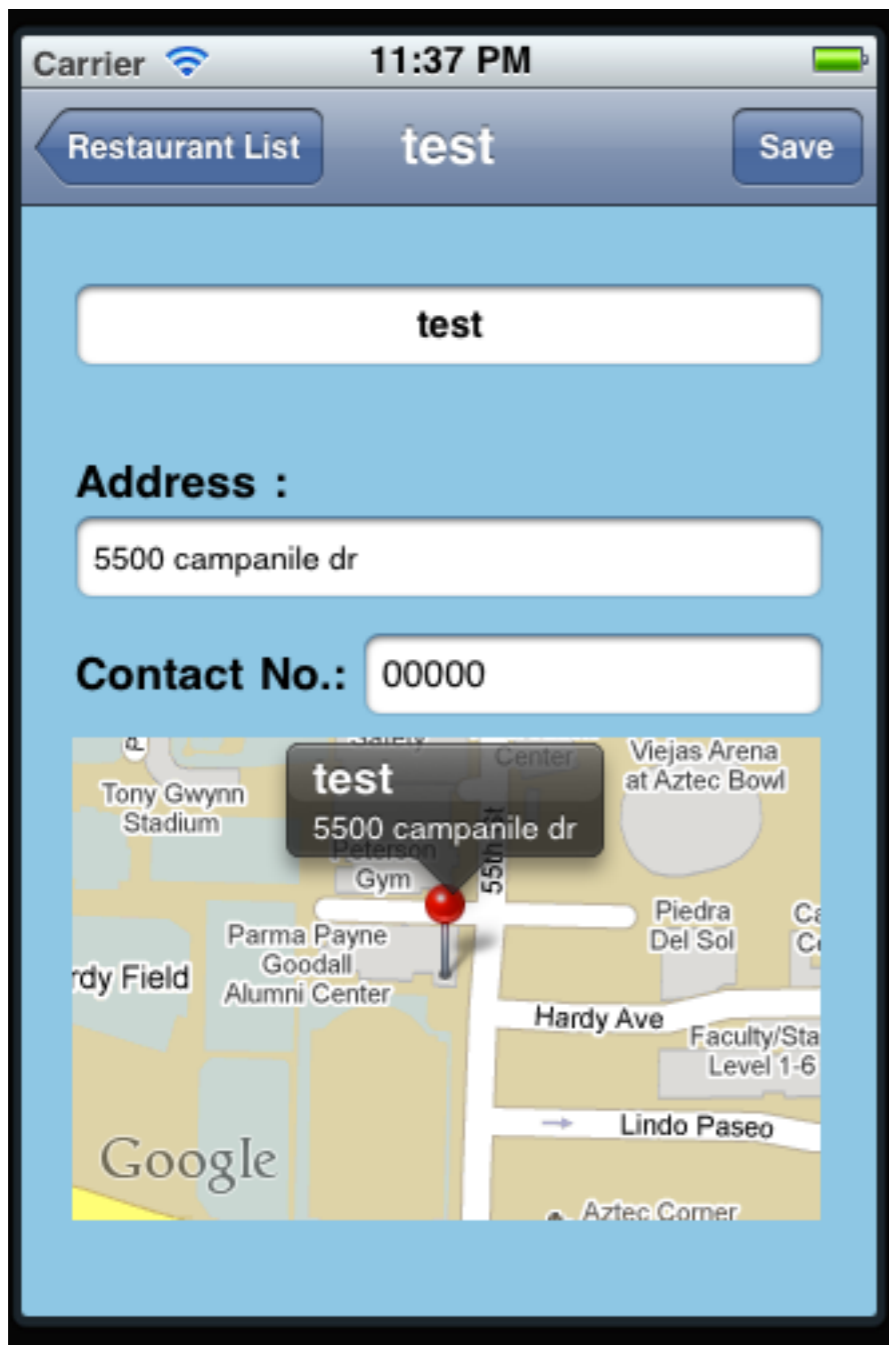


Pizza Hut
11975 Bernardo Plaza Dr
San Diego CA 92127 USA
Phone no.:8584543433



Double Trouble





Carrier  6:42 AM 

Name

Phone

Address

Carrier  9:53 PM 

Add Restaurant

Name Contact

Address

3?

[Restaurants](#) **Chicken N Things**

[Map](#) [Edit](#)

Name 3

Chicken N Things

Address

1234 Carroll Canyon Rd. 92131

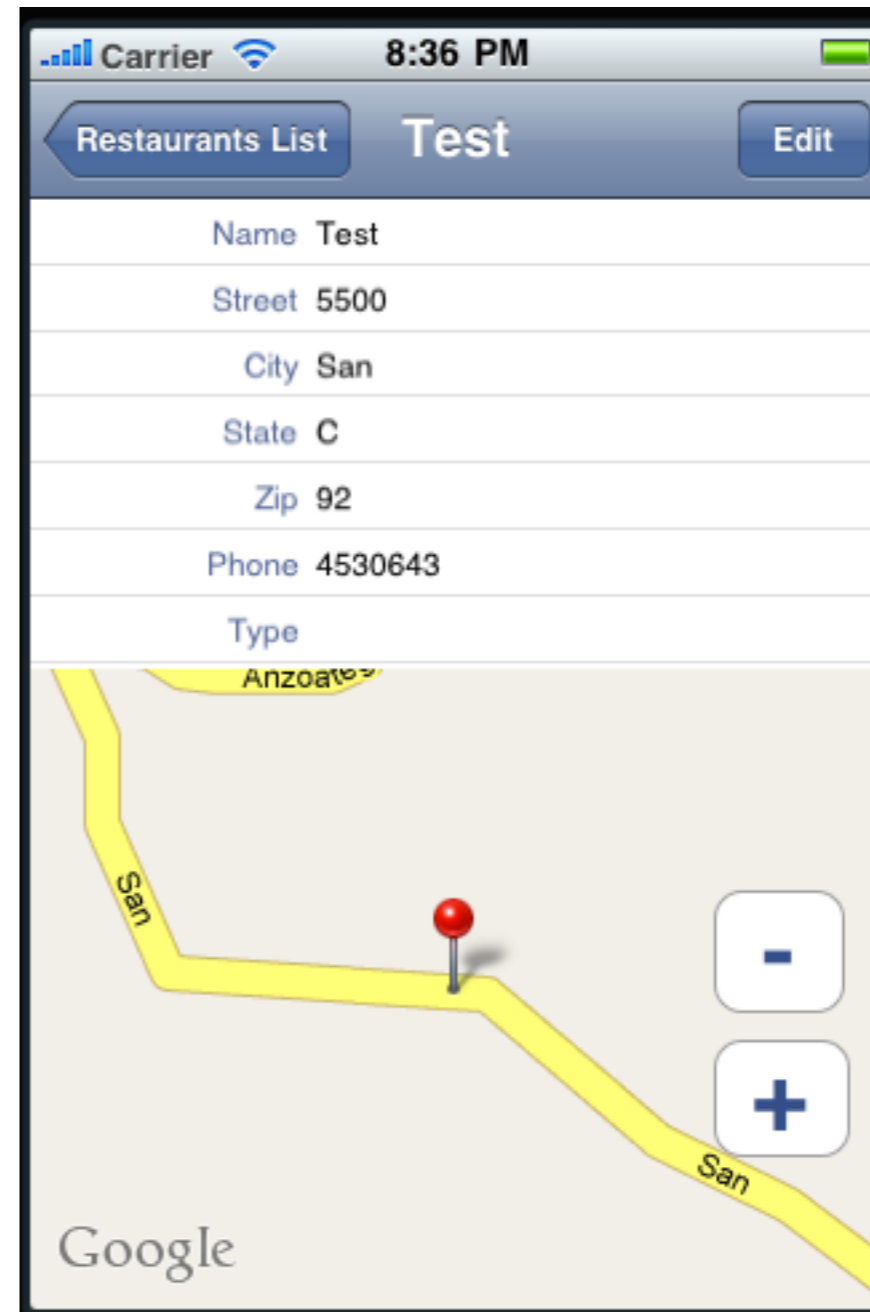
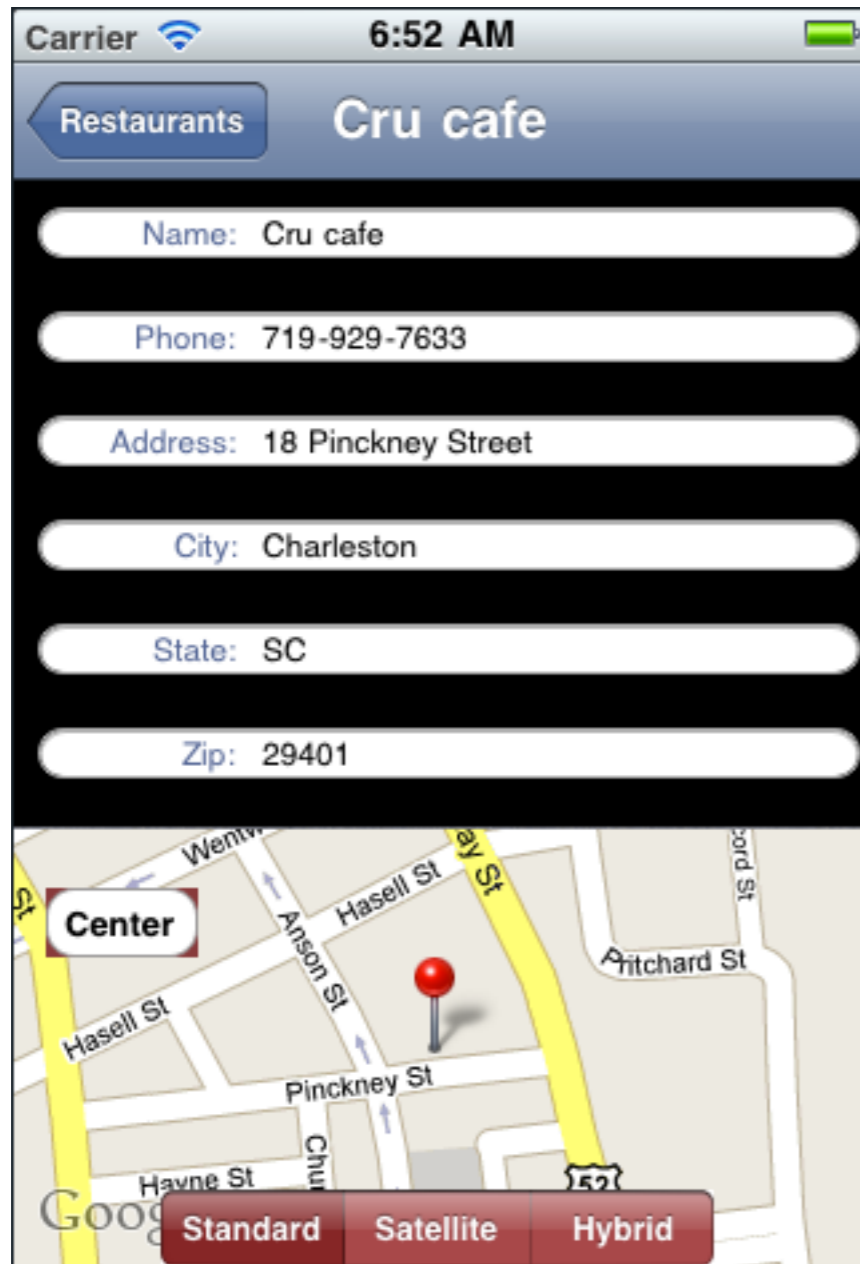
Phone

619 234 5434

Title



A bit small for eyes and fingers



Animation

How to do this?



Main Animation Classes

CALayer

- Buffer with bitmap

- Draw on bitmap

- Then rendered on screen

- Rendering hardware-accelerated

- Each view has one or more CALayers

- All drawing/images done on CALayer

CAAnimation

- Changes view over time

- Changes on property of CALayer at a time

- CABasicAnimation

- CAKeyframeAnimation

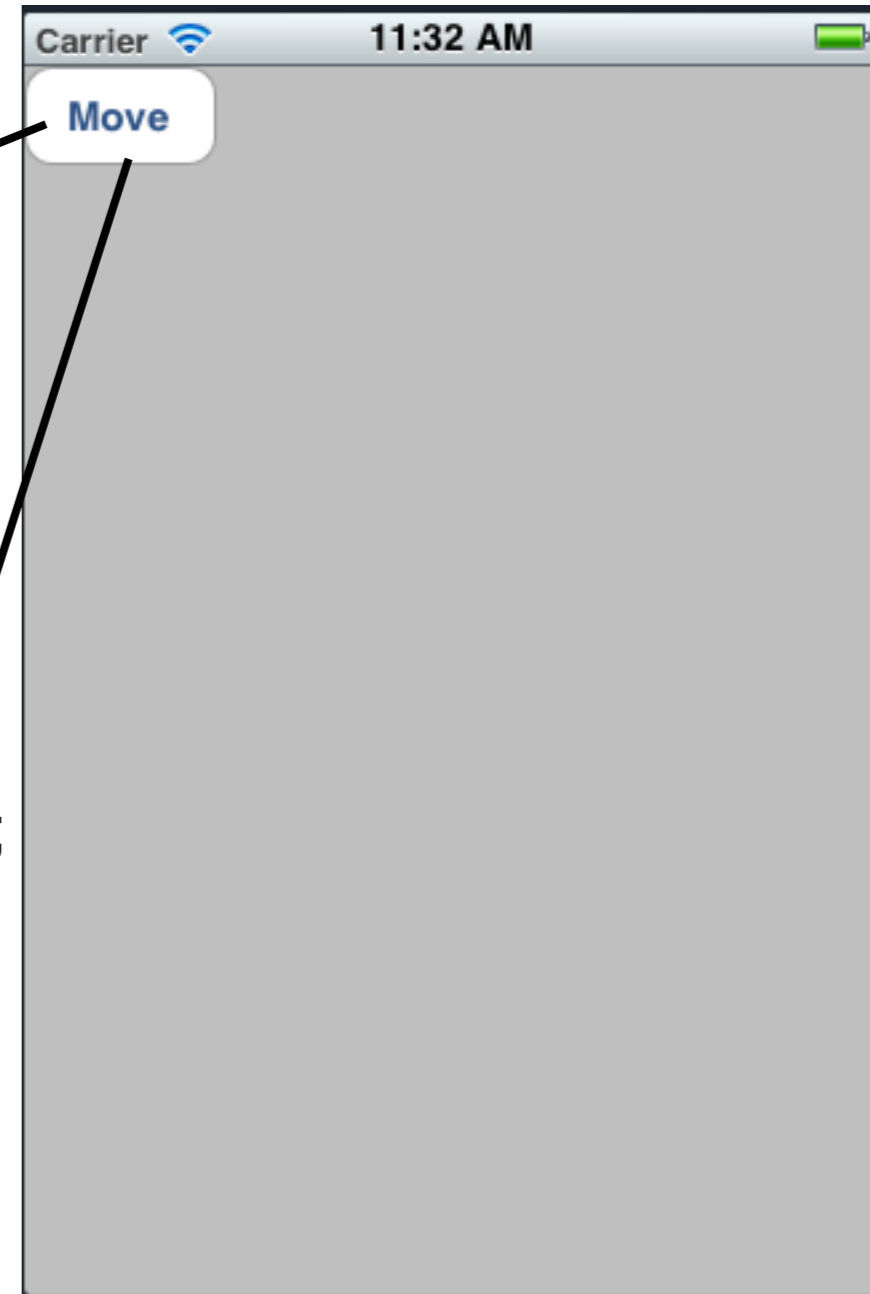
- CAAnimationGroup

QuartzCore.framework

To use Core Animation you need to add QuartzCore.framework to your project

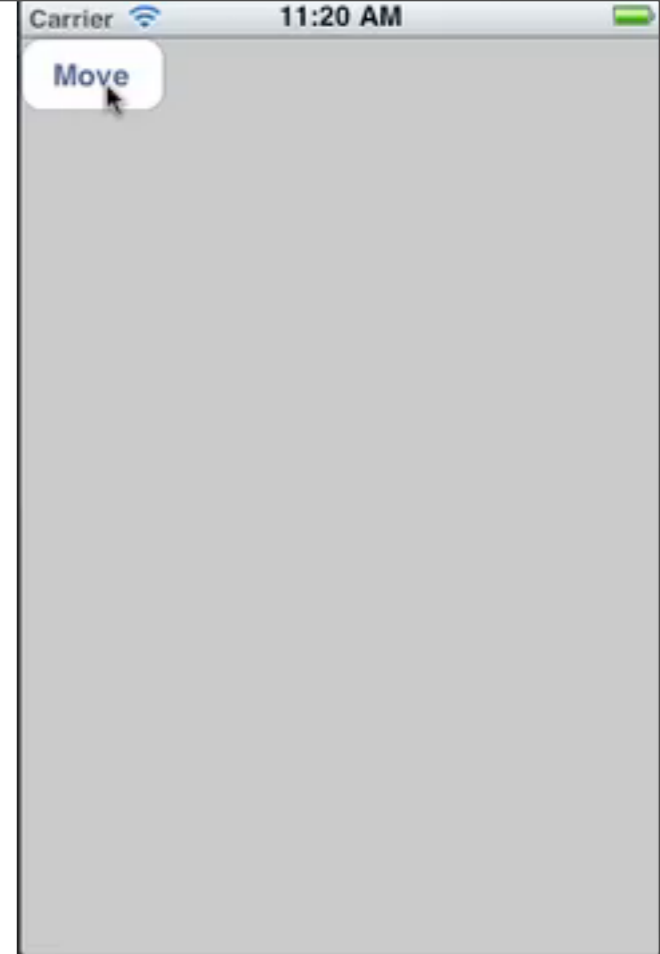
Setup For Simple Examples

```
@interface AnimationLayersViewController :  
    UIViewController {  
}  
  
-(IBAction) move;  
  
@property (nonatomic, retain) IBOutlet UIButton* moveButton;  
  
@end
```



Default Move

```
- (void) move {  
    moveButton.center = CGPointMake(285, 440);  
}
```

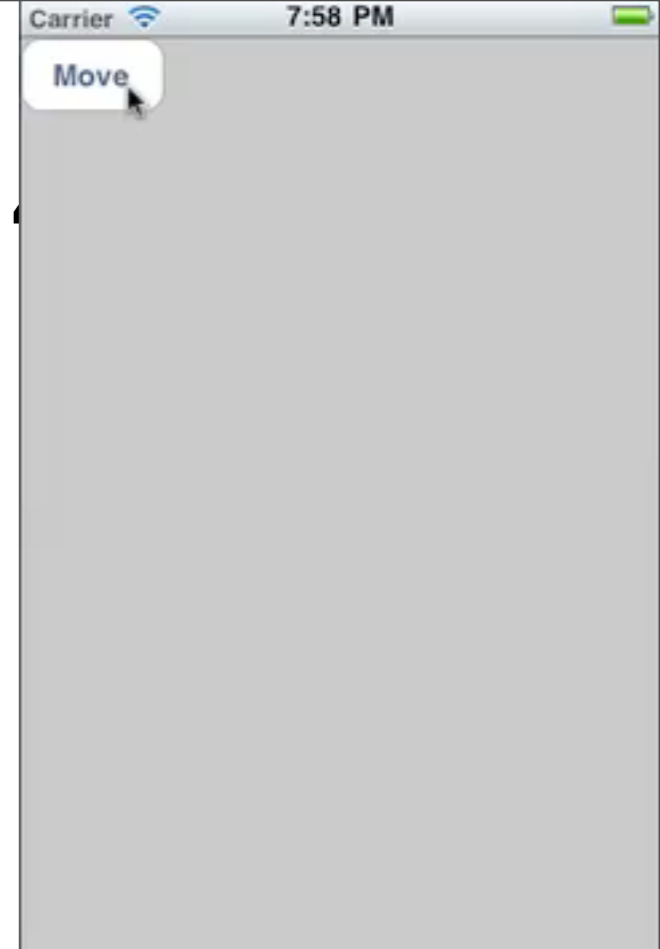


Implicit Animations

All changes to CALayer have implicit animations

May not do what you want

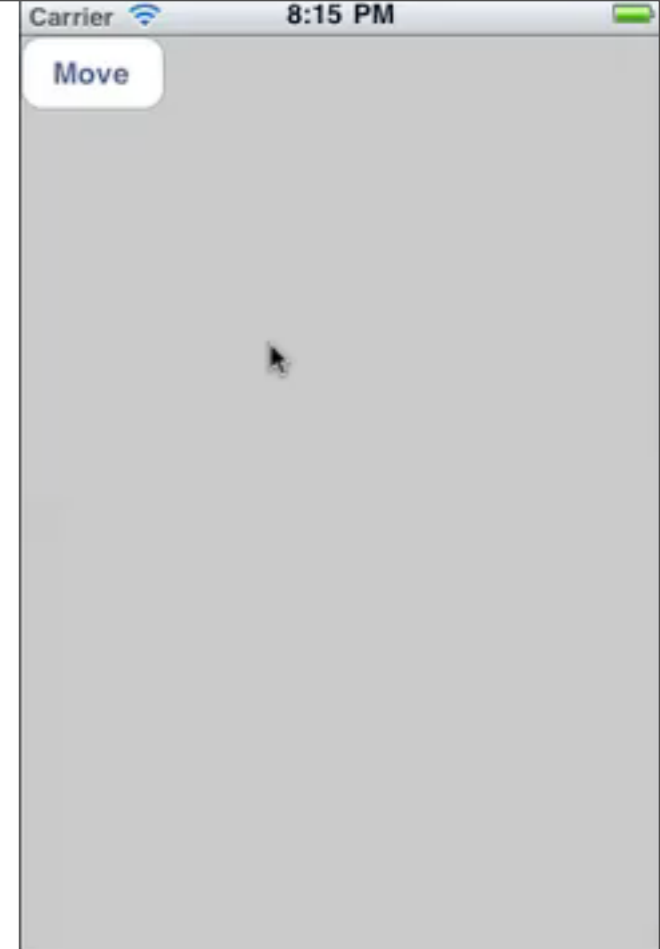
Animation Transactions Pre iOS 4



```
- (void) move {  
    [UIView beginAnimations:nil context:NULL];  
    [UIView setAnimationDuration:0.5];  
    moveButton.center = CGPointMake(285, 440);  
    [UIView commitAnimations];  
}
```

Block Animations

```
- (void) move {  
    CALayer * moveLayer = moveButton.layer;  
    [UIView animateWithDuration:1.0 animations:^(  
        moveButton.center = CGPointMake(285, 440);  
    )];  
}
```

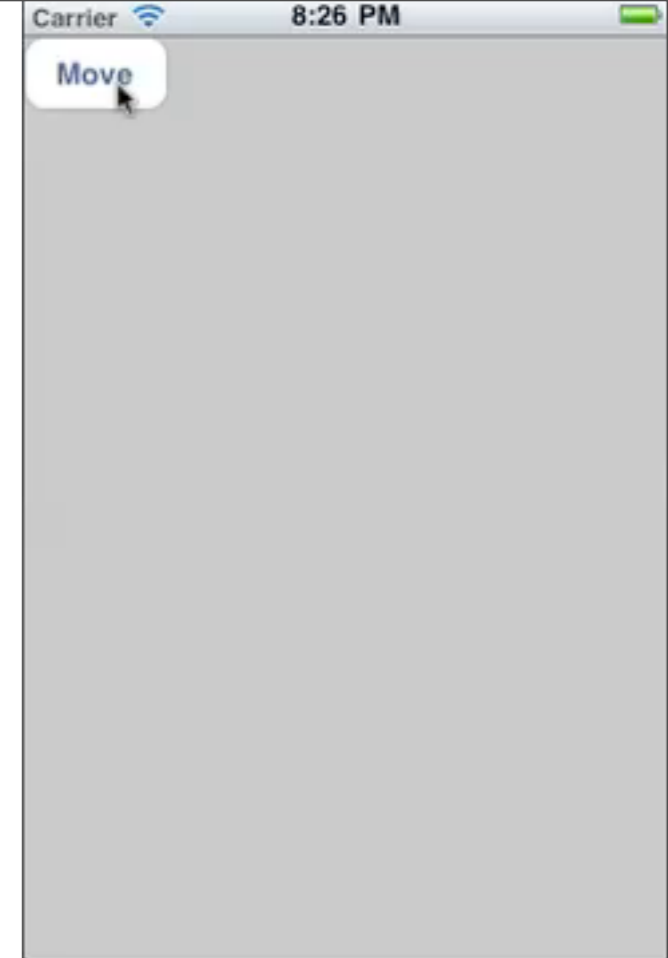


Block Animation method

- + `animateWithDuration:delay:options:animations:completion:`
- + `animateWithDuration:animations:completion:`
- + `animateWithDuration:animations:`
- + `transitionWithView:duration:options:animations:completion:`
- + `transitionFromView:toView:duration:options:completion:`

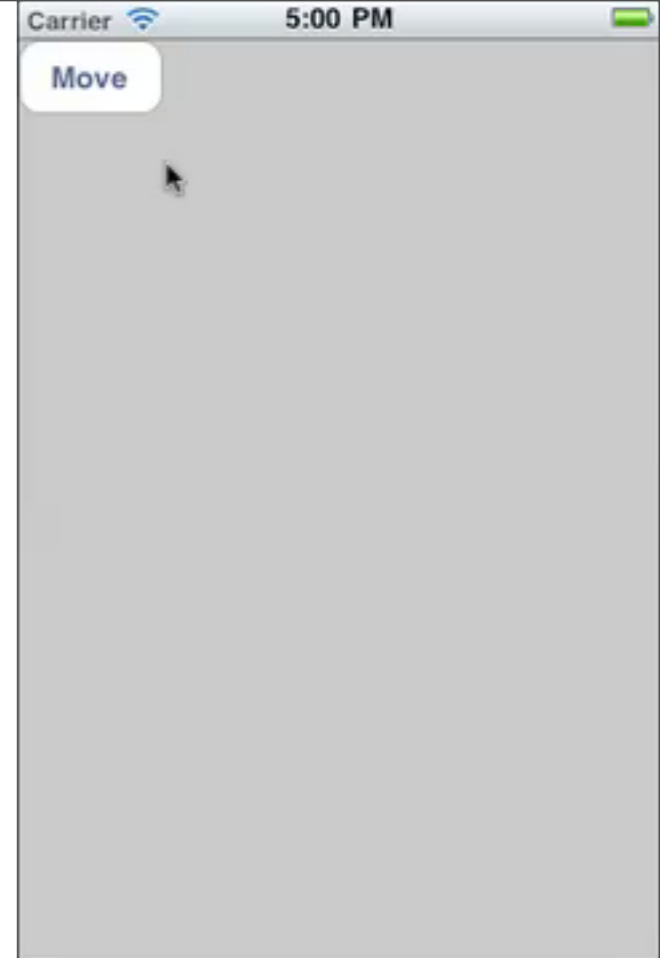
Block Animation with Return

```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    [UIView animateWithDuration:1.0 animations:^(
        moveButton.center = CGPointMake(285, 440);
    )
    completion:^(BOOL finished) {
        [UIView animateWithDuration:1.0 animations:^(
            moveButton.center = CGPointMake(36, 18);
        )];
    }];
}
```



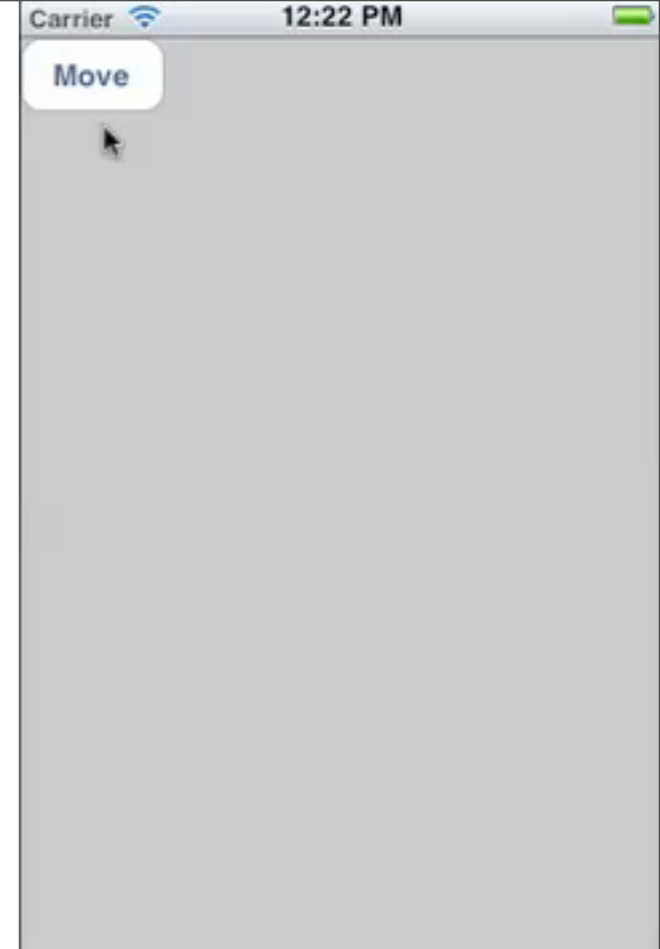
Spin

```
- (void) move {  
    CALayer * moveLayer = moveButton.layer;  
    CABasicAnimation * spin = [CABasicAnimation  
        animationWithKeyPath:@"transform.rotation"];  
    [spin setValue:[NSNumber numberWithFloat:M_PI *  
2.0]];  
    spin.duration = 1.0;  
    [moveLayer addAnimation: spin forKey:@"both"];  
}
```



```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    [UIView beginAnimations:nil context:NULL];
    [UIView setAnimationDuration:0.5];
    CGAffineTransform transform = CGAffineTransformMakeRotation(M_PI);
    moveButton.transform = transform;
    [UIView commitAnimations];
    [UIView beginAnimations:nil context:NULL];
    [UIView setAnimationDuration:0.5];
    transform = CGAffineTransformMakeRotation(M_PI * 2);
    moveButton.transform = transform;
    [UIView commitAnimations];
}
```

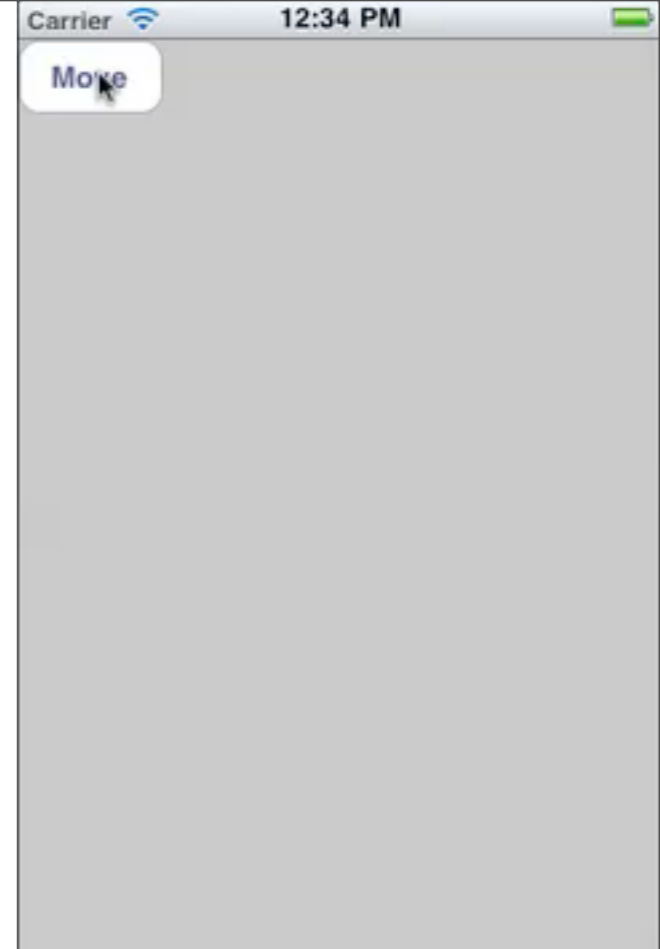
Explicit Animation



```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
[CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    [moveLayer addAnimation:mover forKey:@"moving"];
}
```

Changing End Position 1

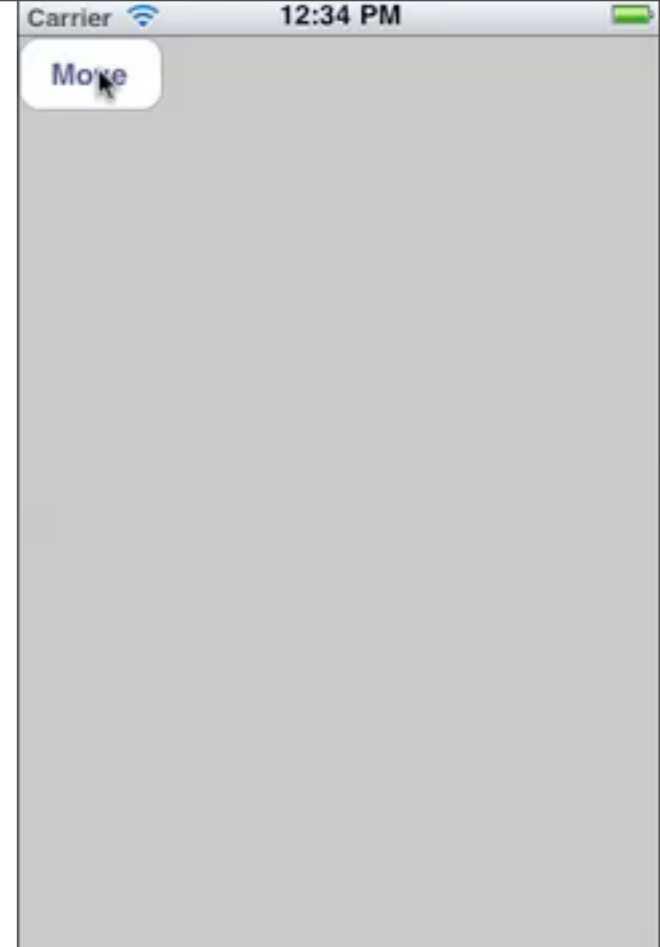
```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
        [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    [moveLayer addAnimation:mover forKey:@"moving"];
    moveLayer.position = CGPointMake(285, 440);
}
```



Using Delegates

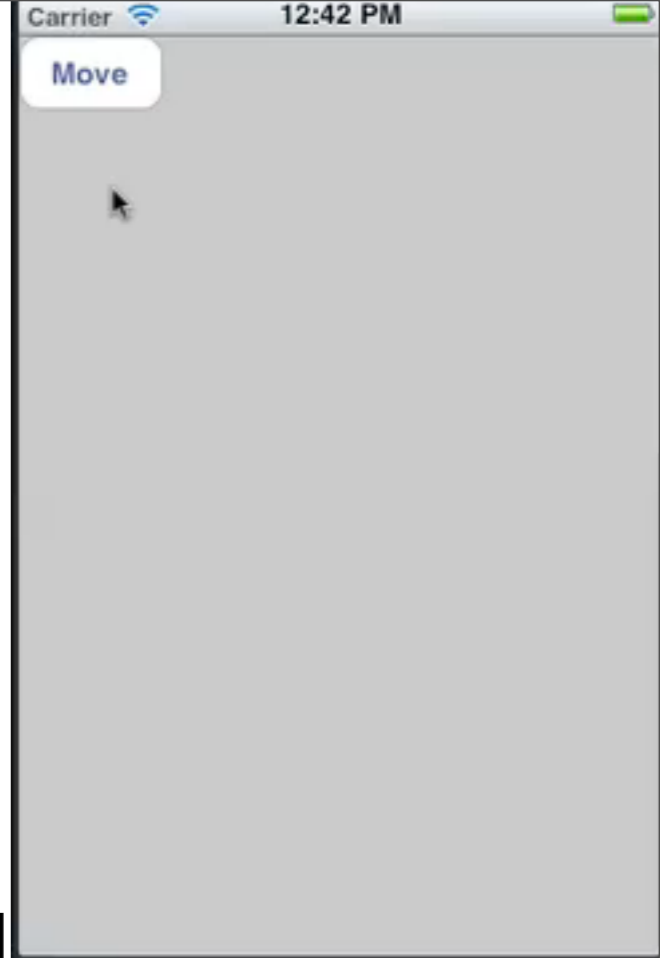
```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
[CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.delegate = self;
    [moveLayer addAnimation:mover forKey:@"moving"];
}

- (void) animationDidStop:(CAAnimation *)animation finished:(BOOL)flag {
    moveButton.layer.position = CGPointMake(285, 440);
}
```



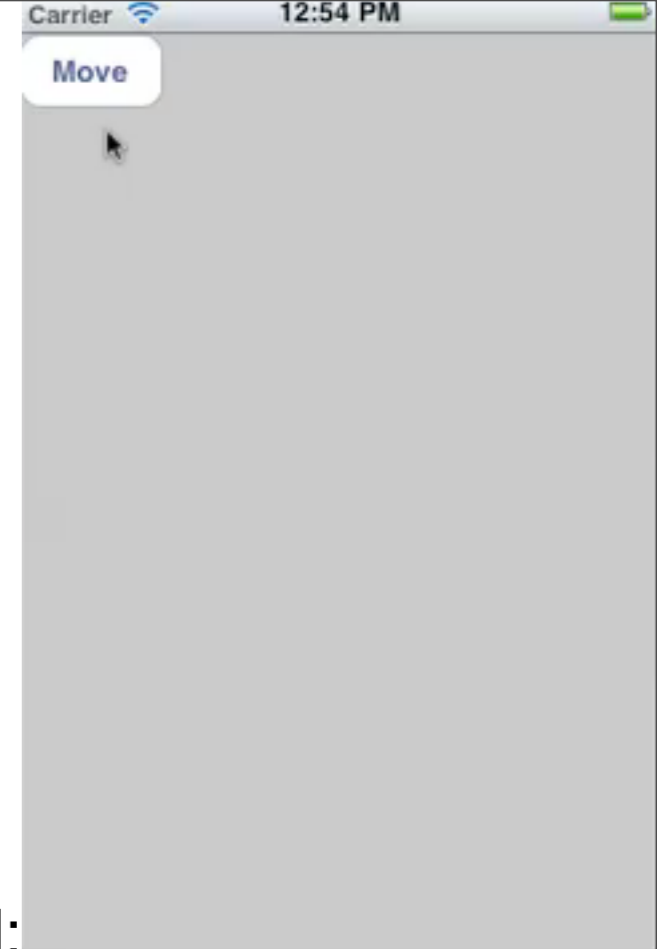
Timing Function

```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
    [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    CAMediaTimingFunction *easeInOut =
    [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    [moveLayer addAnimation:mover forKey:@"moving"];
    moveLayer.position = CGPointMake(285, 440);
}
```



Repeats

```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
[CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.repeatDuration = 6.0;
    CAMediaTimingFunction *easeInOut =
[CAMediaTimingFunction functionName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    [moveLayer addAnimation:mover forKey:@"moving"];
    moveLayer.position = CGPointMake(285, 440);
}
```



Reverse

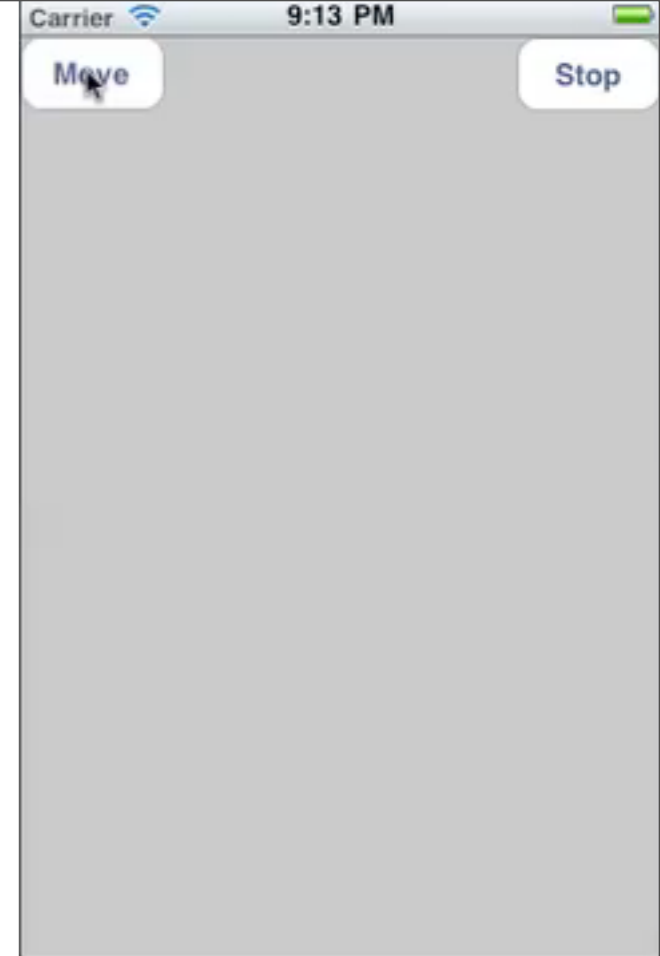


```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
[CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.5;
    mover.fromValue = [NSValue valueWithCGPoint:moveLayer.position],
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.repeatDuration = 4.5;
    mover.autoreverses = YES;
    CAMediaTimingFunction *easeInOut =
[CAMediaTimingFunction functionName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    [moveLayer addAnimation:mover forKey:@"moving"];
    moveLayer.position = CGPointMake(285, 440);
}
```

Ping Pong

```
- (void) move {
    CAAnimation * move = [self moveDown];
    CALayer * moveLayer = moveButton.layer;
    [moveLayer addAnimation: move
    forKey:@"both"];
}

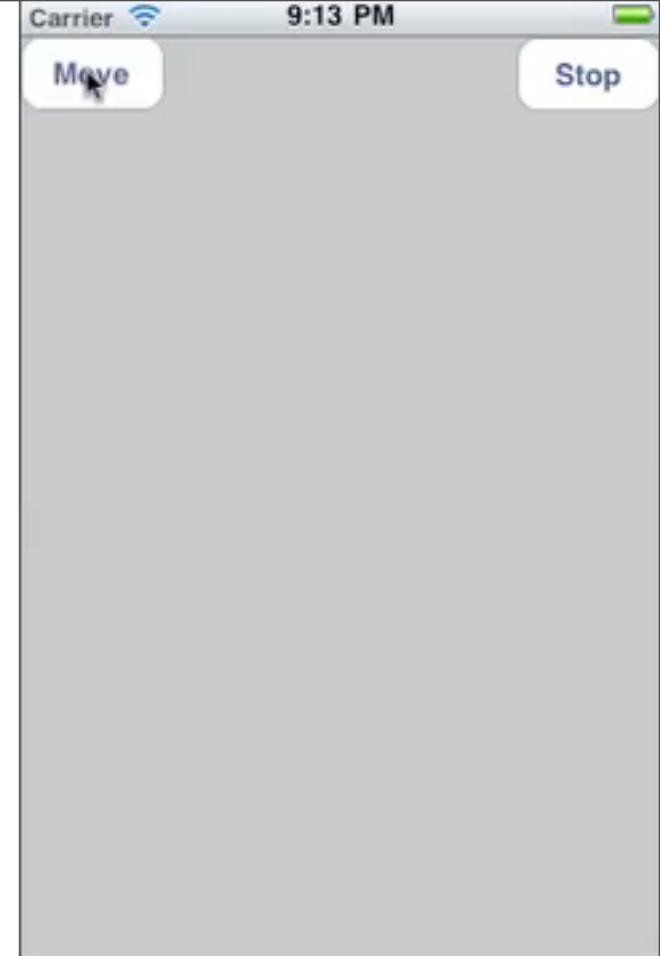
- (CAAnimation *) moveDown {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover = [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.0;
    mover.fromValue = [NSValue valueWithCGPoint: CGPointMake(36, 18)];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.delegate = self;
    CAMediaTimingFunction *easeInOut =
        [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    moveLayer.position = CGPointMake(285, 440);
    return mover;
}
```



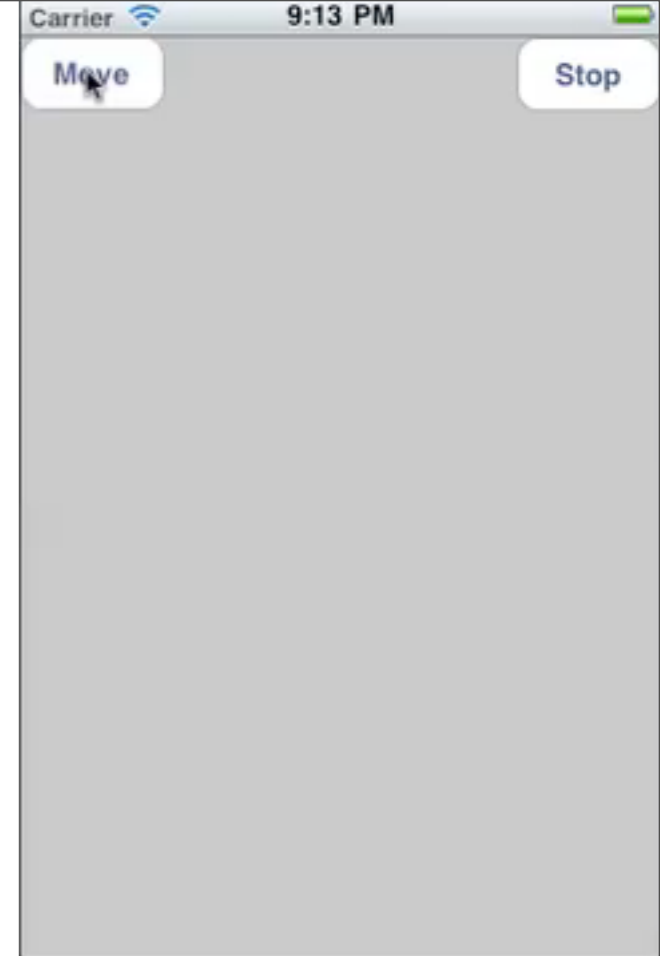
Ping Pong

```
- (void) animationDidStop:(CAAnimation *)animation
    finished:(BOOL)didFinish {
    if (!didFinish) {
        return;
    };
};
```

```
CALayer * moveLayer = moveButton.layer;
if (moveLayer.position.x == 285) {
    [moveLayer addAnimation: [self moveUp] forKey:@"up"];
} else {
    [moveLayer addAnimation: [self moveDown] forKey:@"down"];
}
}
```



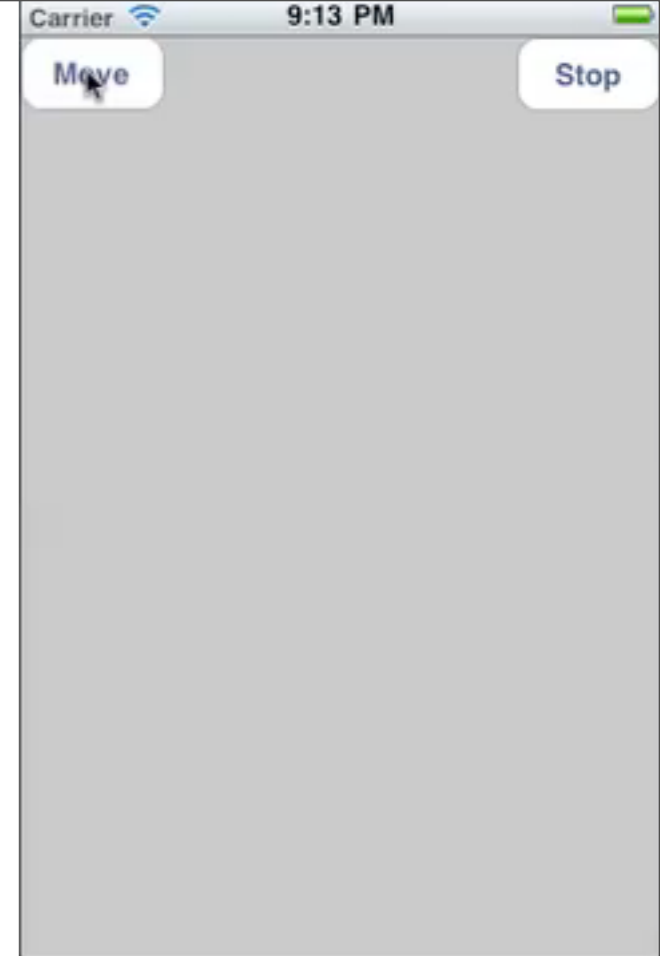
Ping Pong



```
- (CAAnimation *) moveUp {
    CALayer * moveLayer = moveButton.layer;
    CABasicAnimation * mover =
        [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.0;
    mover.fromValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(36, 18)];
    mover.delegate = self;
    CAMediaTimingFunction *easeInOut =
        [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    moveLayer.position = CGPointMake(36, 18);
    return mover;
}
```

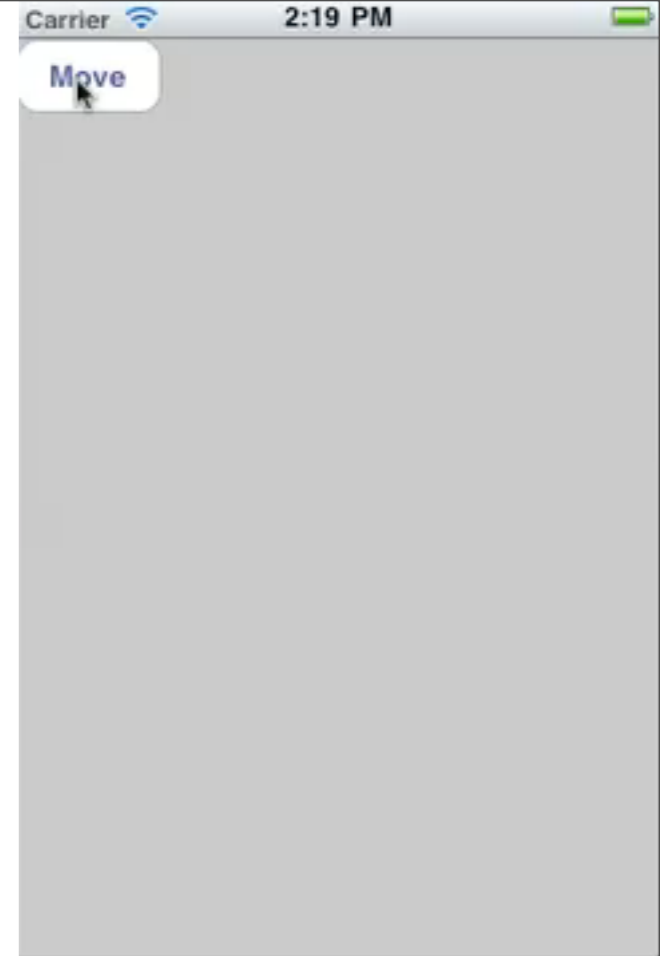
Ping Pong

```
- (IBAction) stop {  
    [moveButton.layer removeAllAnimations];  
}
```



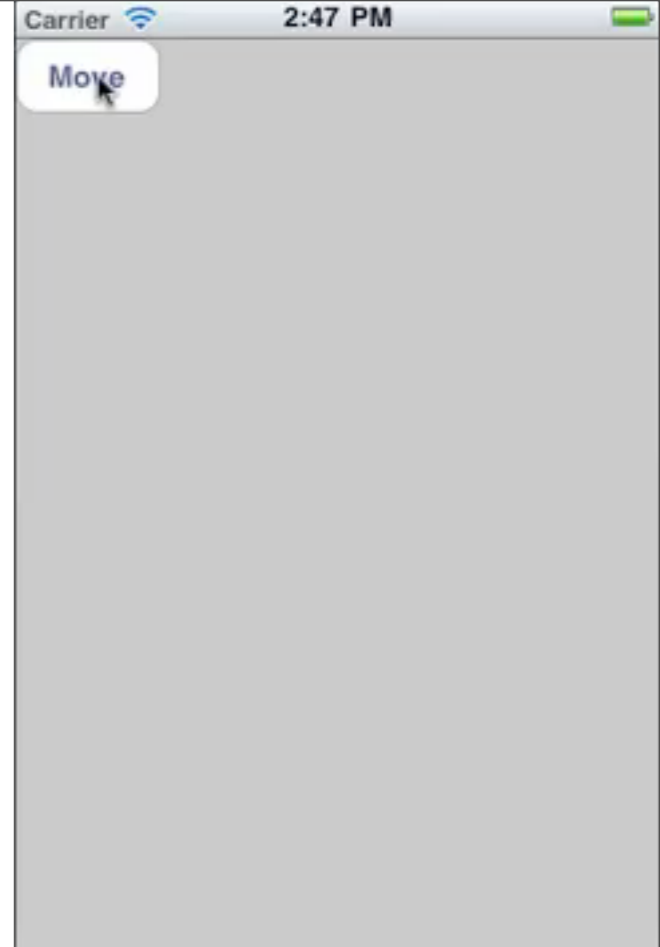
Pulse

```
- (void) move {
    CALayer * moveLayer = moveButton.layer;
    CAKeyframeAnimation * pulse =
        [CAKeyframeAnimation animationWithKeyPath:@"bounds"];
    pulse.duration = 1.0;
    pulse.repeatDuration = 5.0;
    NSMutableArray * sizes = [NSMutableArray array];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 75, 40)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 80, 45)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 72, 37)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 65, 30)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 72, 37)]];
    pulse.values = sizes;
    [moveLayer addAnimation: pulse forKey:@"pulse"];
}
```



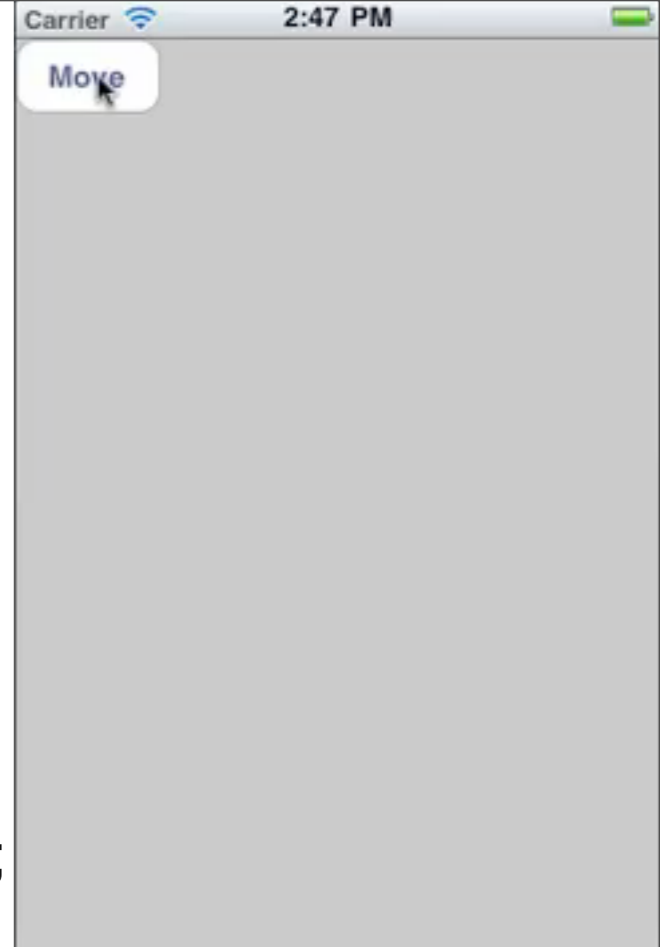
Move & Pulse

```
- (void) move {  
    CAAAnimation * move = [self moveDown];  
    CAAAnimation * pulse = [self pulseAnimation];  
    CAAAnimationGroup * group = [CAAAnimationGroup animation];  
    [group setAnimations:[NSArray arrayWithObjects:move, pulse, nil]];  
    group.repeatDuration = 6.0;  
    group.duration = 2.0;  
    [moveLayer addAnimation: group forKey:@"both"];  
}
```



Move & Pulse

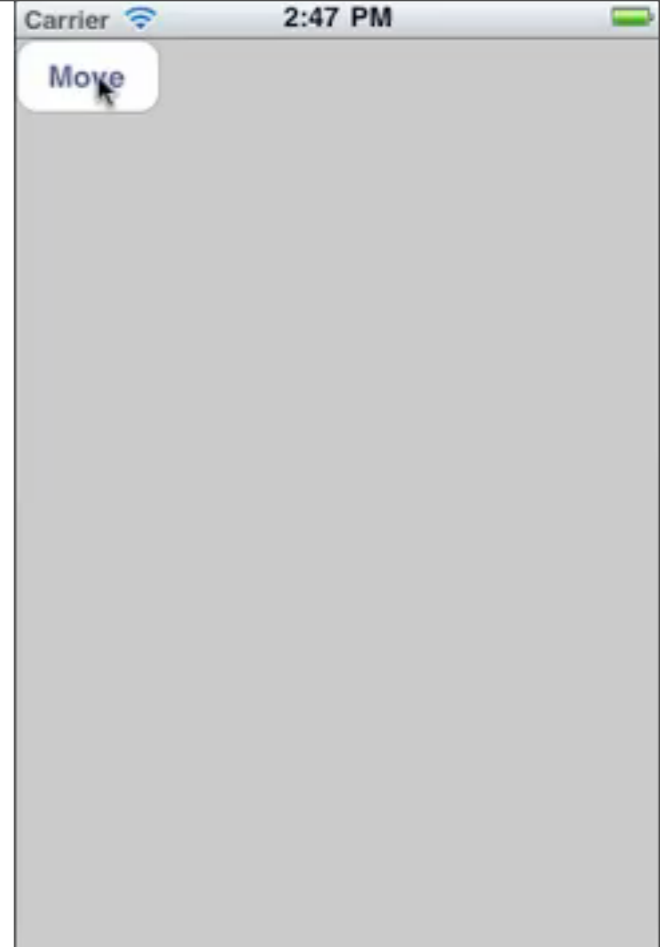
```
- (CAAnimation *) moveDown {
    CABasicAnimation * mover =
        [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.0;
    mover.fromValue = [NSValue valueWithCGPoint: CGPointMake(36, 18)];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(285, 440)];
    mover.delegate = self;
    mover.autoreverses = YES;
    CAMediaTimingFunction *easeInOut =
        [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    return mover;
}
```



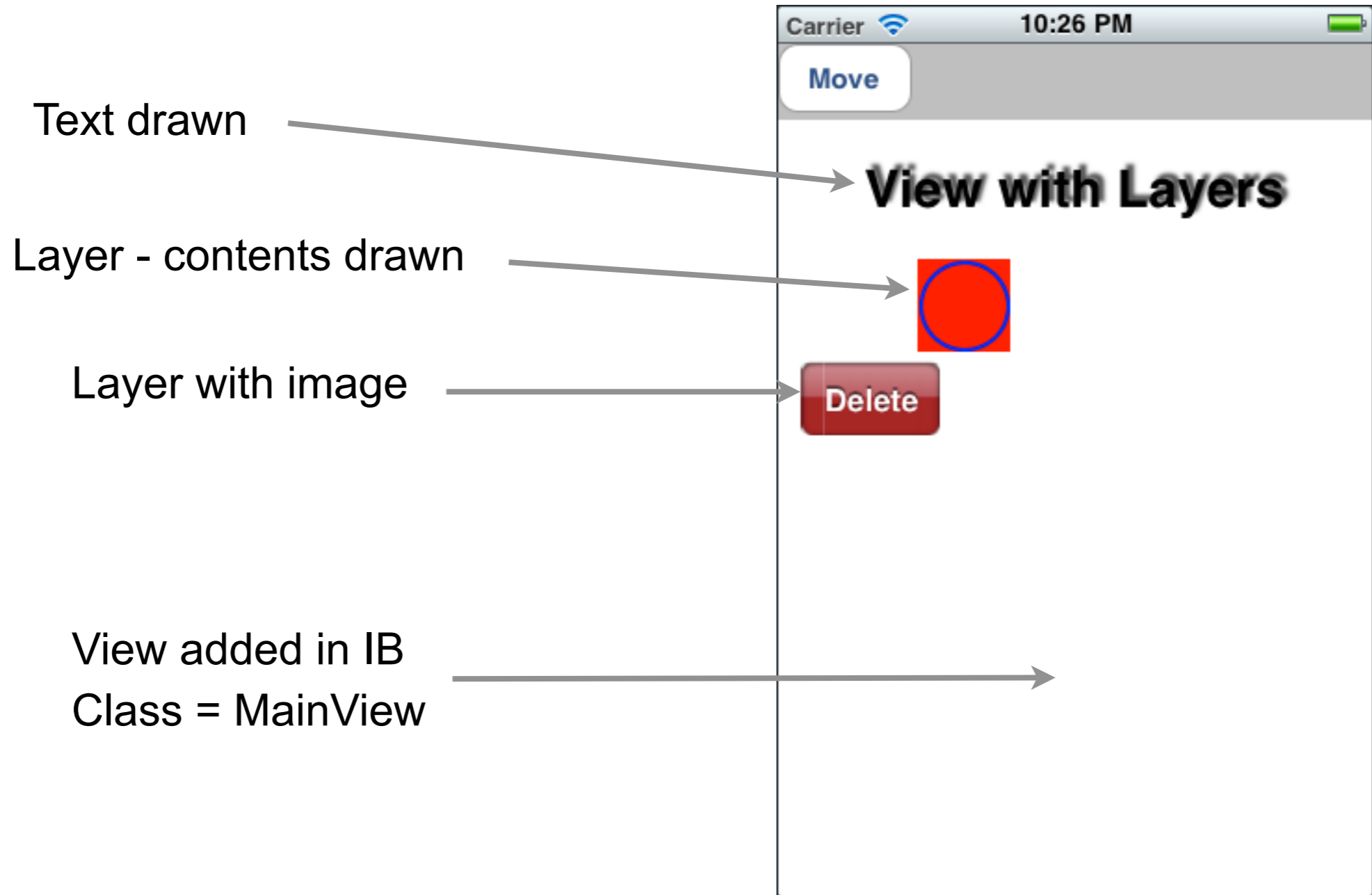
Move & Pulse

```
- (CAAnimation *) pulseAnimation {
    CAKeyframeAnimation * pulse =
    [CAKeyframeAnimation animationWithKeyPath:@"bounds"];
    pulse.duration = 1.0;
    pulse.repeatDuration = 6.0;
```

```
    NSMutableArray * sizes = [NSMutableArray array];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 75, 40)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 80, 45)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 72, 37)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 65, 30)]];
    [sizes addObject: [NSValue valueWithCGRect:CGRectMake(0, 0, 72, 37)]];
    pulse.values = sizes;
    return pulse;
}
```



CALayer example



CALayer example

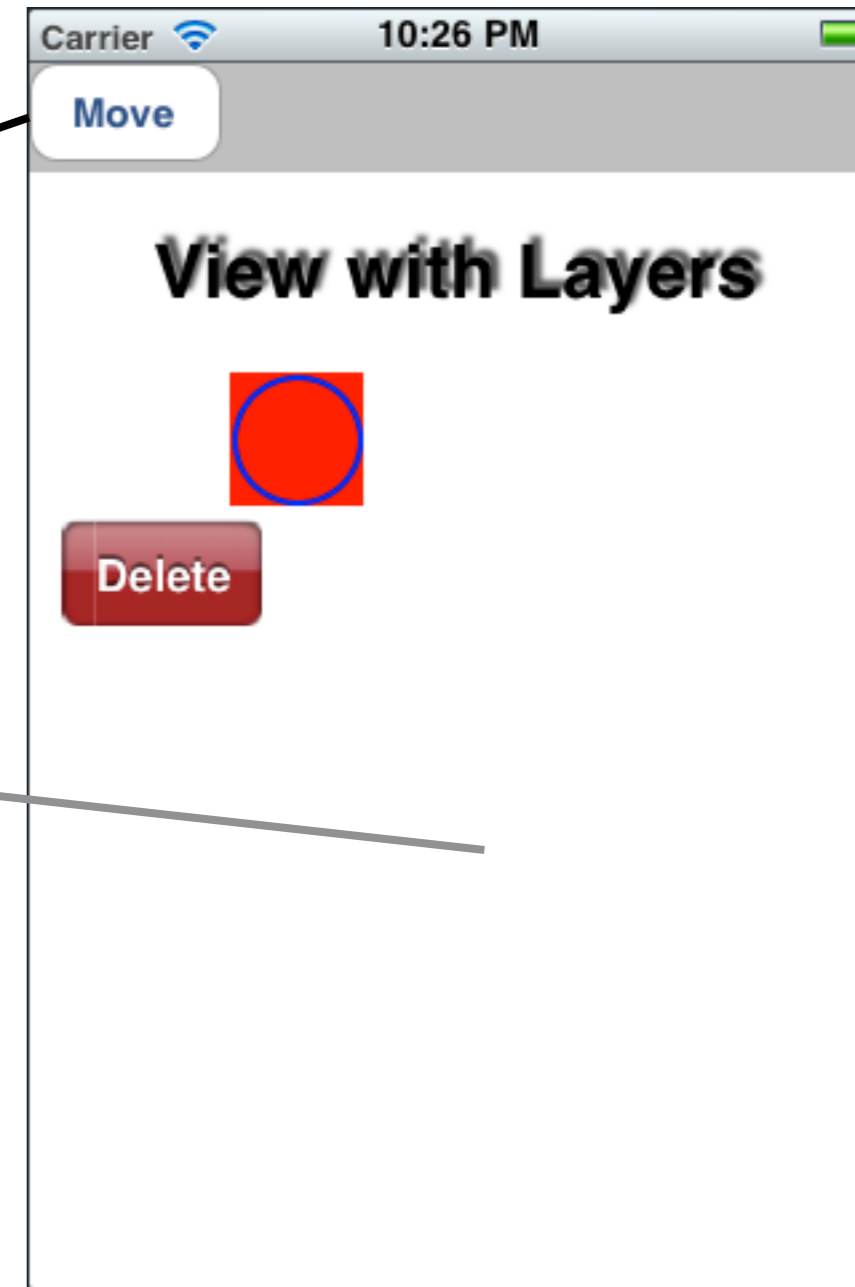
```
#import <UIKit/UIKit.h>  
#import "MainView.h"
```

```
@interface AnimationLayersViewController : UIViewController {  
}
```

```
-(IBAction) move;
```

```
@property (nonatomic, retain) IBOutlet UIView * sampleView;
```

```
@end
```



CALayer example

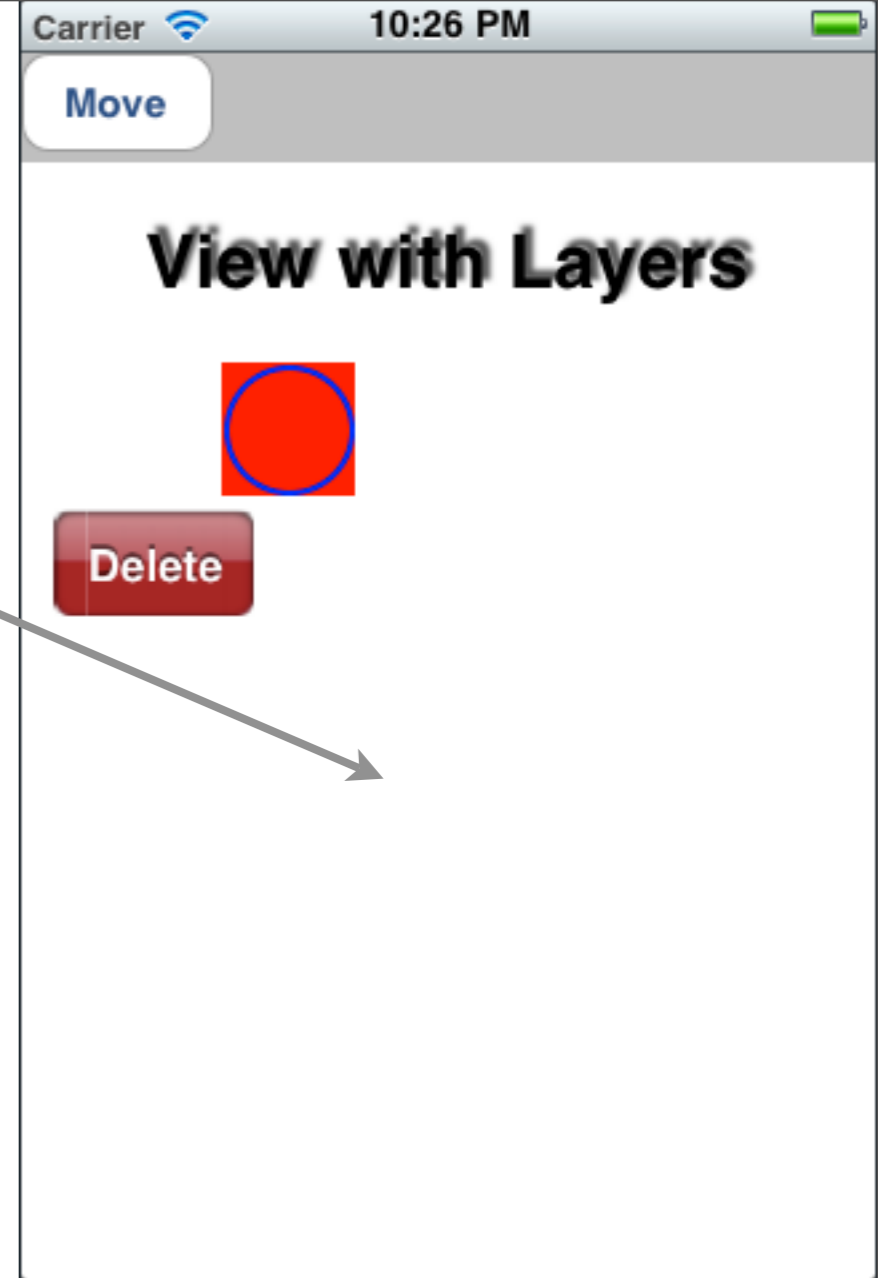
View added in IB
Class = MainView

```
#import <Foundation/Foundation.h>
#import <QuartzCore/QuartzCore.h>
#import "DrawLayer.h"

@interface MainView : UIView {
    CALayer *imageLayer;
    DrawLayer *drawLayer;
}

- (void) animateDelete;

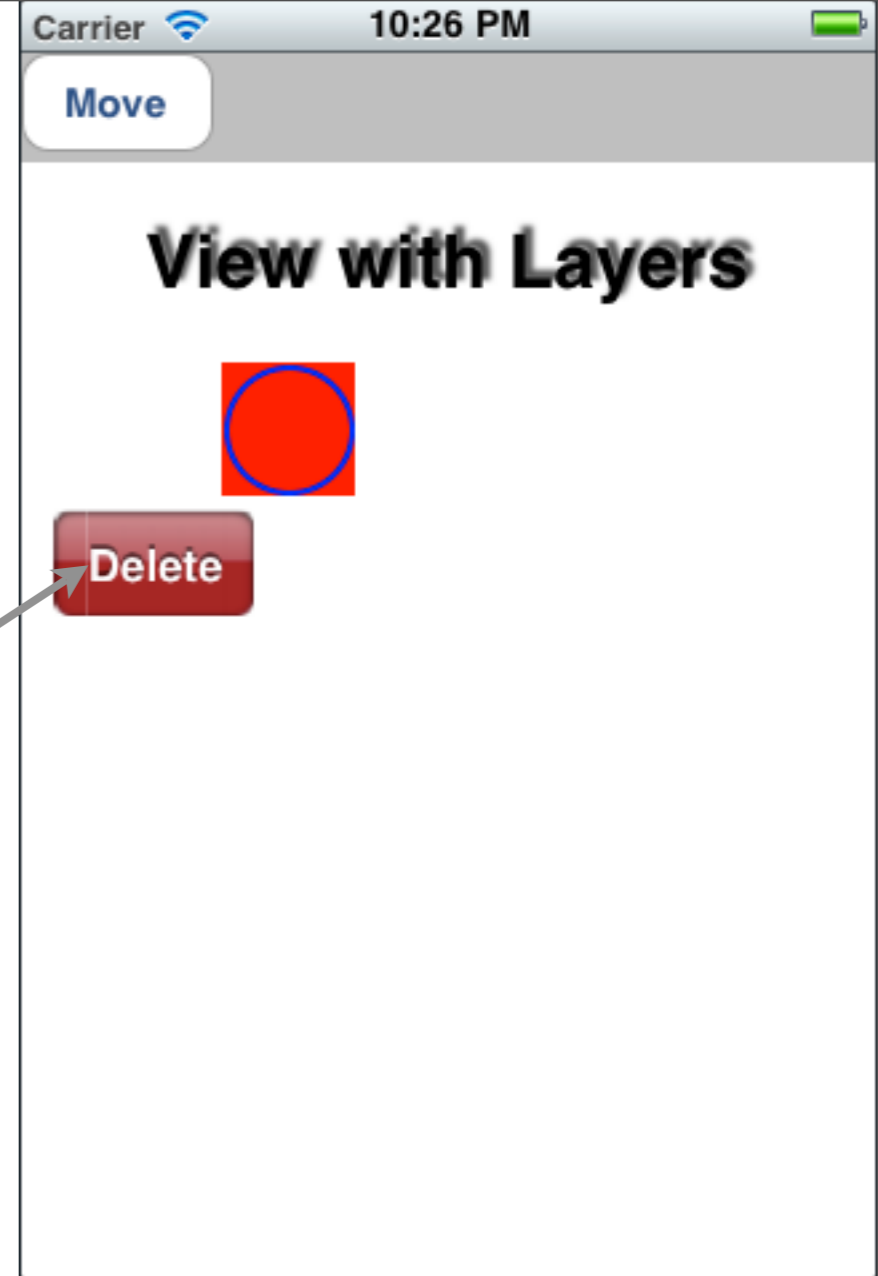
@end
```



CALayer example

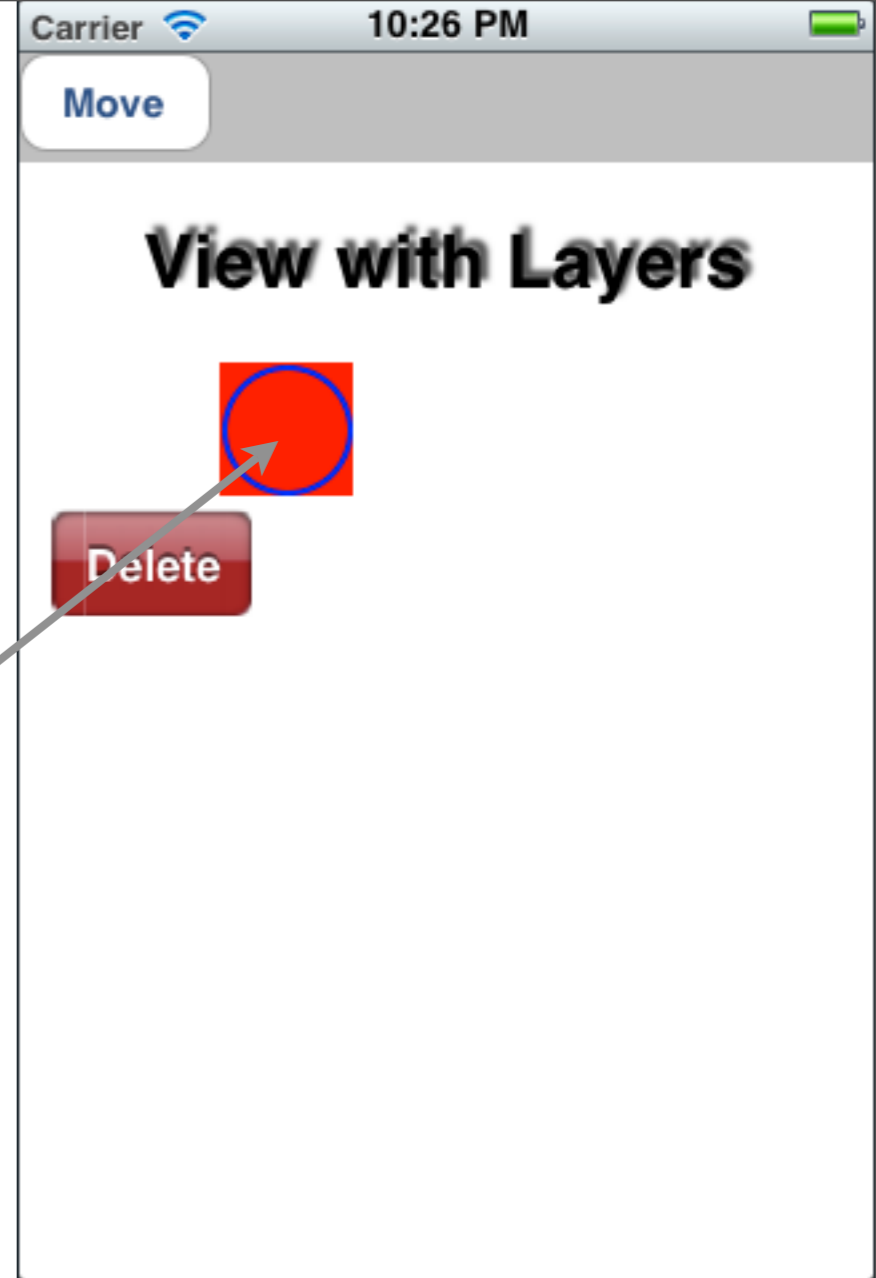
```
-(void)awakeFromNib {
    [self addImageLayer];
    [self addDrawlayer];
}

- (void) addImageLayer {
    UIImage * deleteUIImage =
        [UIImage imageNamed:@"delete.png"];
    imageLayer = [[CALayer alloc] init];
    imageLayer.bounds = CGRectMake(0, 0, 75, 40);
    imageLayer.position = CGPointMake(50, 150);
    [imageLayer setContents:(id) [deleteUIImage CGImage]];
    [imageLayer setContentsGravity:kCAGravityResizeAspect];
    [[self layer] addSublayer:imageLayer];
    [imageLayer release];
}
```



CALayer example

```
-(void)awakeFromNib {  
    [self addImageLayer];  
    [self addDrawlayer];  
}  
  
- (void) addDrawlayer {  
    drawLayer = [[DrawLayer alloc] init];  
    [drawLayer setBounds: CGRectMake(0, 0, 50, 50)];  
    [drawLayer setPosition:CGPointMake(100, 100)];  
    [drawLayer setBackgroundColor:[[UIColor redColor] CGColor]];  
    [[self layer] addSublayer:drawLayer];  
    [drawLayer release];  
}
```



CALayer example

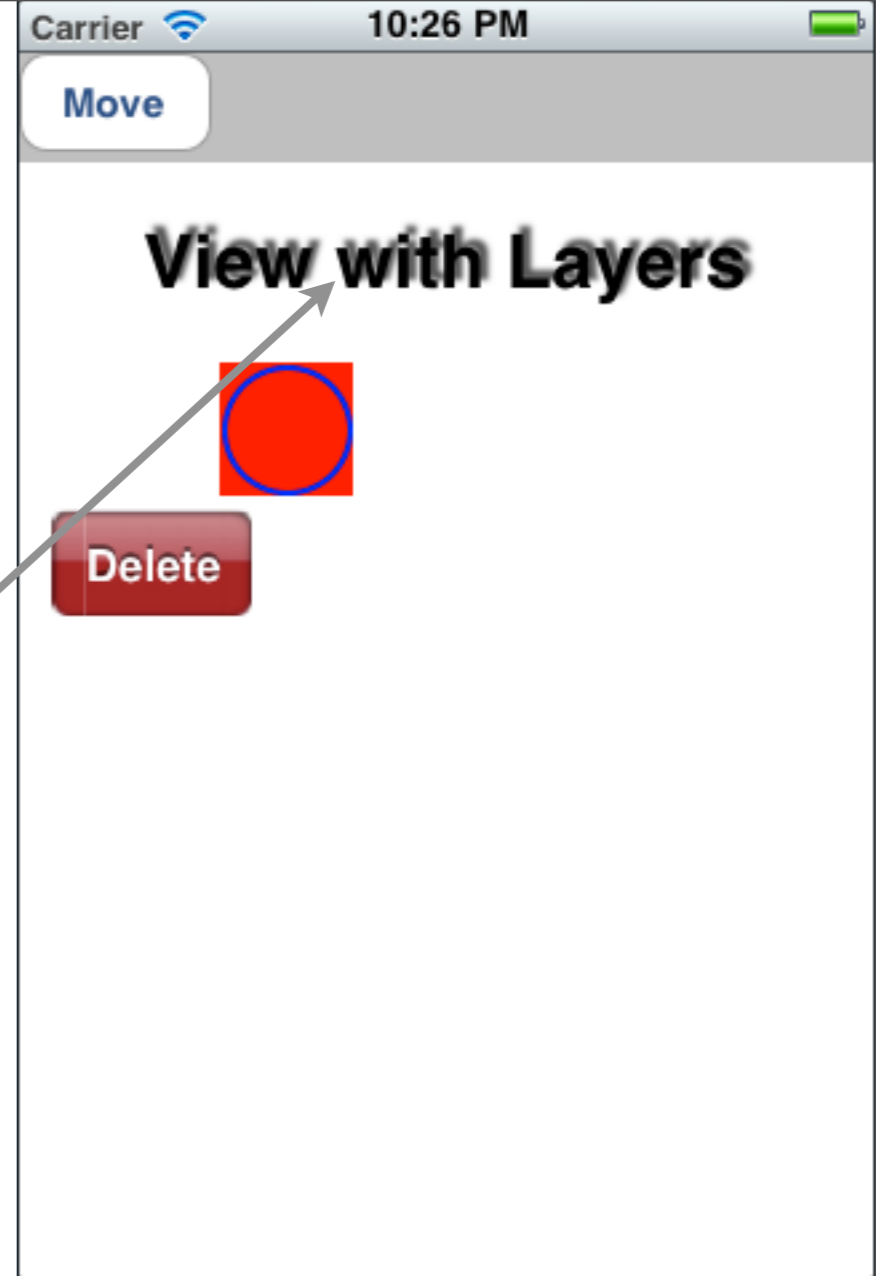
```
- (void)drawRect:(CGRect)rect {
    CGContextRef context = UIGraphicsGetCurrentContext();
    CGRect bounds = [self bounds];

    CGPoint center;
    center.x = bounds.origin.x + bounds.size.width / 2.0;
    center.y = bounds.origin.y + bounds.size.height / 2.0;

    NSString *text = @"View with Layers";
    UIFont *font = [UIFont boldSystemFontOfSize:28];
    CGRect textRect;
    textRect.size = [text sizeWithFont:font];
    textRect.origin.x = (bounds.size.width - textRect.size.width) / 2.0;
    textRect.origin.y = 20;

    [[UIColor blackColor] setFill];
    CGSize offset = CGSizeMake(4, -3);
    CGColorRef color = [[UIColor darkGrayColor] CGColor];
    CGContextSetShadowWithColor(context, offset, 2.0, color);

    [text drawInRect:textRect withFont:font];
    [drawLayer setNeedsDisplay];
}
```



[drawLayer setNeedsDisplay];

Must call setNeedsDisplay on CALayer that draws its contents

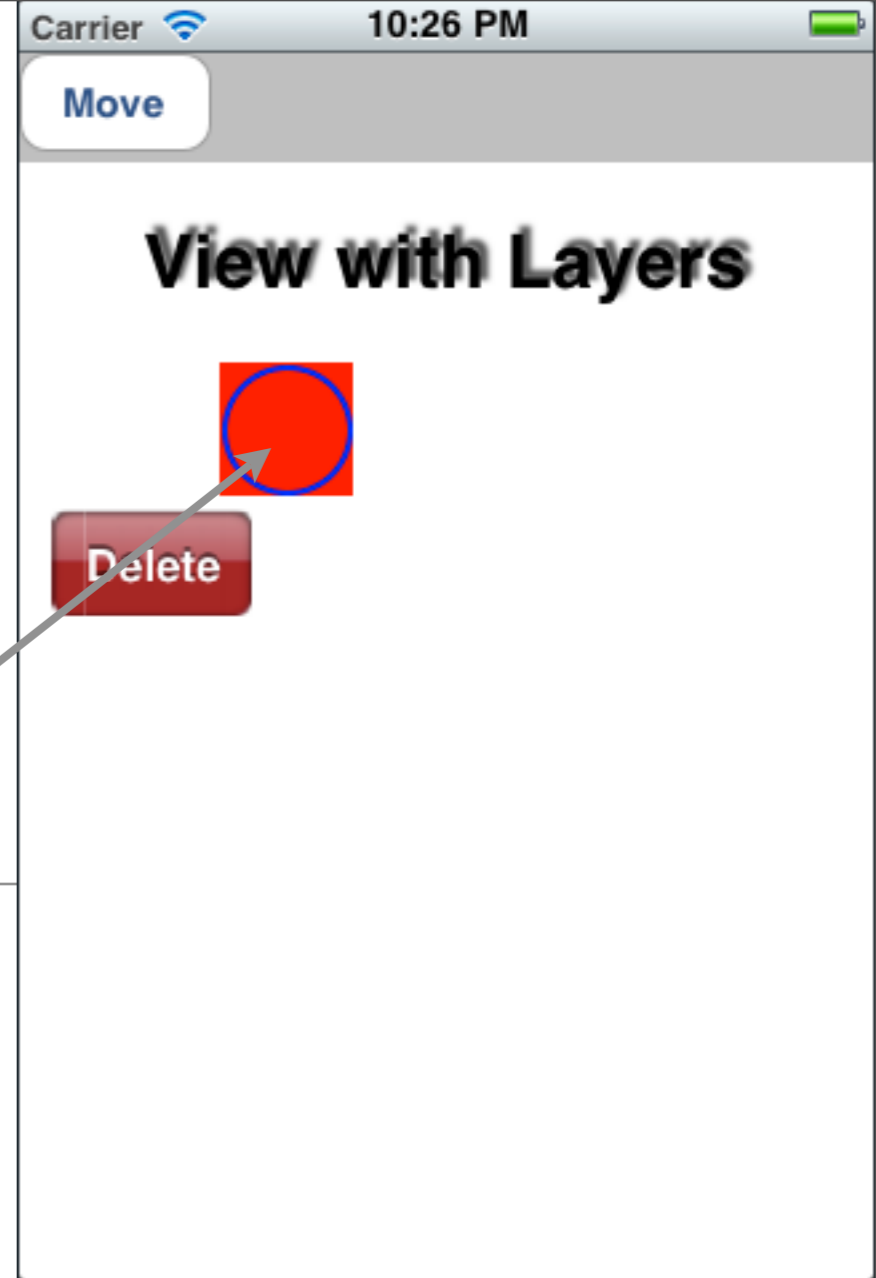
CALayer example

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

@interface DrawLayer : CALayer {
}
@end
```

```
#import "DrawLayer.h"

@implementation DrawLayer
- (void) drawInContext: (CGContextRef) context {
    CGContextSetStrokeColorWithColor(context,
        [UIColor colorWithRed: 0.0 green: 0 blue: 1.0 alpha:1].CGColor);
    CGContextSetLineWidth(context, 2);
    CGRect circleBoundry = CGRectMake(2,2,47,47);
    CGContextAddEllipseInRect(context, circleBoundry);
    CGContextDrawPath(context, kCGPathStroke);
}
@end
```



CALayer - Subclassing or Delegate

Subclassing

To draw implement

- (void) drawInContext: (CGContextRef) context

Delegate

To draw delegate implements

- (void drawLayer: (CALayer *) inContext: (CGContextRef) context

CALayer, Delegates & Views

Each View has a CALayer used to draw its contents

The View is the delegate for the CALayer

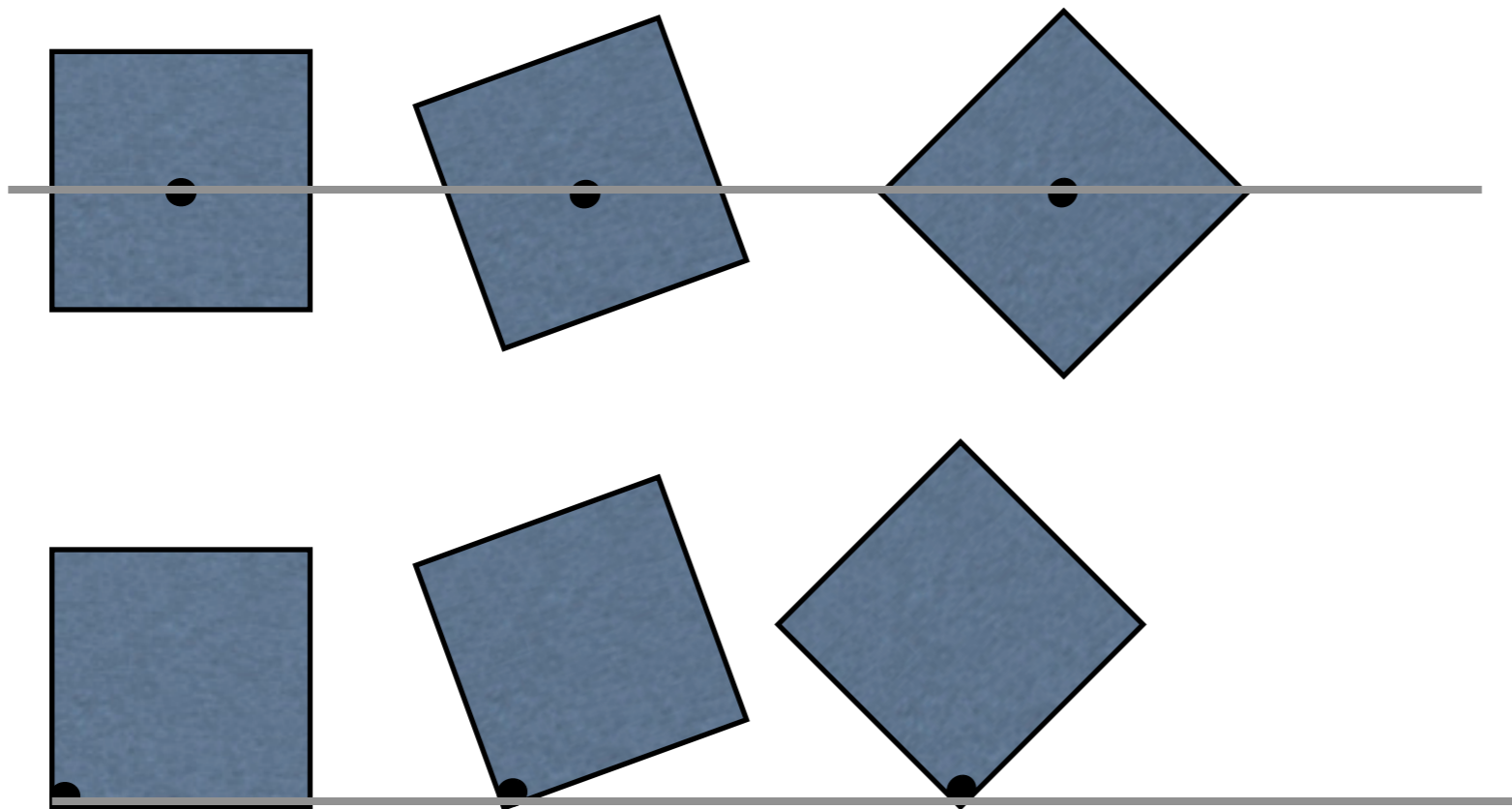
View's drawRect: is called by the CALayer delegate method

SO

If a View has sublayers don't make the View the delegate of other layers

CALayer - anchorPoint

Point where CALayer rotates around
Can be changed



CALayer - anchorPoint, position, bounds

position of CALayer

location of anchorPoint in containers coordinates

anchorPoint

Default is center of CALayer

bounds

width & height

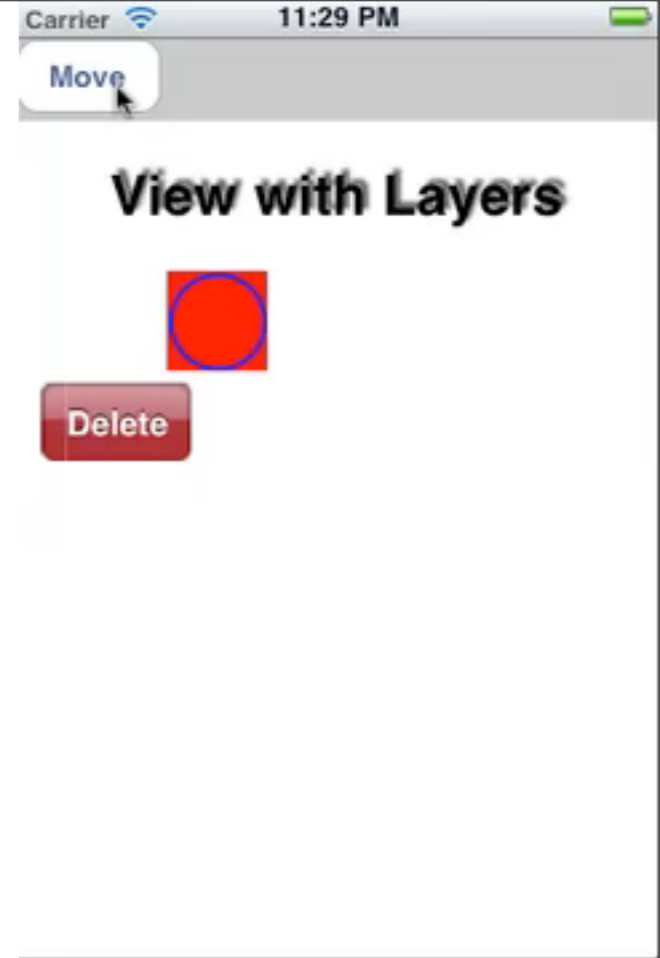
frame

Computed

Don't try to animate frame

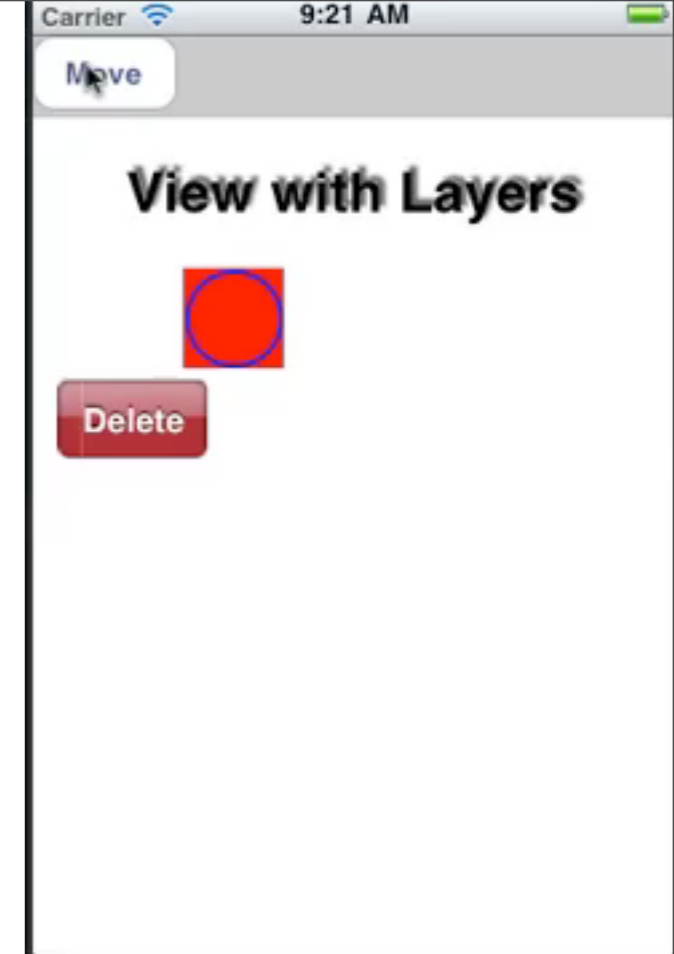
Implicit Animation

```
- (void) animateDelete {  
    imageLayer.position = CGPointMake(200, 400);  
}
```



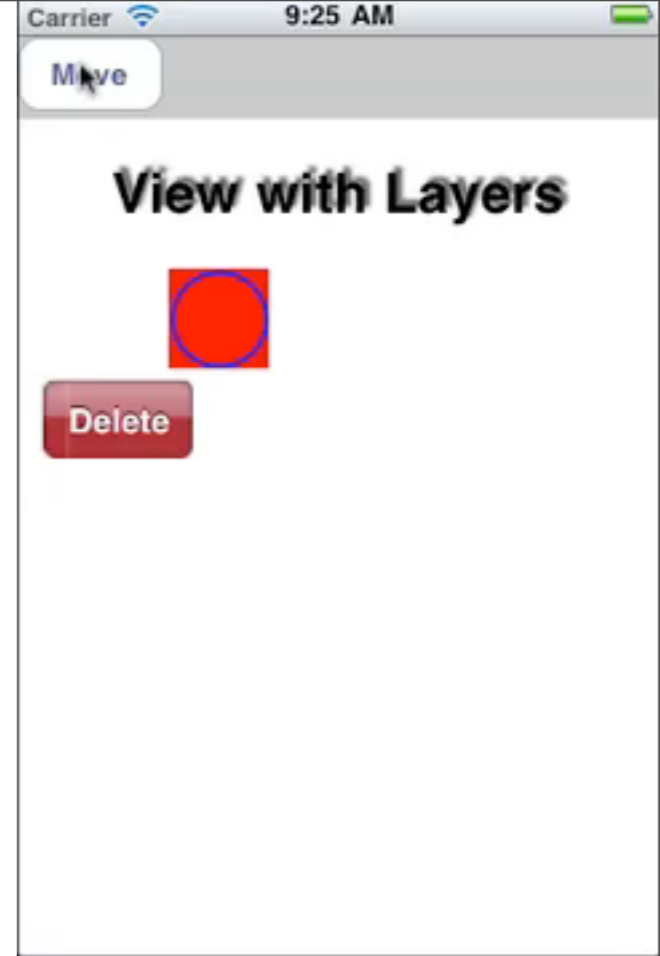
Explicit Animation

```
- (void) animateDelete {
    NSLog(@"animateDelete");
    CABasicAnimation * mover =
    [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.0;
    mover.fromValue = [NSValue valueWithCGPoint: CGPointMake(50, 150)];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(200, 400)];
    CAMediaTimingFunction *easeInOut =
    [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    mover.delegate = self;
    imageLayer.position = CGPointMake(200, 400);
    [imageLayer addAnimation:mover forKey:@"moving"];
}
```



Implicit Animation too

```
- (void) animateDelete {
    NSLog(@"animateDelete");
    CABasicAnimation * mover =
        [CABasicAnimation animationWithKeyPath:@"position"];
    mover.duration = 1.0;
    mover.fromValue = [NSValue valueWithCGPoint: CGPointMake(50, 150)];
    mover.toValue = [NSValue valueWithCGPoint: CGPointMake(200, 400)];
    CAMediaTimingFunction *easeInOut =
        [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];
    mover.timingFunction = easeInOut;
    mover.delegate = self;
    [imageLayer addAnimation:mover forKey:@"moving"];
    imageLayer.position = CGPointMake(200, 400);
}
```



Thread Example



AnimationLayersViewController

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

```
@interface AnimationLayersViewController : UIViewController {
```

```
}
```

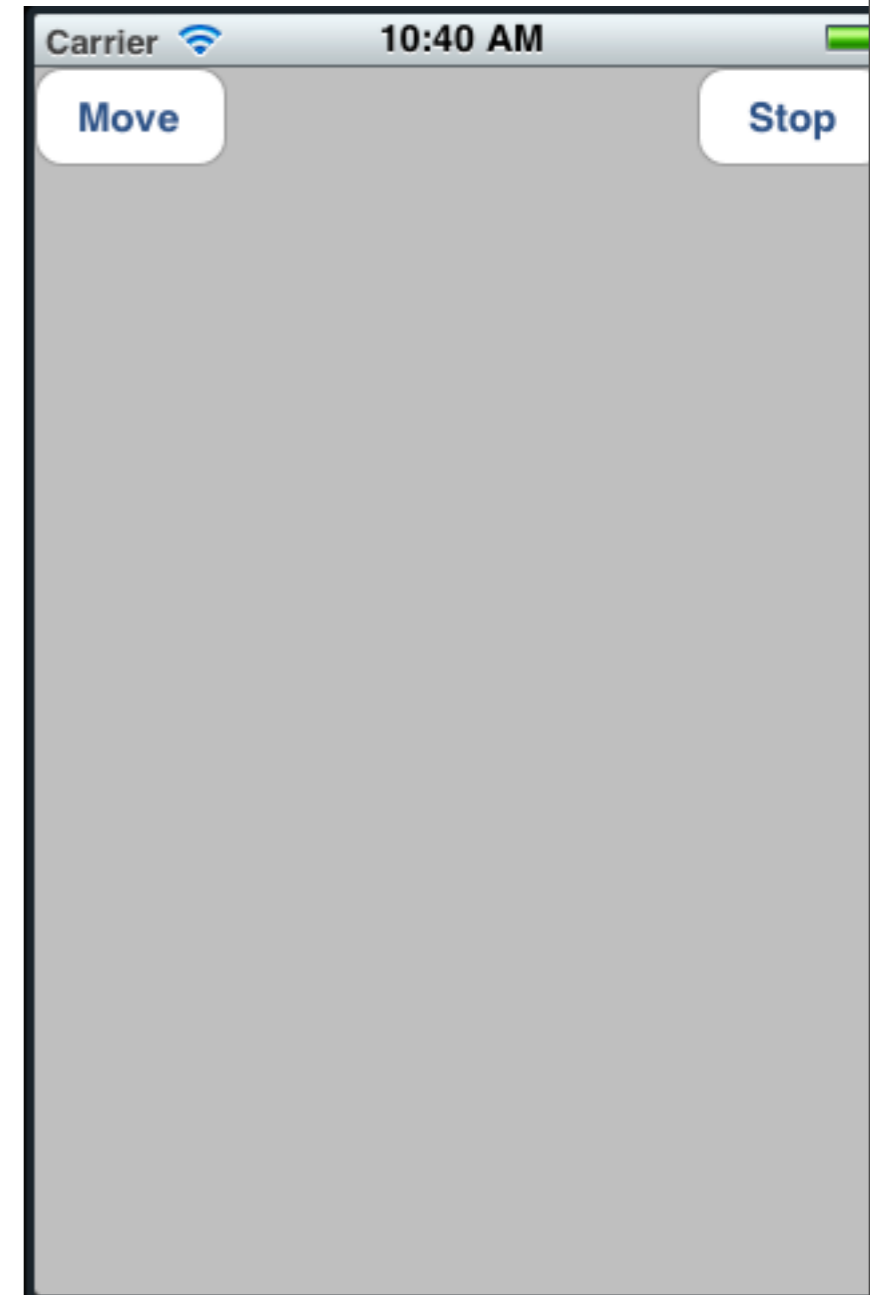
```
@property (nonatomic, retain) NSThread *moveThread;
```

```
@property (nonatomic, retain) IBOutlet UIButton * moveButton;
```

```
-(IBAction) move;
```

```
-(IBAction) stop;
```

```
@end
```



Private Methods/fields

```
@interface AnimationLayersViewController (PrivateMethods)
```

```
    CGFloat deltaX = 1.0;
```

```
    CGFloat deltaY = 1.0;
```

```
- (void)waitForThreadToFinish;
```

```
- (void)startThread;
```

```
@end
```

```
@implementation AnimationLayersViewController
```

```
@synthesize sampleView;
```

```
@synthesize moveButton;
```

```
@synthesize moveThread;
```

Starting Thread

```
- (void) move {
    [self startThread];
}

- (void)startThread {
    if (moveThread != nil) {
        [moveThread cancel];
        [self waitForThreadToFinish];
    }
    NSThread *driverThread = [[NSThread alloc] initWithTarget:self
        selector:@selector(animateButton) object:nil];
    self.moveThread = driverThread;
    [driverThread release];
    [self.moveThread start];
}
```

Run By Thread

```
-(void) animateButton {
    BOOL continueRunning = YES;
    while (continueRunning) {
        if ([moveThread isCancelled] == YES) {
            continueRunning = NO;
            return;
        }
        [self performSelectorOnMainThread:@selector(moveButtonOneStep)
         withObject:nil waitUntilDone:NO];
        [NSThread sleepForTimeInterval:0.01];
    }
}
```

Moving the Button

```
-(void) moveButtonOneStep {
    CALayer * buttonLayer = [moveButton layer];
    if (buttonLayer.position.x >285) {
        deltaX = -1.0;
    }
    if (buttonLayer.position.x <35) {
        deltaX = 1.0;
    }
    if (buttonLayer.position.y >440) {
        deltaY = -1.0;
    }
    if (buttonLayer.position.y <18) {
        deltaY = 1.0;
    }
    CGFloat newX = buttonLayer.position.x + deltaX;
    CGFloat newY = buttonLayer.position.y + deltaY;

    buttonLayer.position = CGPointMake(newX, newY);
}
```

Stopping the thread

```
- (void)stop {  
    [self.moveThread cancel];  
    [self waitForThreadToFinish];  
    self.moveThread = nil;  
}
```

```
- (void)waitForThreadToFinish {  
    while (moveThread && ![moveThread isFinished]) {  
        [NSThread sleepForTimeInterval:0.1];  
    }  
}
```