CS 696 Mobile Application Development
Fall Semester, 2010
Doc 6 iPhone App Intro
Sep 9, 2010

References

View Controller Programming Guide for iOS

iPhone Application Programming Guide

Various UIKit class's documentation

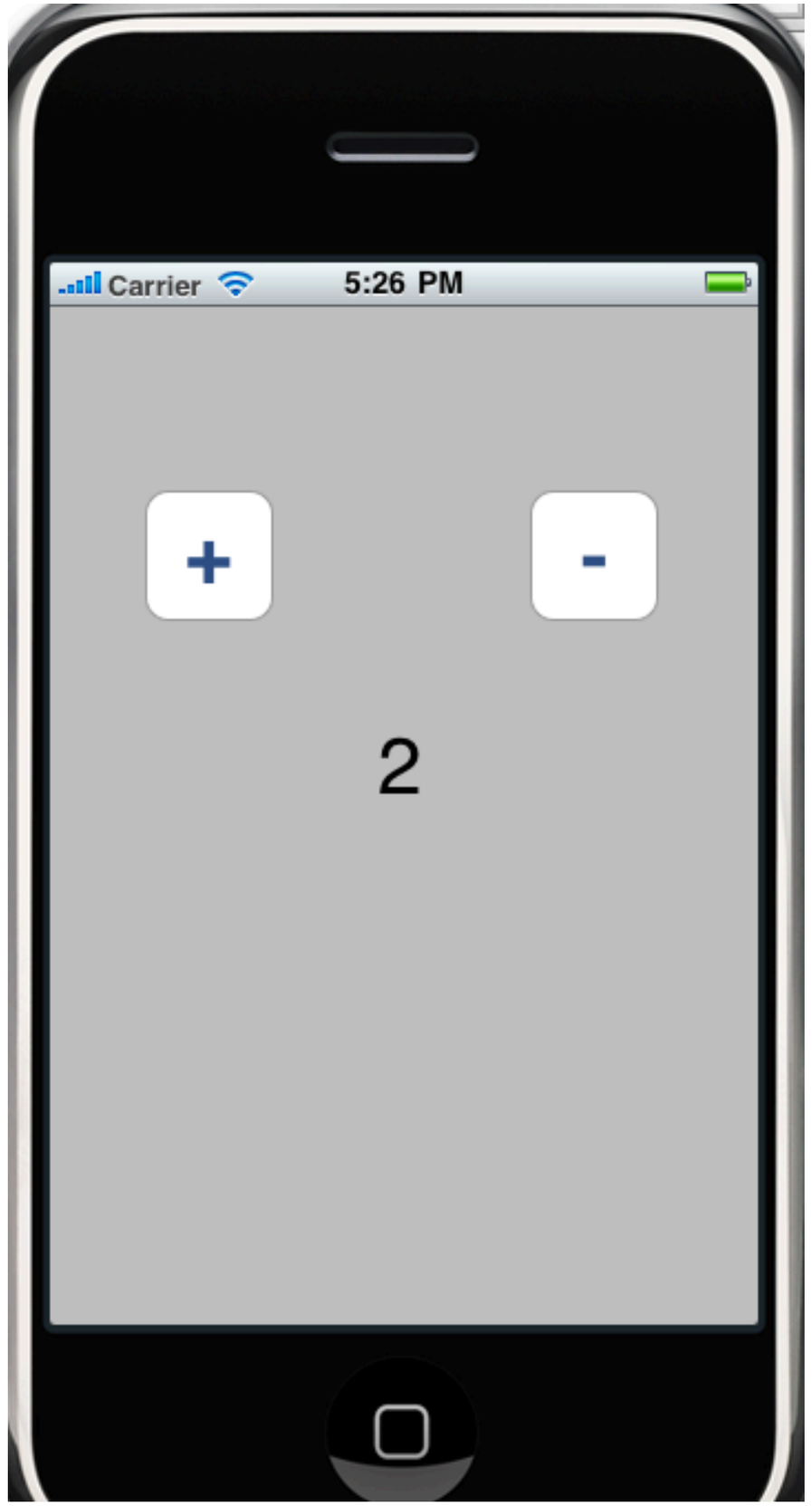
CS 193P Standford iPhone Application Development, Lecture 4, http://itunes.apple.com/us/podcast/lecture-4-slides-january-14/id384233225?i=85092607

First Example

Counter

+ increases count

- decrease count



Things to Master

Concepts

Application & Delegation

Model View Contoller

Events

Frameworks

UIKit
Classes, methods
How it works

Tools

Xcode
Interface builder
xib files
resources etc

MVC - Model View Controller

Controller

Intermediary between Model & View

When model changes updates view

When user interacts with view updates model

App logic lives here (sigh)

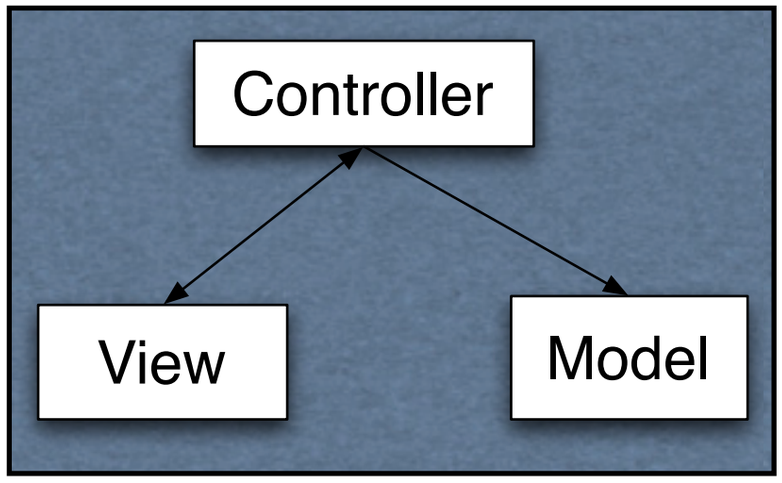
Model

App data & functionality

View

The user interface

App Code



UIApplication

All iOS apps are instances of UIApplication class

Handles lifecycle of the application

Handles events from OS

Manages status bar

Contains all applicaition windows

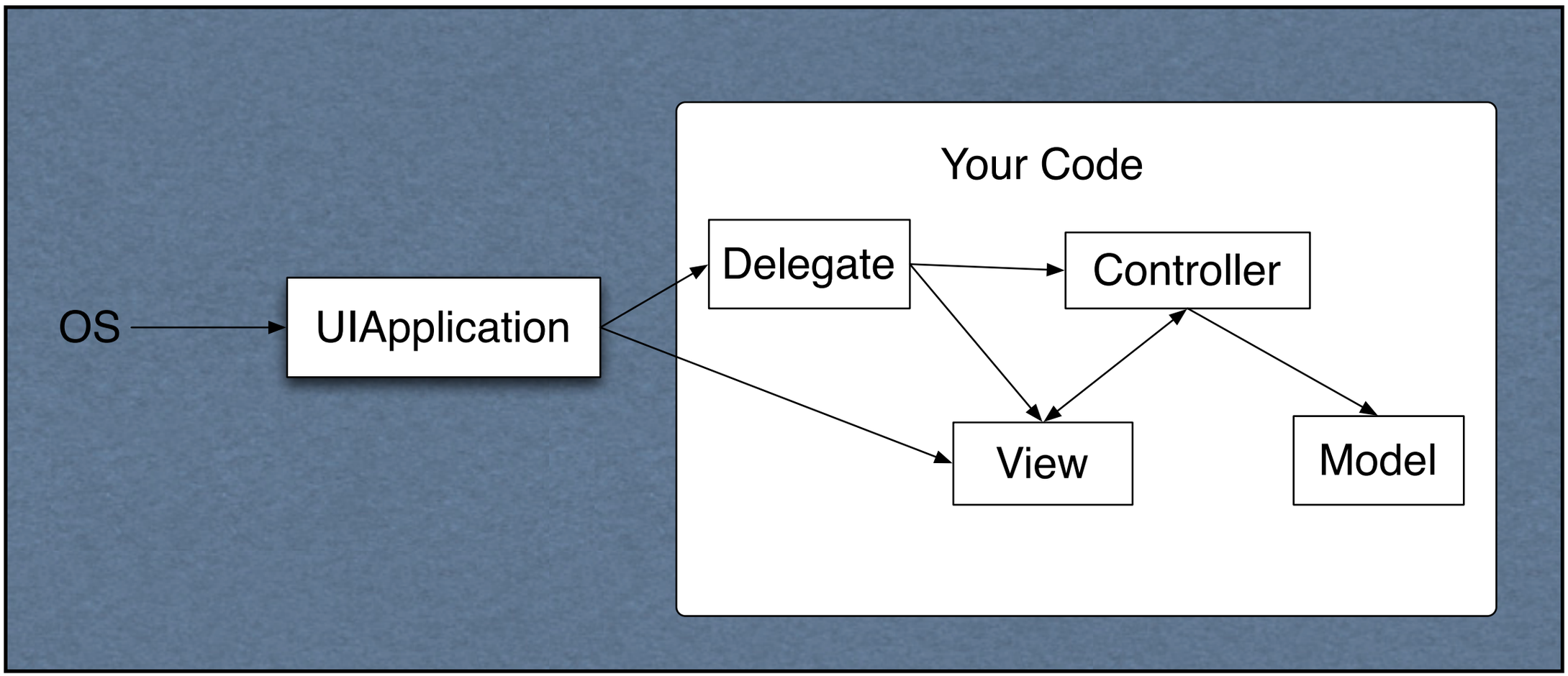
OS Events & Your App

Some OS events have to be handled by your code

App started
App moving to background
App terminating
App received memory warning
App received notification

Delegate

entrust (a task or responsibility) to another person



Delegate Pattern

Common pattern in iOS framework

Framework class in charge of some task

Parts of task must be done by your code

Rather than subclass framework class

- Framework class uses a delegate class

Uses delegate to perform some of its tasks

We implement a delegate class

Counter Project

Click on +

Calls increasePressed method on controller

Controller has view update value

Click on -

Calls increasePressed method on controller

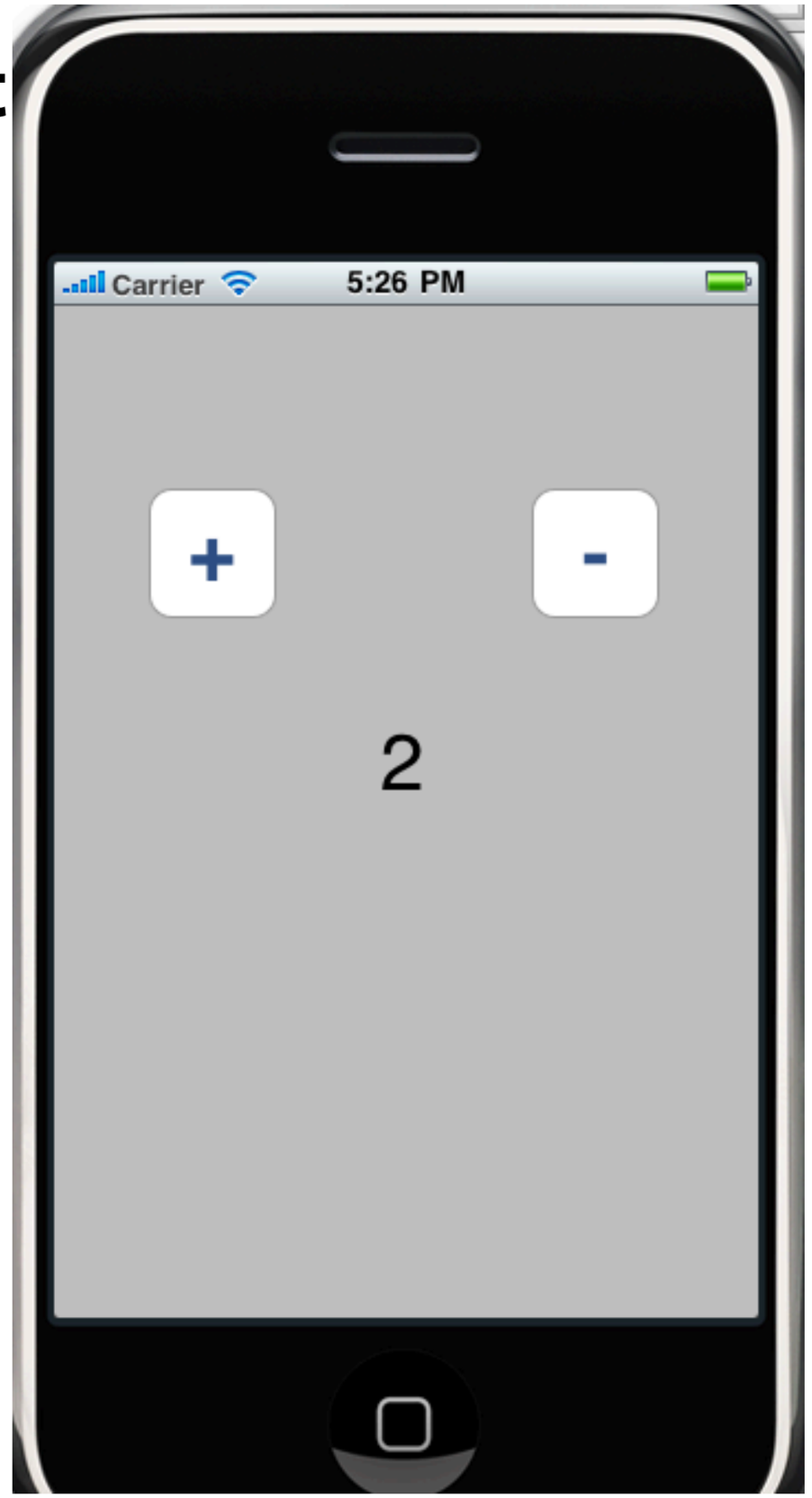
Controller has view update value

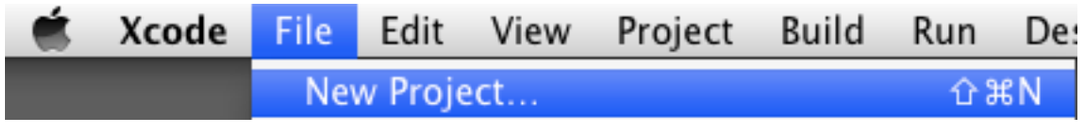
+ & -

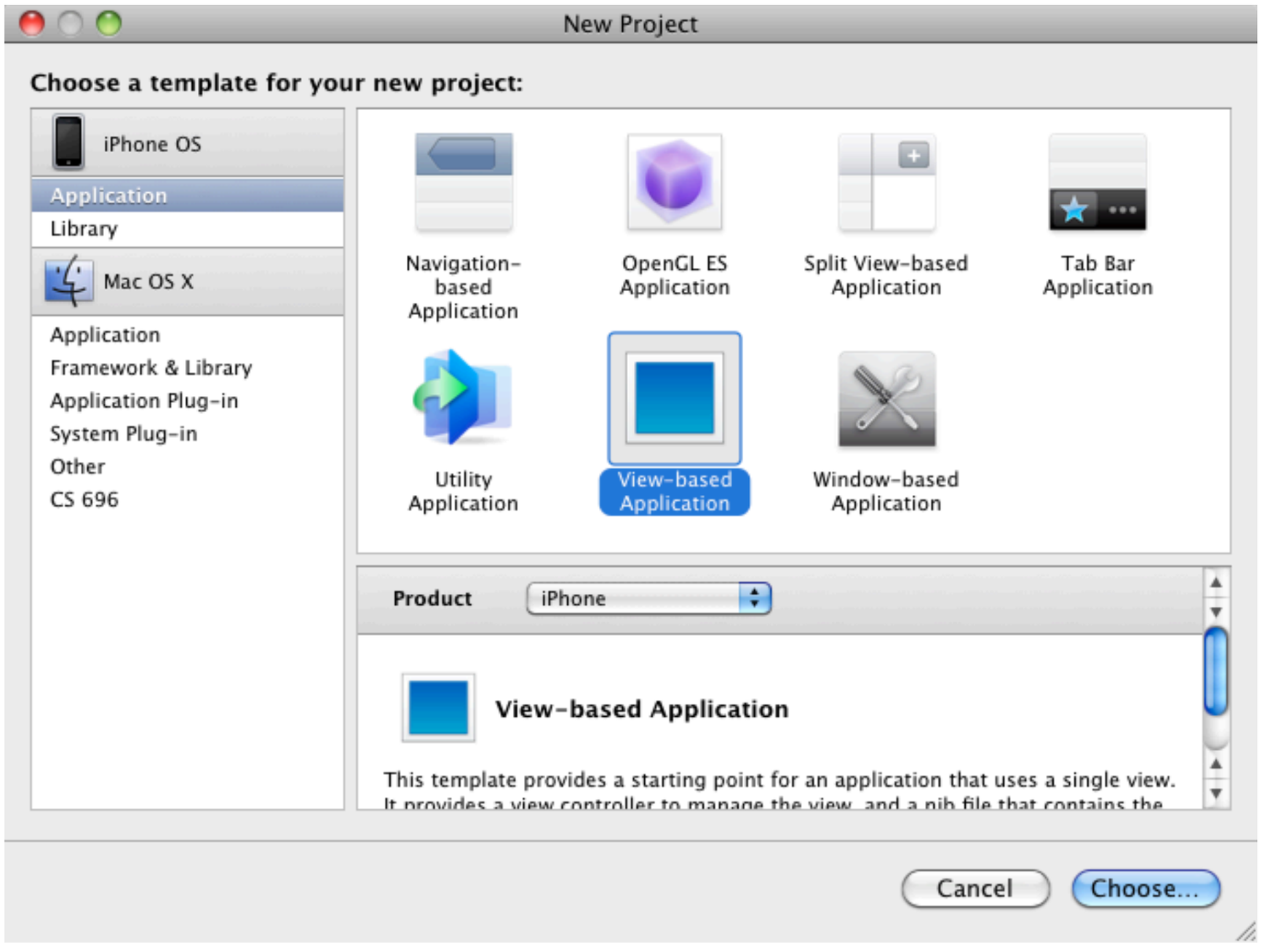
button

2

label that changes with count




Xcode
File
Edit
View
Project
Build
Run
New Project…
⇧⌘N


New Project
Choose a template for your new project:
iPhone OS
Application
Library
Mac OS X
Application
Framework & Library
Application Plug-in
System Plug-in
Other
CS 696
Navigation-based Application
OpenGL ES Application
Split View-based Application
Tab Bar Application
Utility Application
View-based Application
Window-based Application
Product
iPhone
View-based Application
This template provides a starting point for an application that uses a single view.
Cancel
Choose…

Files Created For Counter Project

Classes

- CounterAppDelegate.h
- CounterAppDelegate.m
- CounterViewController.h
- CounterViewController.m

Other Sources

- Counter_Prefix.pch
- main.m

Resources

- CounterViewController.xib
- MainWindow.xib
- Counter-Info.plist

main.m

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

int main(int argc, char *argv[]) {

    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

Counter_Prefix.pch

```
#ifdef __OBJC__
    #import <Foundation/Foundation.h>
    #import <UIKit/UIKit.h>
#endif
```

Counter-Info.plist

plist - property list

Key	Value
▼ Information Property List	(12 items)
Localization native development re	English
Bundle display name	${PRODUCT_NAME}
Executable file	${EXECUTABLE_NAME}
Icon file	
Bundle identifier	com.yourcompany.${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	6.0
Bundle name	${PRODUCT_NAME}
Bundle OS Type code	APPL
Bundle creator OS Type code	????
Bundle version	1.0
Application requires iPhone enviror	☑
Main nib file base name	MainWindow

CounterViewController.h

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

@interface CounterViewController : UIViewController {

}

@end
```


We will add to this class

CounterAppDelegate

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

@class CounterViewController;

@interface CounterAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    CoutnerViewController *viewController;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet CounterViewController *viewController;

@end
```

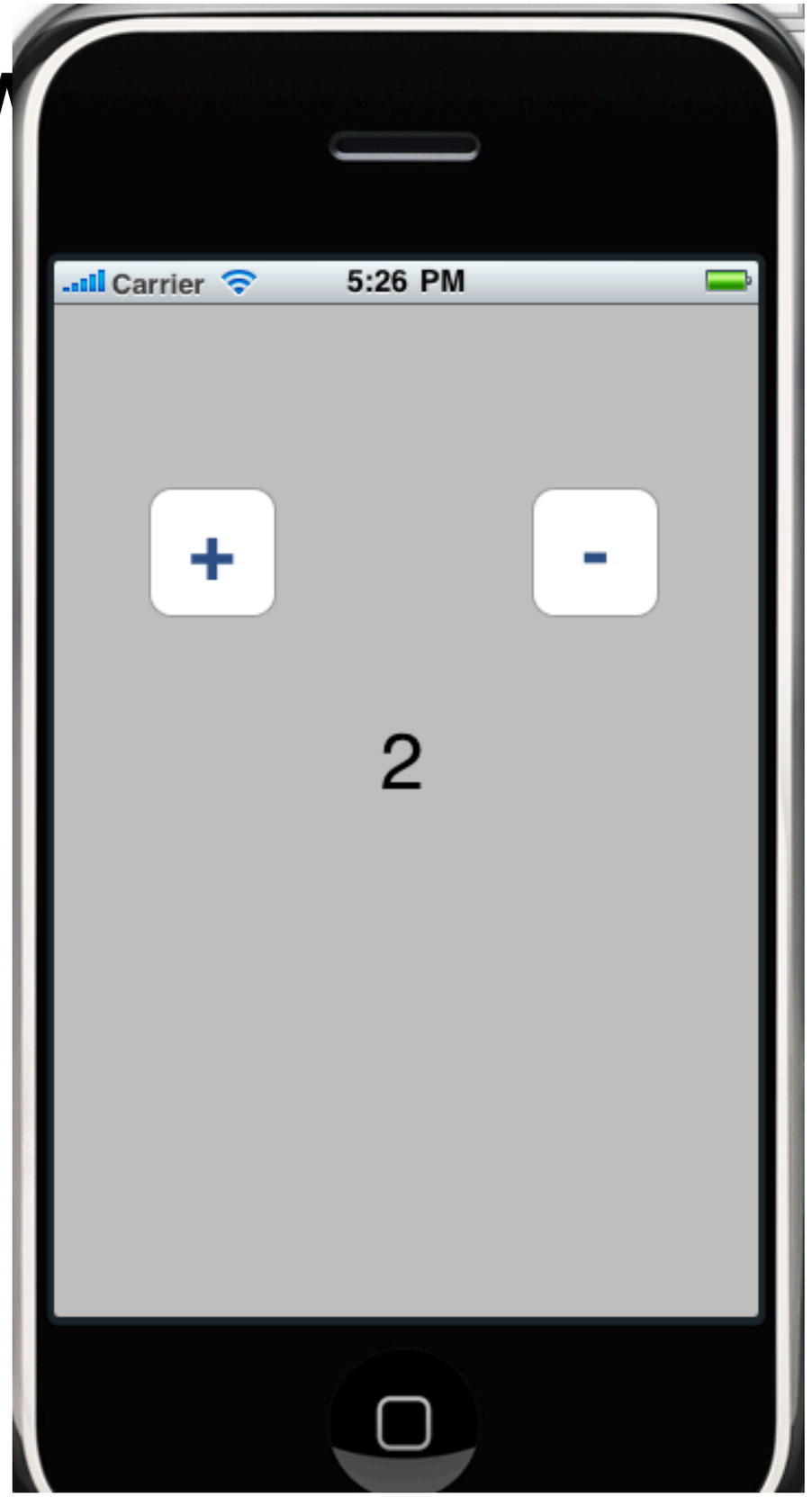

We will deal this class for a while

Creating the view

Use Interface Builder to draw view

or

Create view in code



Views with Interface Builder

Draw the view in Interface Builder

Save the view (xib file)

Declare connections in code

Make connection in Interface Builder

View

Normally defined in xib & nib files

xib

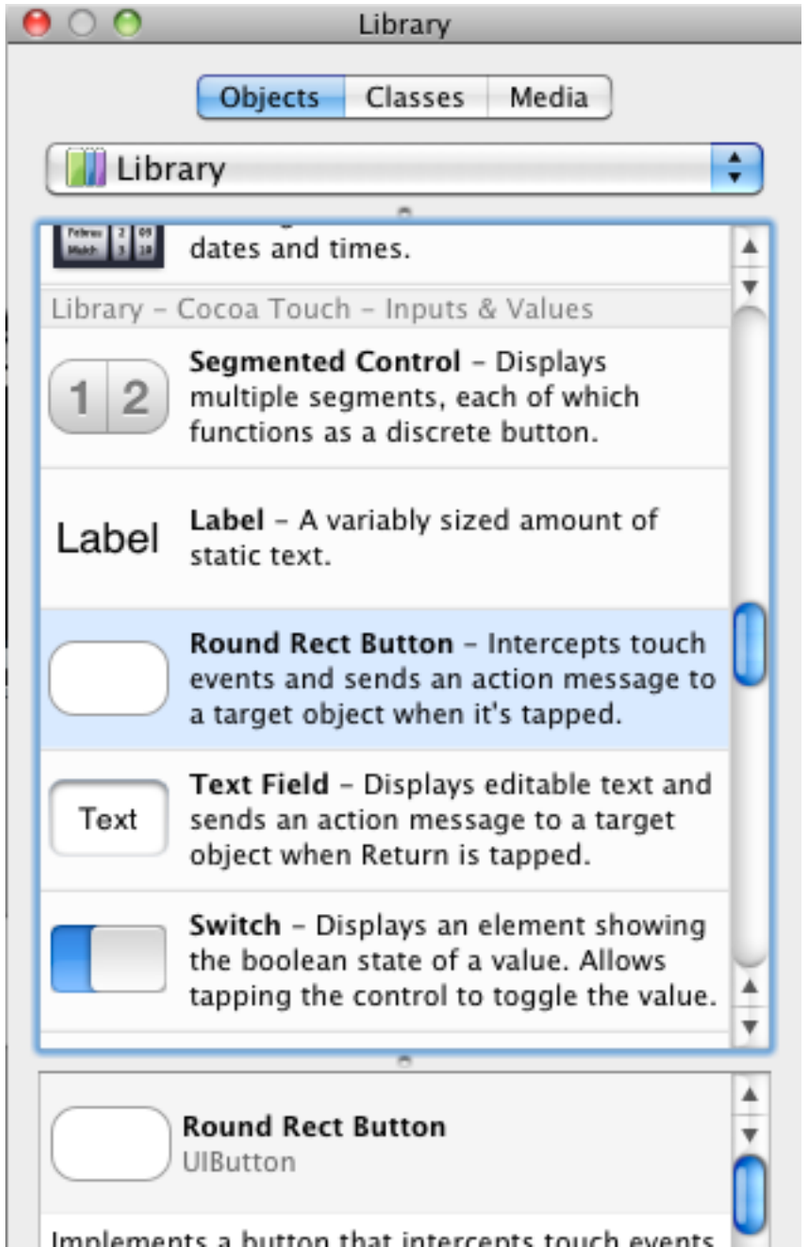
XML file containing UI layout for a screen

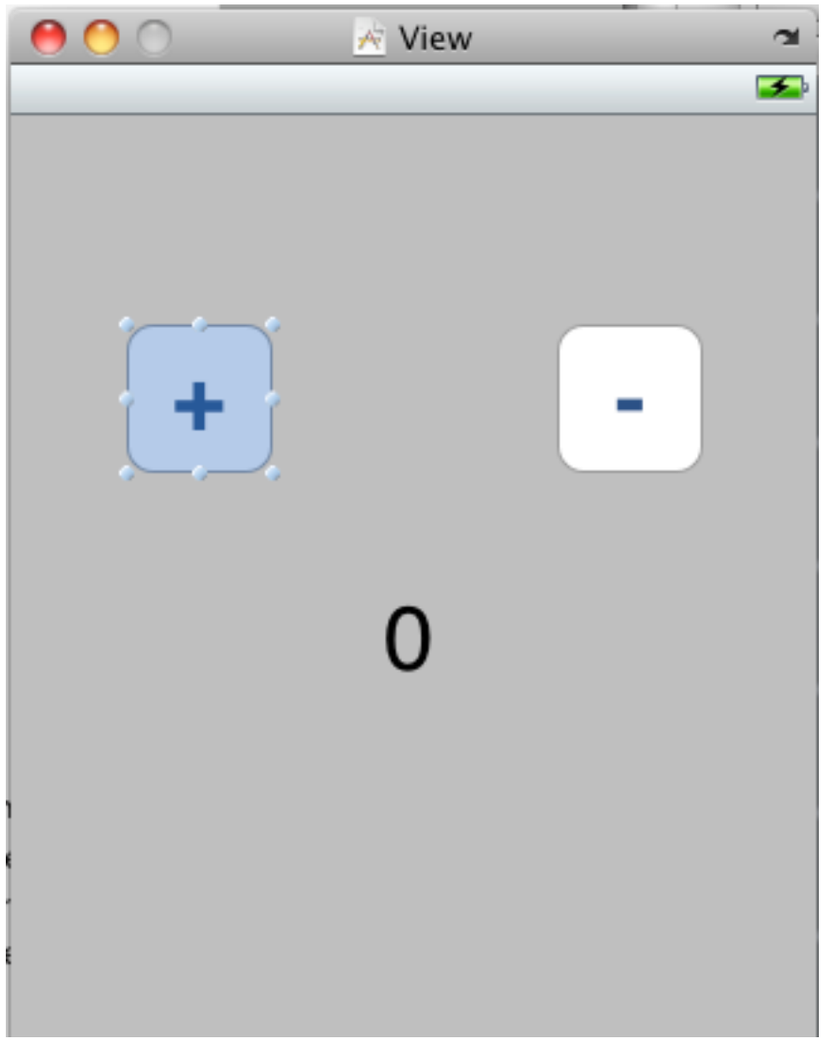
Generated by Interface Builder

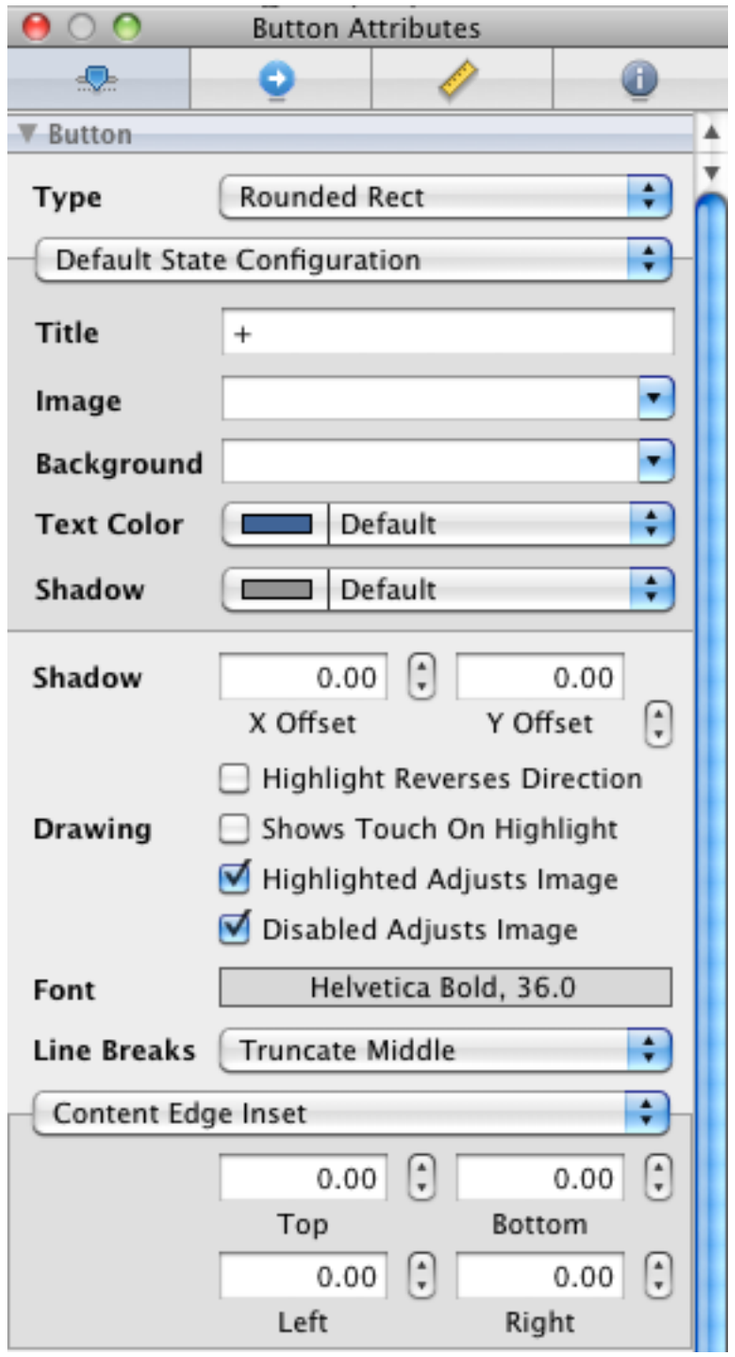
nib

binary version of xib used in compiled app

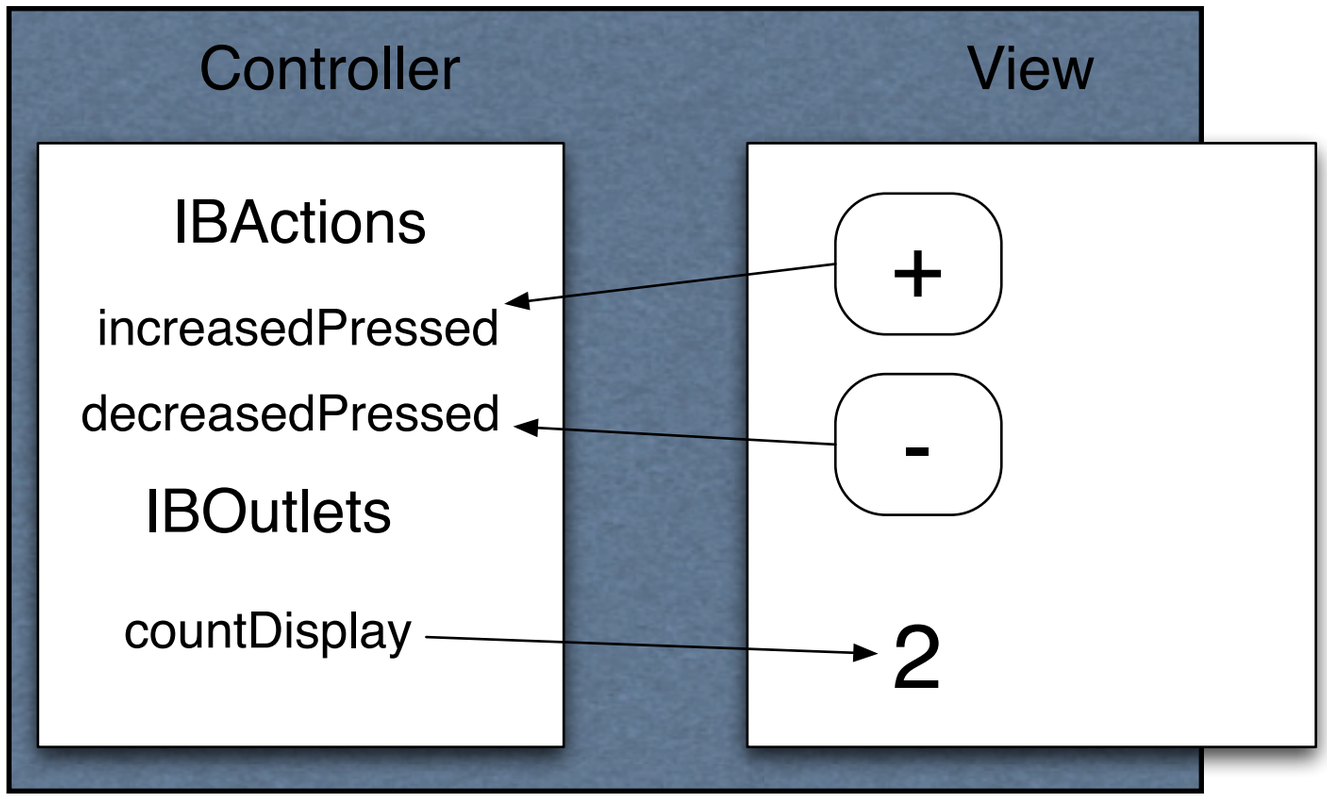
Interface Builder







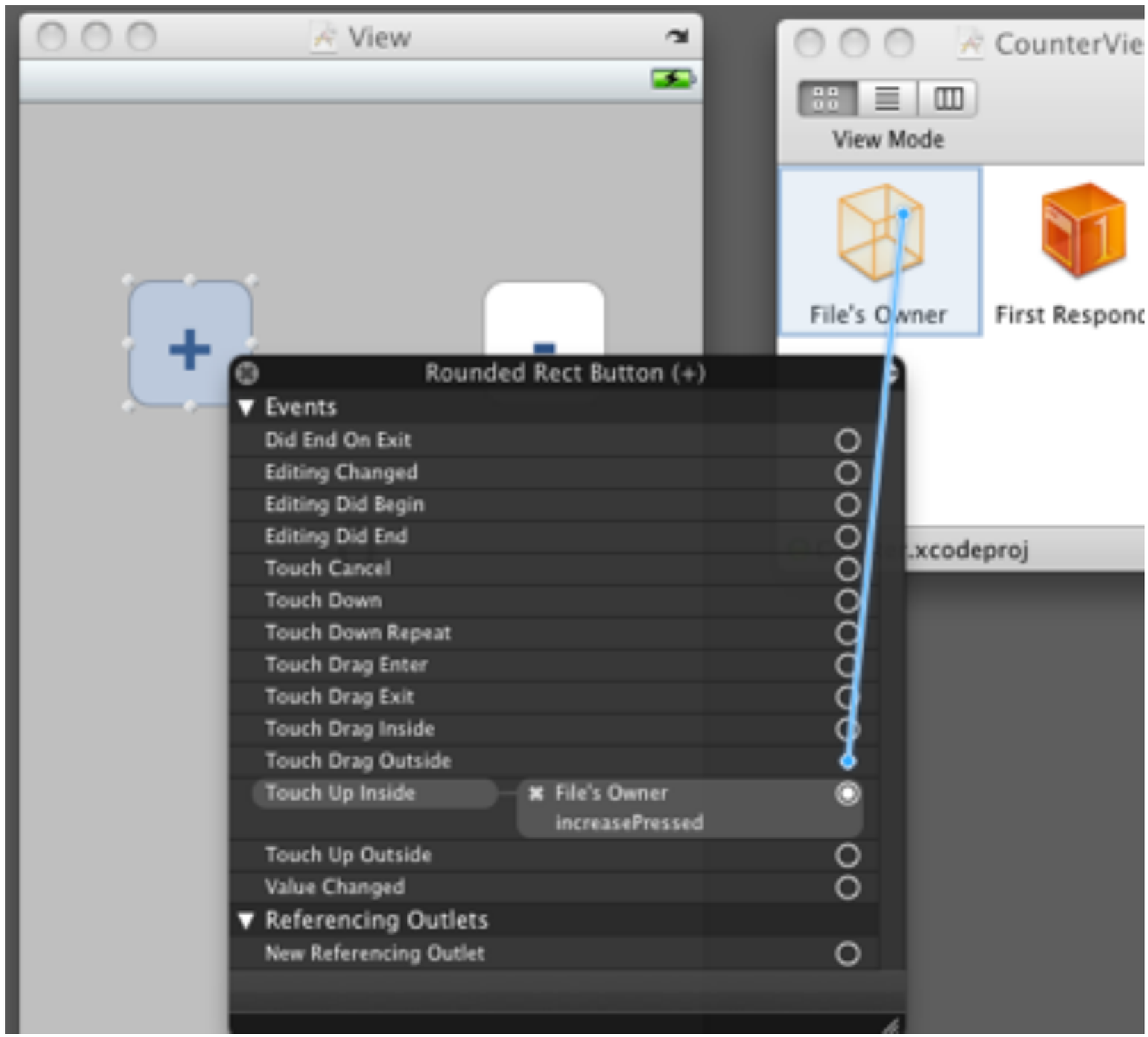
Declaring connections in Code




```
@interface CounterViewController : UIViewController {
      int * count;
}
@property (nonatomic, retain) IBOutlet UILabel * countDisplay;

- (IBAction) increasePressed;
- (IBAction) decreasePressed;
@end
```

Making the Connection in Interface Builder



The Connections

IBAction

- Method in controller
- Connect UI event to method
- UI element triggers method
- Typedef - is void

IBOutlet

- Instance variable in controller
- Reference ti UI element
- Empty typedef

Demo

CounterViewController.h

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

@interface CounterViewController : UIViewController {
	int count;
}
@property (nonatomic, retain) IBOutlet UILabel * countDisplay;

- (IBAction) increasePressed;
- (IBAction) decreasePressed;

@end
```

CounterViewController.h

```
#import "CounterViewController.h"
@implementation CounterViewController
@synthesize countDisplay;

- (void) displayCount {
  [countDisplay setText:[NSString stringWithFormat:@"%i", count]];
}

- (IBAction) increasePressed {
      count++;
      [self displayCount];
}

- (IBAction) decreasePressed {
      count--;
      [self displayCount];
}
```

```
- (void)dealloc {
    [super dealloc];
       [countDisplay release];
}
@end
```

CounterViewController

No changes made to generated code

Action Method options

```
- (void)actionMethod;
- (void)actionMethod:(id)sender;
- (void)actionMethod:(id)sender withEvent:(UIEvent *)event;
```

- (void)actionMethod;

When no data is needed from UI element

```
- (IBAction) increasePressed {
        count++;
        [self displayCount];
}
```

- (void)actionMethod:(id)sender;

When need data from sender

When need to interact with sender

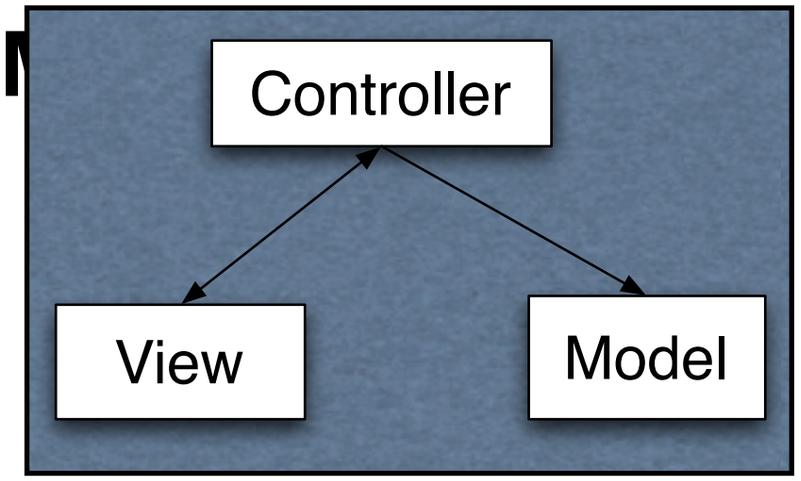
```
- (IBAction) increasePressed: (id) sender {
        UIButton * buttonPressed = (UIButton*) sender;
        if ([buttonPressed.titleLabel.text isEqualToString: @"+"]) {
                count++;
                [self displayCount];
                if (count > 5) {
                        [buttonPressed setTitle: @"X" forState: UIControlStateNormal];
                }
        }
}
```

- (void)actionMethod:(id)sender withEvent: (UIEvent *)event;

Event contains information about the UI event

```
- (IBAction) increasePressed: (id) sender withEvent: (UIEvent *) event {
    NSSet * touches = [event allTouches];
    count = count + [touches count];
    [self displayCount];
}
```

Where is the M



int count was model

Used default initial value

Counter Model

```
@interface Counter : NSObject {
        int count;
}

-(void) increase;
-(void) decrease;
-(int) count;

@end
```

```
@implementation Counter

-(void) increase { count++; }

-(void) decrease { count--; }

-(int) count { return count; }

- (id) init {
        if (self = [super init]) {
                count = 0;
        }
        return self;
}
@end
```

CounterViewController.h

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

@interface CounterViewController : UIViewController {
	Counter * count;
}
@property (nonatomic, retain) IBOutlet UILabel * countDisplay;

- (IBAction) increasePressed;
- (IBAction) decreasePressed;
@end
```

Using the model

```
- (void) displayCount {
  [countDisplay setText:[NSString stringWithFormat:@"%i", [count count]]];
}

- (IBAction) increasePressed {
        [count increase];
        [self displayCount];
}

- (IBAction) decreasePressed {
        [count decrease];
        [self displayCount];
}
```

Creating & releasing the Model

```
- (void)dealloc {
   [super dealloc];
       [countDisplay release];
       [count release];
}

- (void) awakeFromNib {
       NSLog(@"awakefromNib");
       count = [Counter new];
}
```

App life cycle

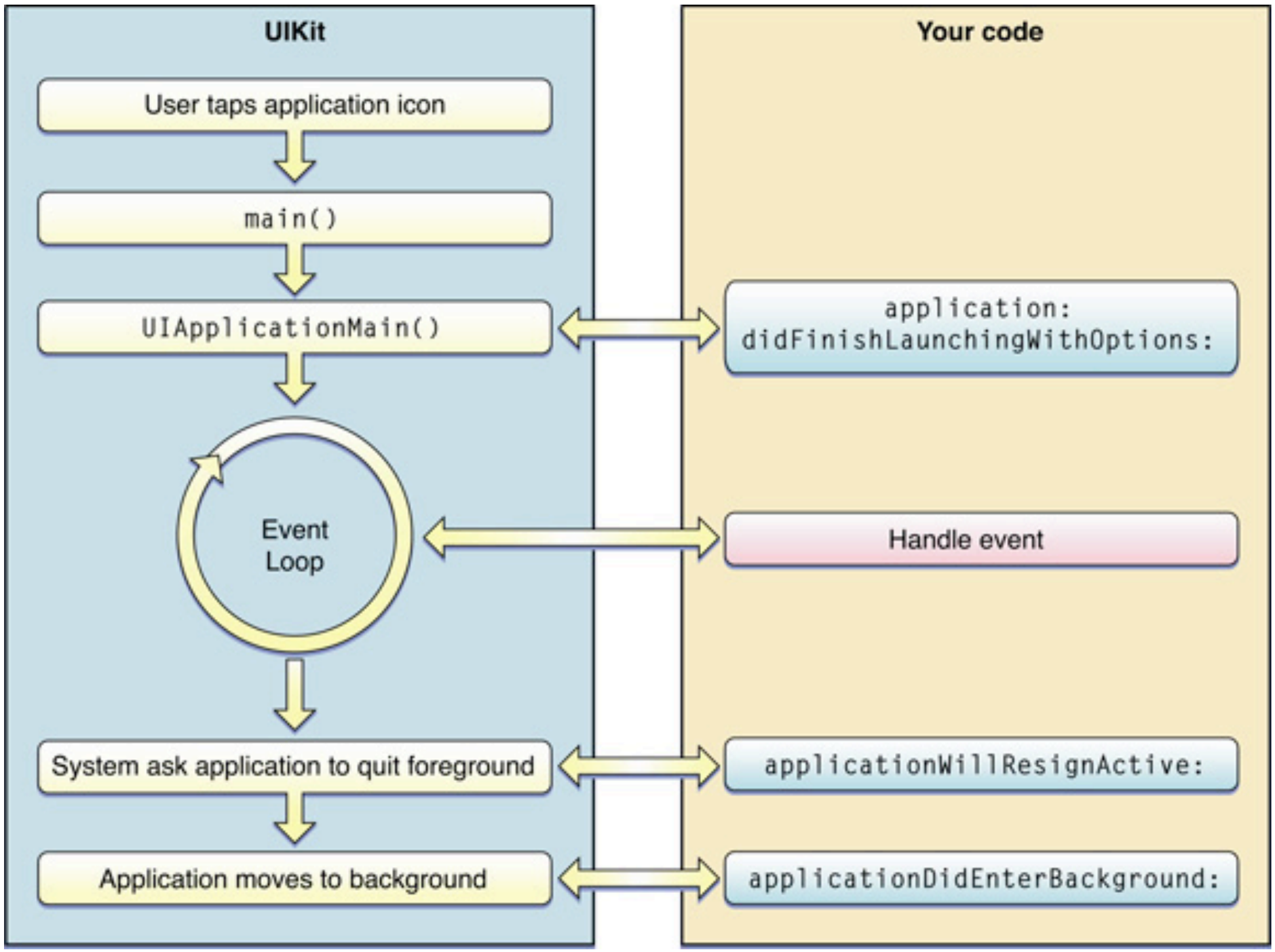


diagram from "iPhone Application Programming Guide"

Starting points for your app code

App Delegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *) launchOptions

Controller

- (id) initWithCoder:(NSCoder *) aDecoder
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
- (void) awakeFromNib

Creating Views in

When count >=3
Create & display "Done"

CounterViewController.h

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

@interface CounterViewController : UIViewController {
	Counter * count;
}
@property (nonatomic, retain) IBOutlet UILabel * countDisplay;
@property (nonatomic, retain) IBOutlet UILabel * doneLabel;

- (IBAction) increasePressed;
- (IBAction) decreasePressed;
@end
```

CounterViewController.m

```
#import "CounterViewController.h"

@implementation CounterViewController

@synthesize countDisplay;
@synthesize doneLabel;

- (void) displayCount {
  [countDisplay setText:[NSString stringWithFormat:@"%i", [count count]]];
}
```

CounterViewController.m

```
- (IBAction) increasePressed {
        [count increase];
        [self displayCount];
        if ([count count] >= 3) {
                [self displayDoneLabel];
        }
}
```

```
- (void) displayDoneLabel {
        if (doneLabel == nil) {
                [self createDoneLabel];
        }
        else {
                doneLabel.hidden = NO;
        }
}
```

CounterViewController.m

```
- (void) createDoneLabel {
    CGRect frame = CGRectMake(115, 300, 90, 30);
    [self setDoneLabel: [[UILabel alloc] initWithFrame:frame]];
    doneLabel.textAlignment = UITextAlignmentCenter;
    doneLabel.font = [UIFont fontWithName:@"Verdana" size:24];
    doneLabel.text = @"Done";
    [self.view addSubview:doneLabel];
}
```

CounterViewController.m

```
- (IBAction) decreasePressed {
        [count decrease];
        [self displayCount];
        if ([count count] < 3) {
                doneLabel.hidden = YES;
            }
}
```