MVC and Interface Builder

IAP 2010 ❄

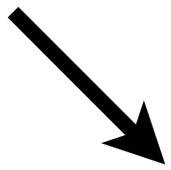
iphonedev.csail.mit.edu

edward benson / eob@csail.mit.edu

Information-Driven Applications

Application Flow

UIApplication



Main NIB Initialized

UIAppDelegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
{
    [window makeKeyAndVisible];
}
```

Initialize Your Root Controller & Interface

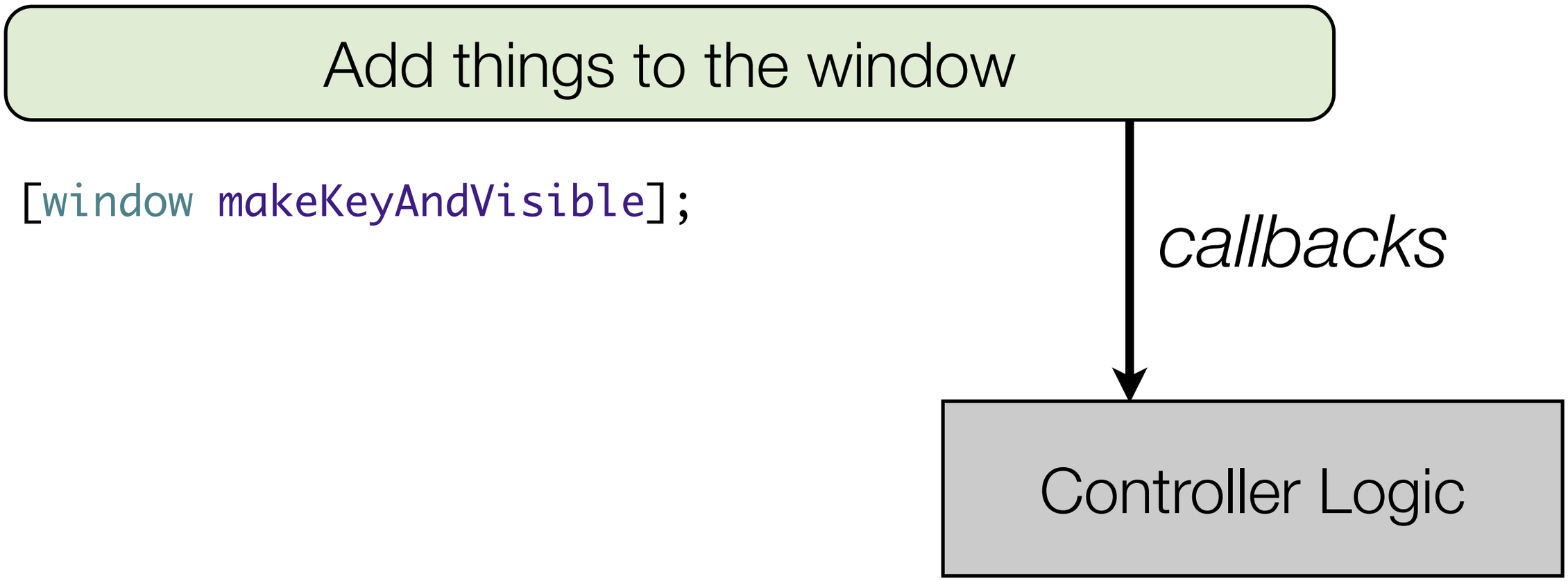
After which point it is..

So Application Design and
UI Design are intimately paired.

Application Flow

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
{

    [window makeKeyAndVisible];
}
```



Exercise 1

App Delegate .h

```
@interface RPS2AppDelegate : NSObject <UIApplicationDelegate> {
    NSManagedObjectModel *managedObjectModel;
    NSManagedObjectContext *managedObjectContext;
    NSPersistentStoreCoordinator *persistentStoreCoordinator;
    UITableViewController *tableViewController;
    UIWindow *window;
}
```

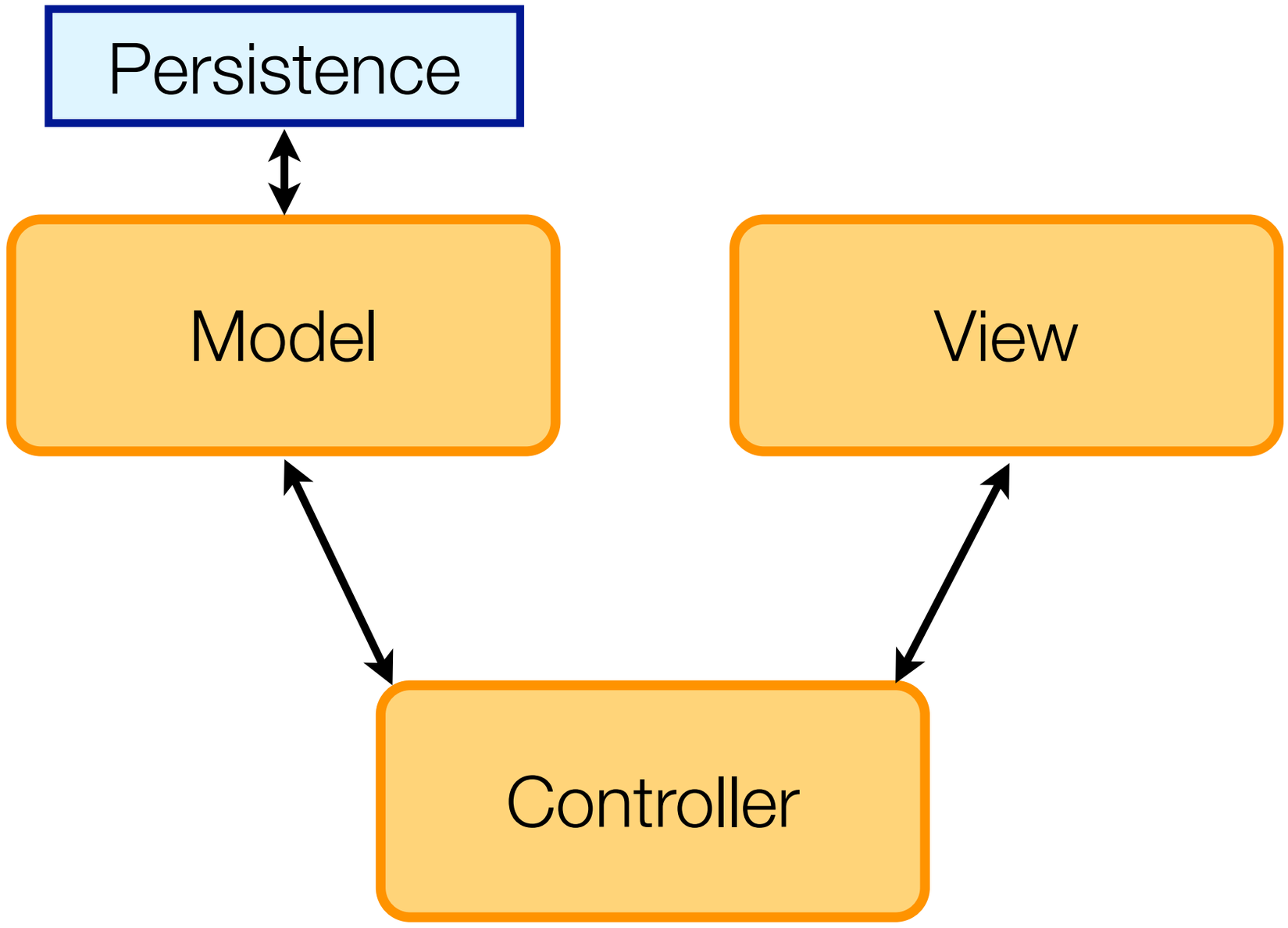
App Delegate .m

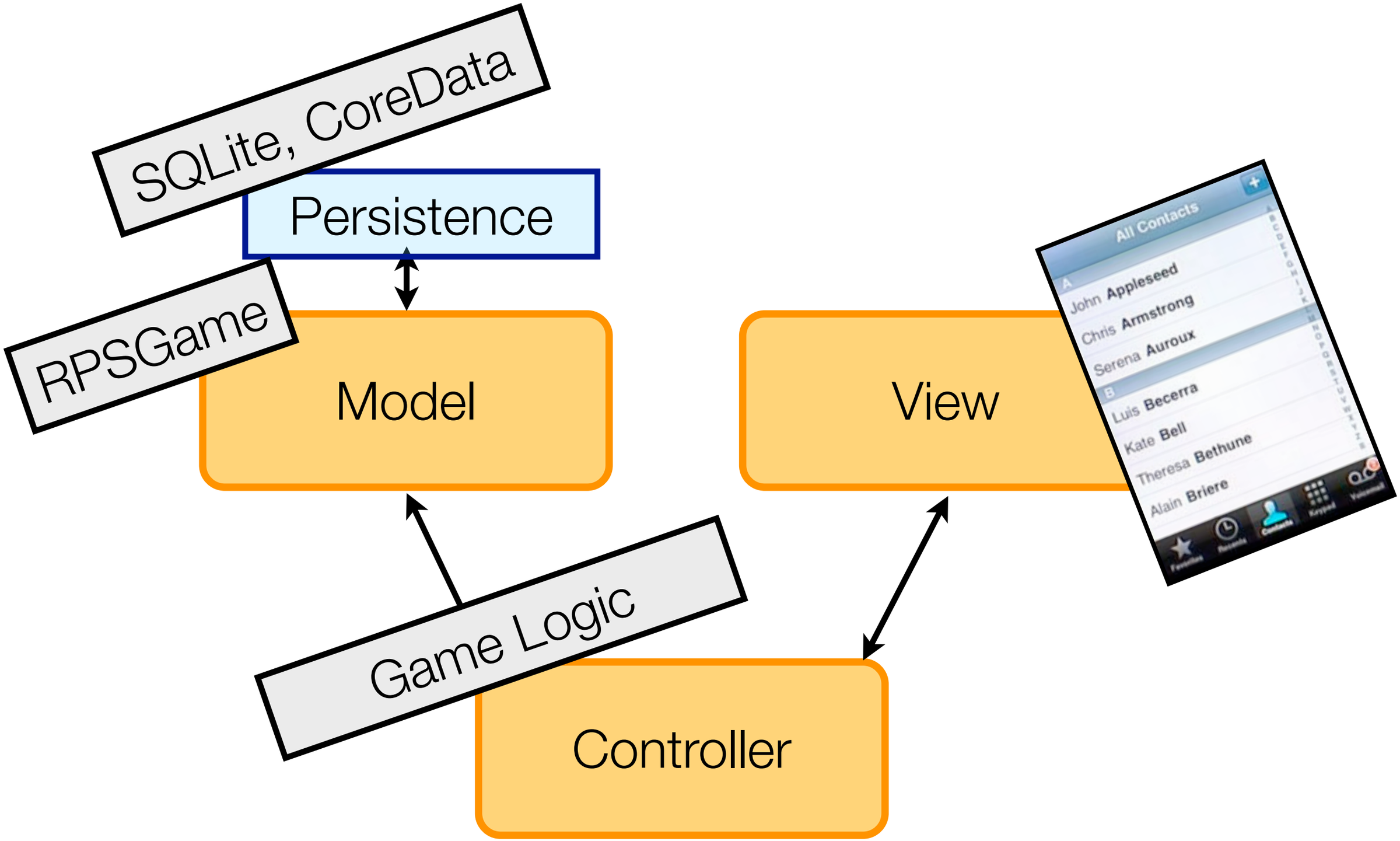
```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    // Override point for customization after app launch
    tableViewController = [[UITableViewController alloc]
                           initWithStyle:UITableViewStylePlain];
    [window addSubview:tableViewController.view];
    [window makeKeyAndVisible];
}
```

App Delegate .m -- dealloc method

```
[tableViewController release];
```

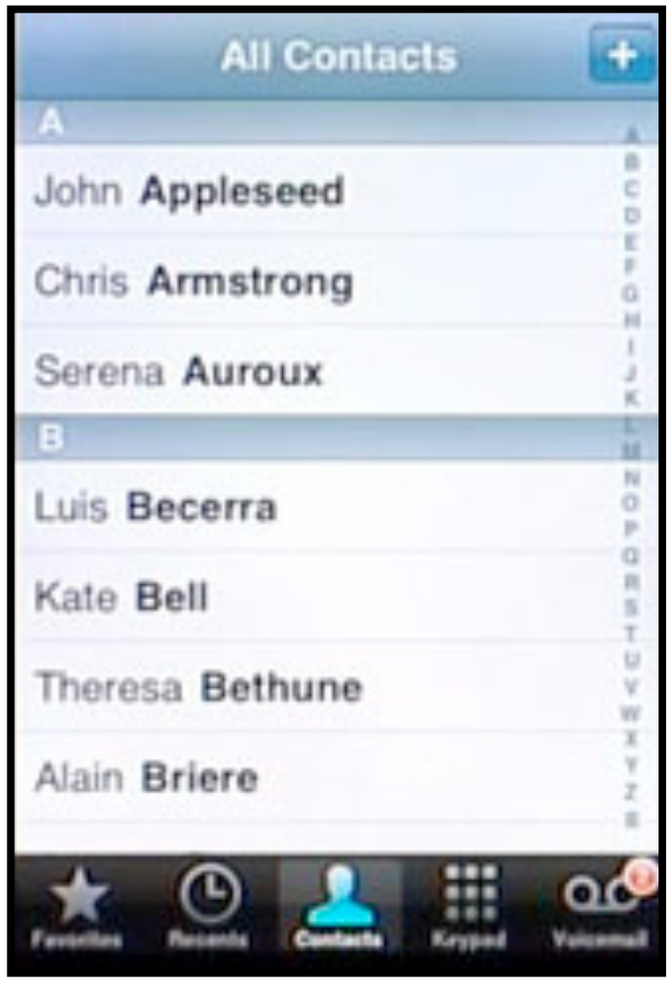
Model-View-Controller Design


Persistence
Model
View
Controller


SQLite, CoreData
Persistence
RPSGame
Model
View
All Contacts
John Appleseed
Chris Armstrong
Serena Auroux
Luis Becerra
Kate Bell
Theresa Bethune
Alain Briere
Game Logic
Controller

MVC on the iPhone

- You rarely have a View without a ViewController



UITableView

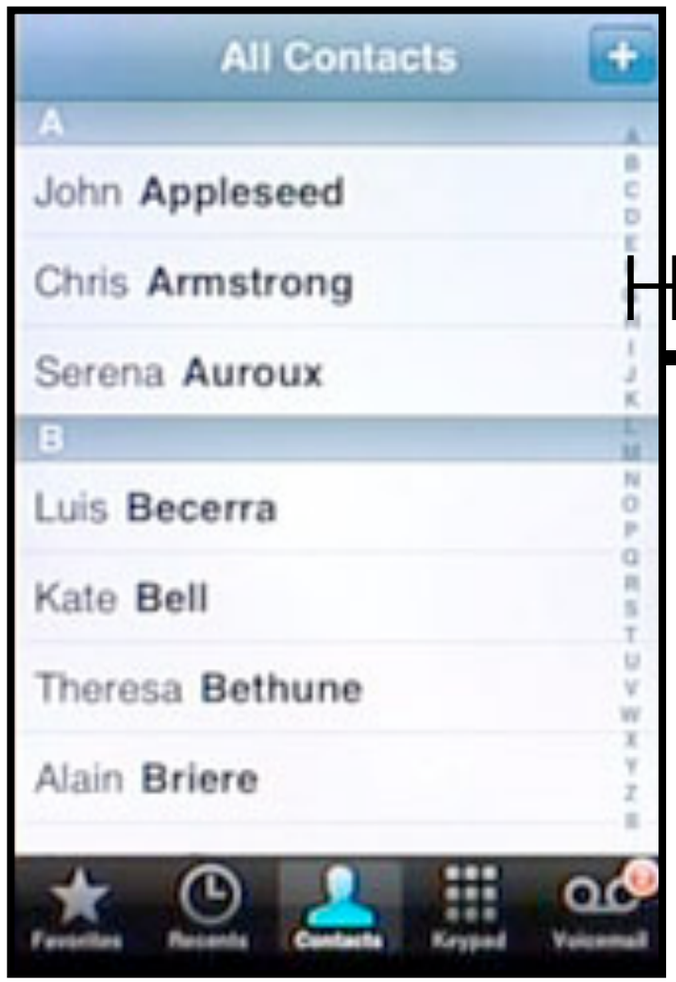
Controller

UITableViewController

MVC on the iPhone

- The view can drive the relationship, asking things of the controller

(Delegate Pattern)



How many sections?

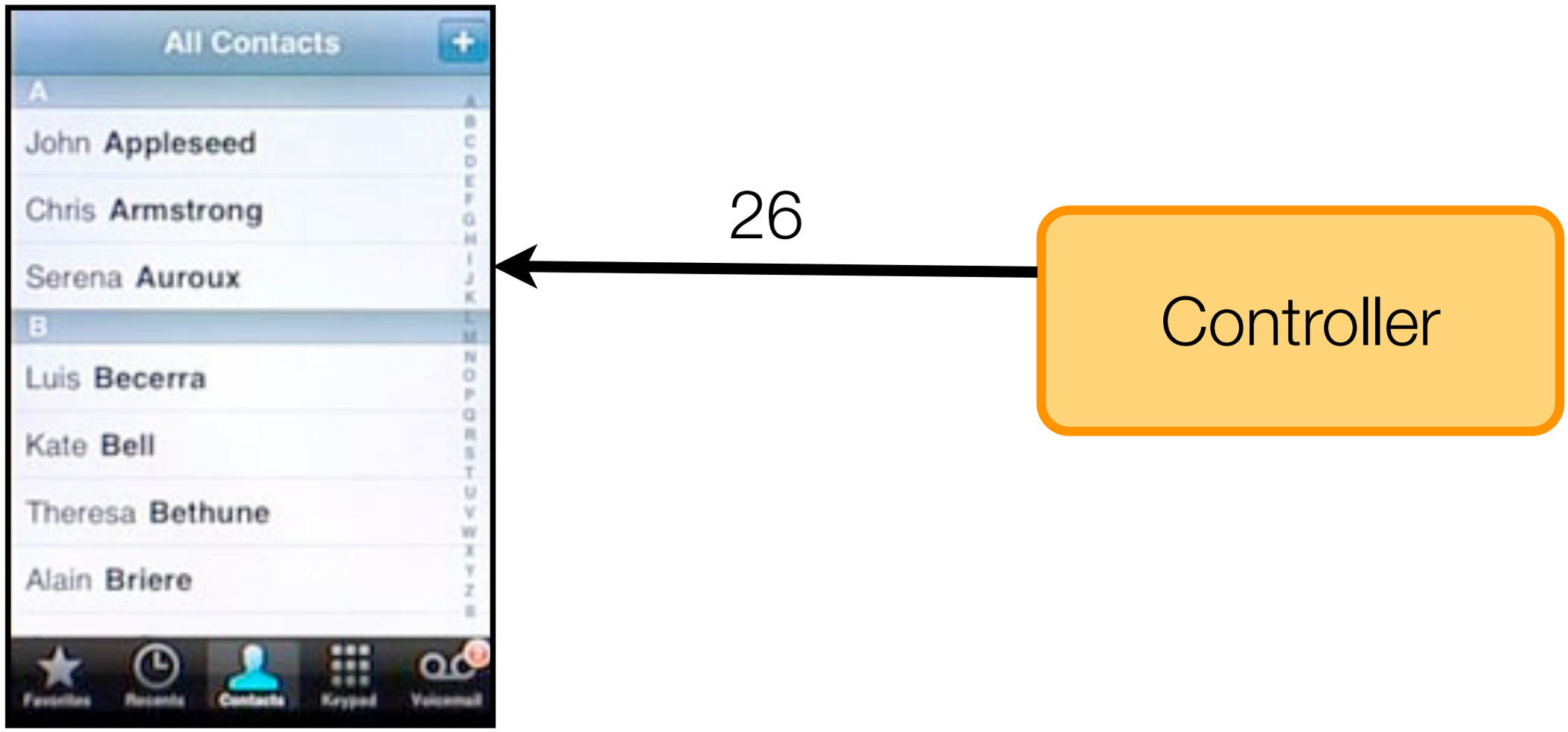
Controller

UITableView

UITableViewController

MVC on the iPhone

- The View asks things of the controller

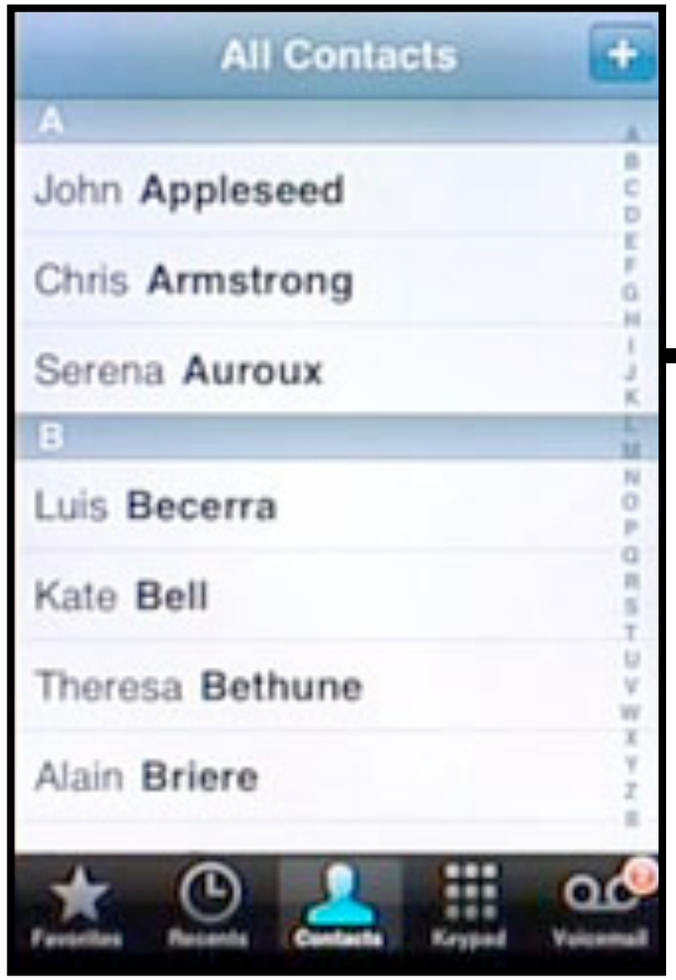


UITableView

UITableViewController

MVC on the iPhone

- The View asks things of the controller



How many rows in section 1?

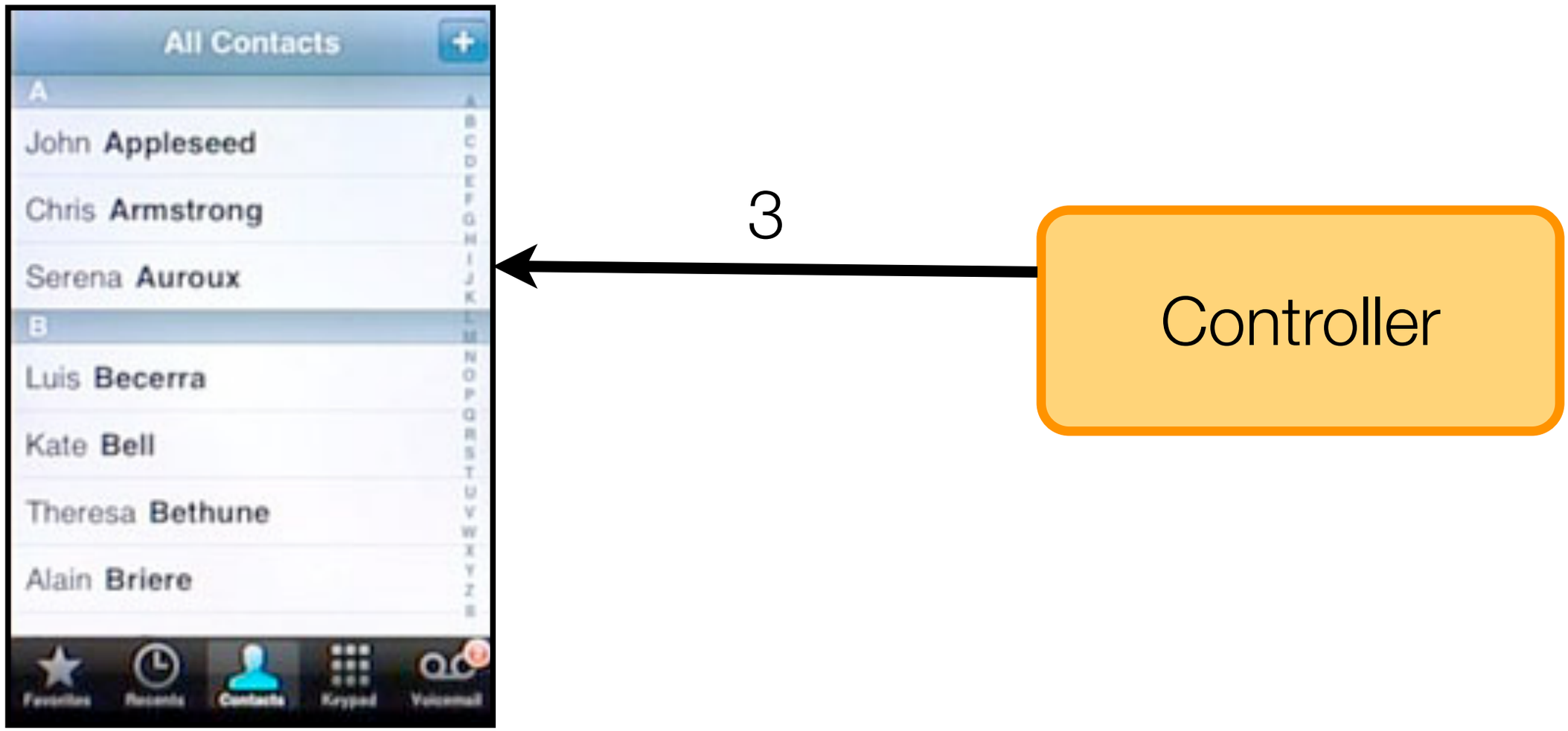
Controller

UITableView

UITableViewController

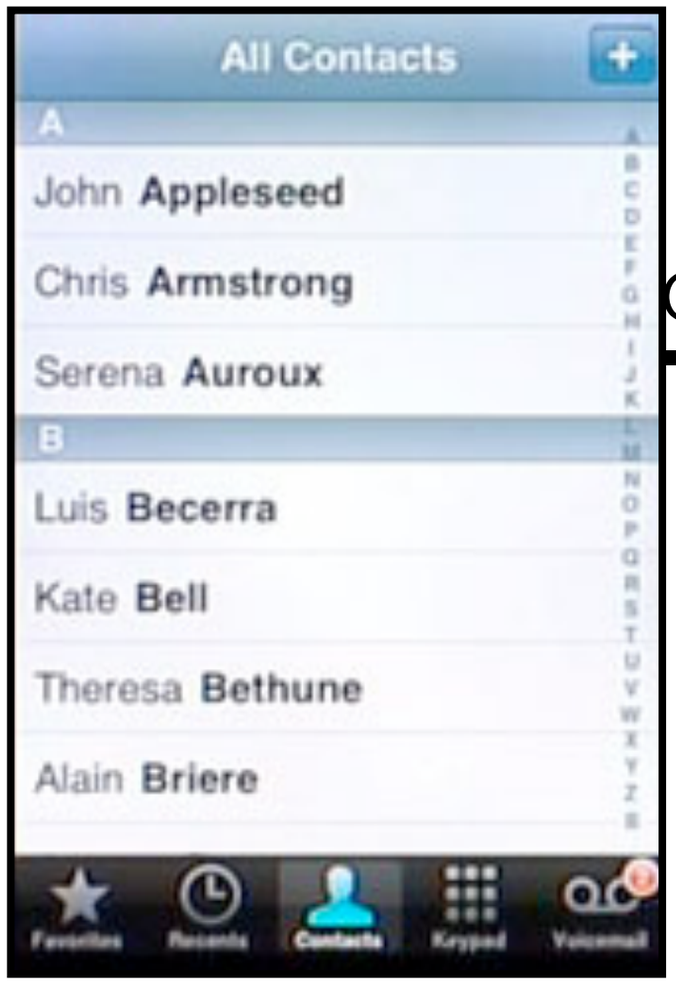
MVC on the iPhone

- The View asks things of the controller



MVC on the iPhone

- The View asks things of the controller



What is the cell object for row 1:1?

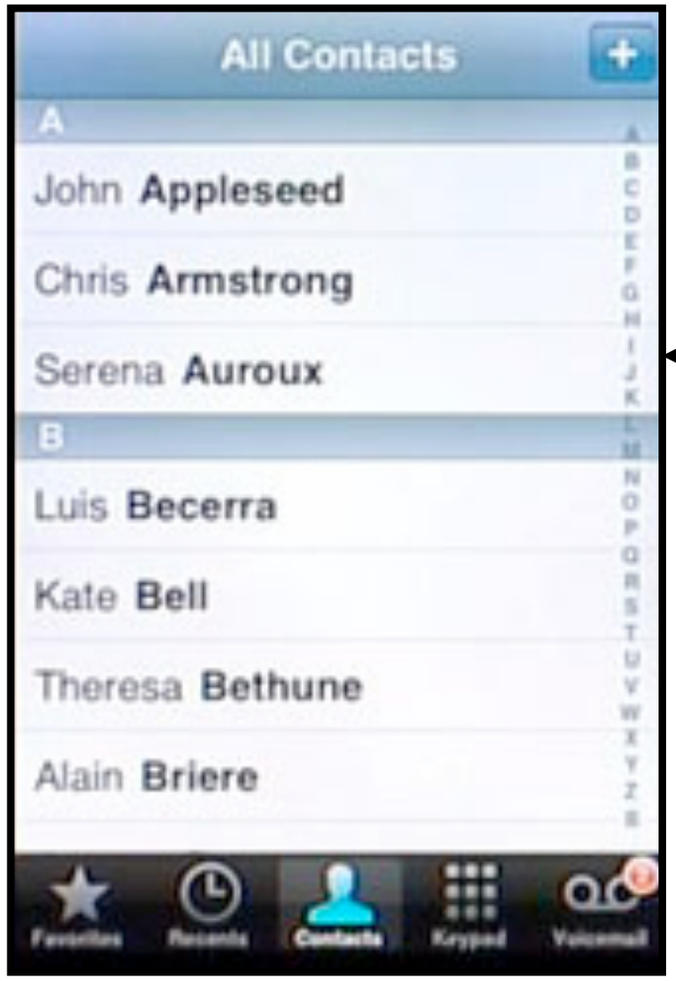
Controller

UITableView

UITableViewController

MVC on the iPhone

- The View asks things of the controller



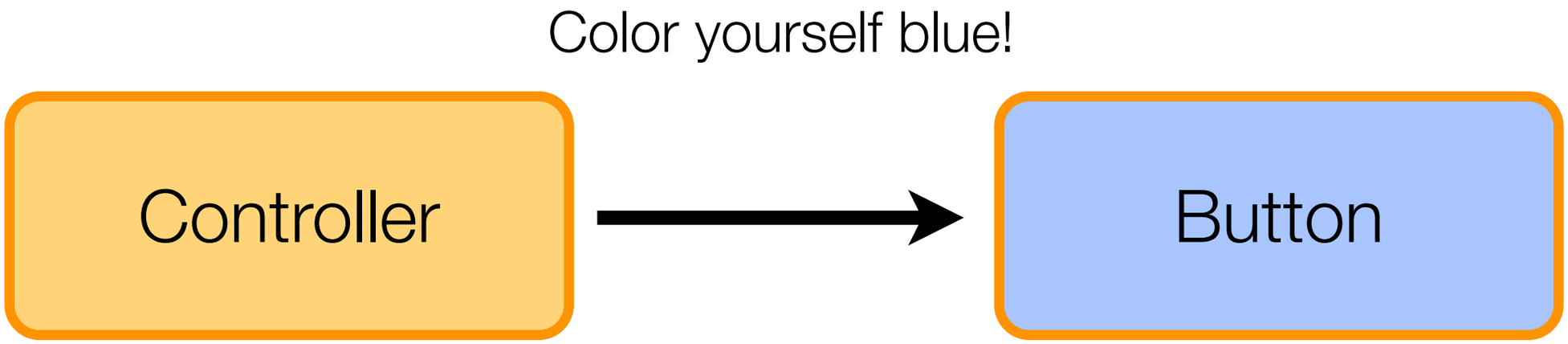
John Appleseed

Controller

UITableView

UITableViewController

Or the controller can instruct the view



MyCustomController

UIButton

Three ways to organize this relationship

---

1. Use a pre-packaged view and just implement its delegate in the controller

Tables
Camera
Maps
Address Book
etc

Three ways to organize this relationship

---

2. Have the controller programmatically construct a custom view

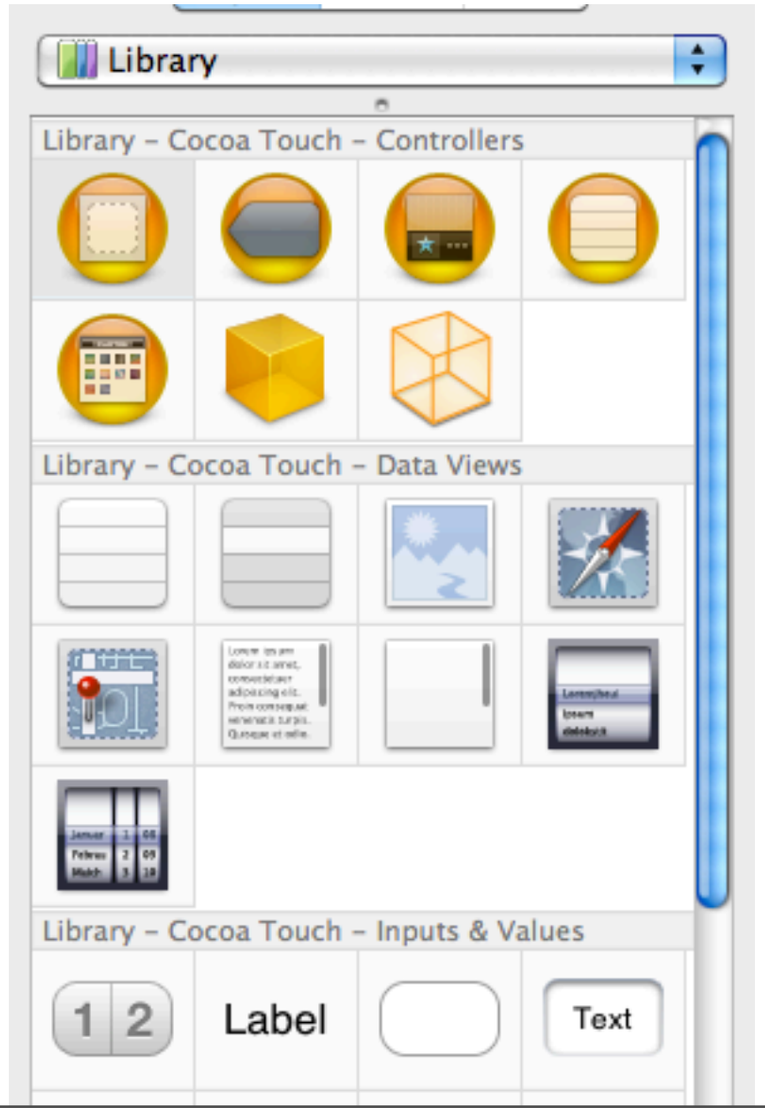
```
overlaySize = CGRectMake(0,
                         self.tableView.height - 22,
                         self.tableView.size.width,
                         22);
TTActivityLabel *banner = [[TTActivityLabel alloc]
                           initWithStyle:TTActivityLabelStyleBlackBanner]
banner.text = [ModelBase syncMessage];
[banner sizeToFit];
```

This can consist of a lot of pixel math

Three ways to organize this relationship

---

3. Create a custom view in InterfaceBuilder and then drive it using the controller

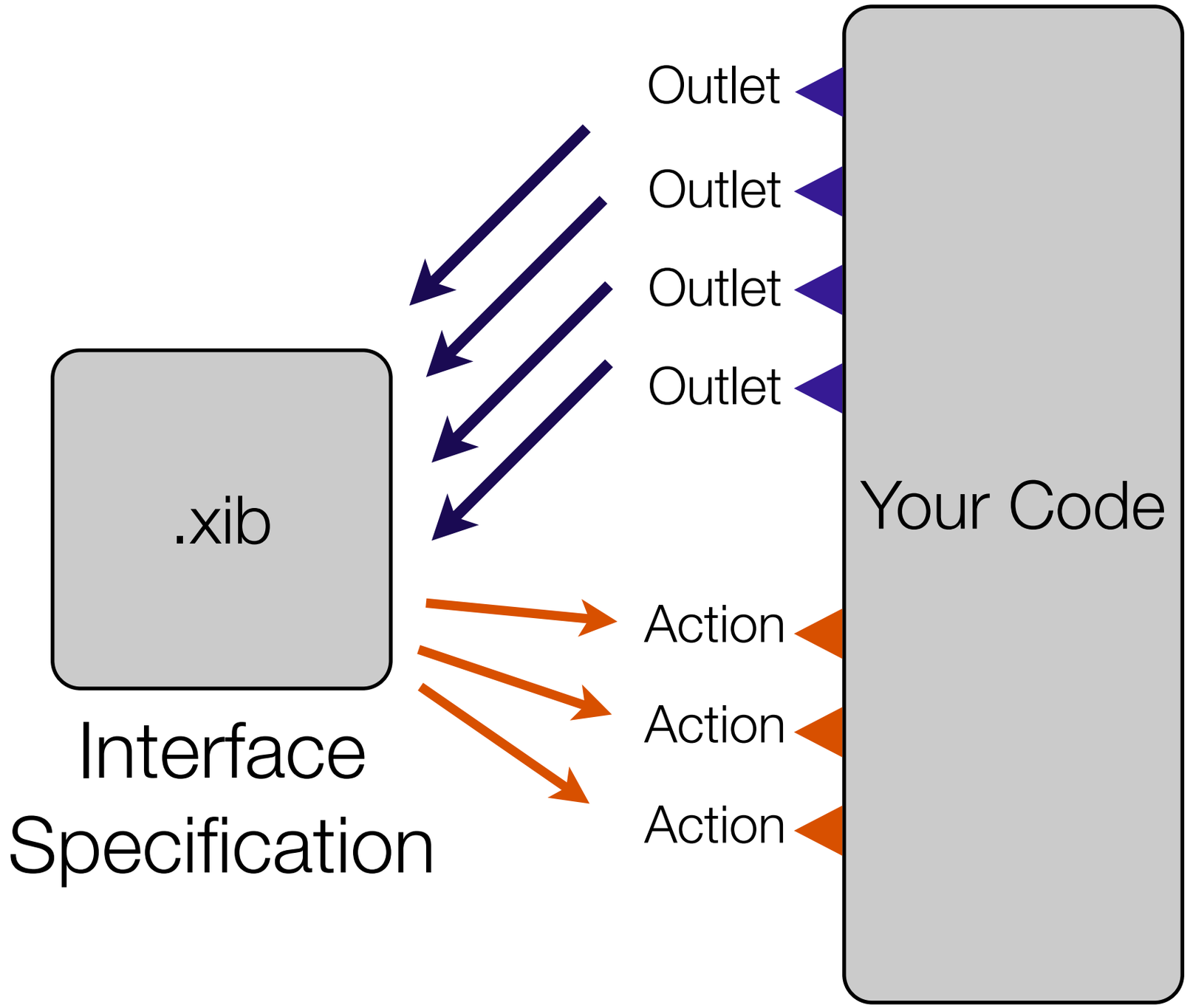


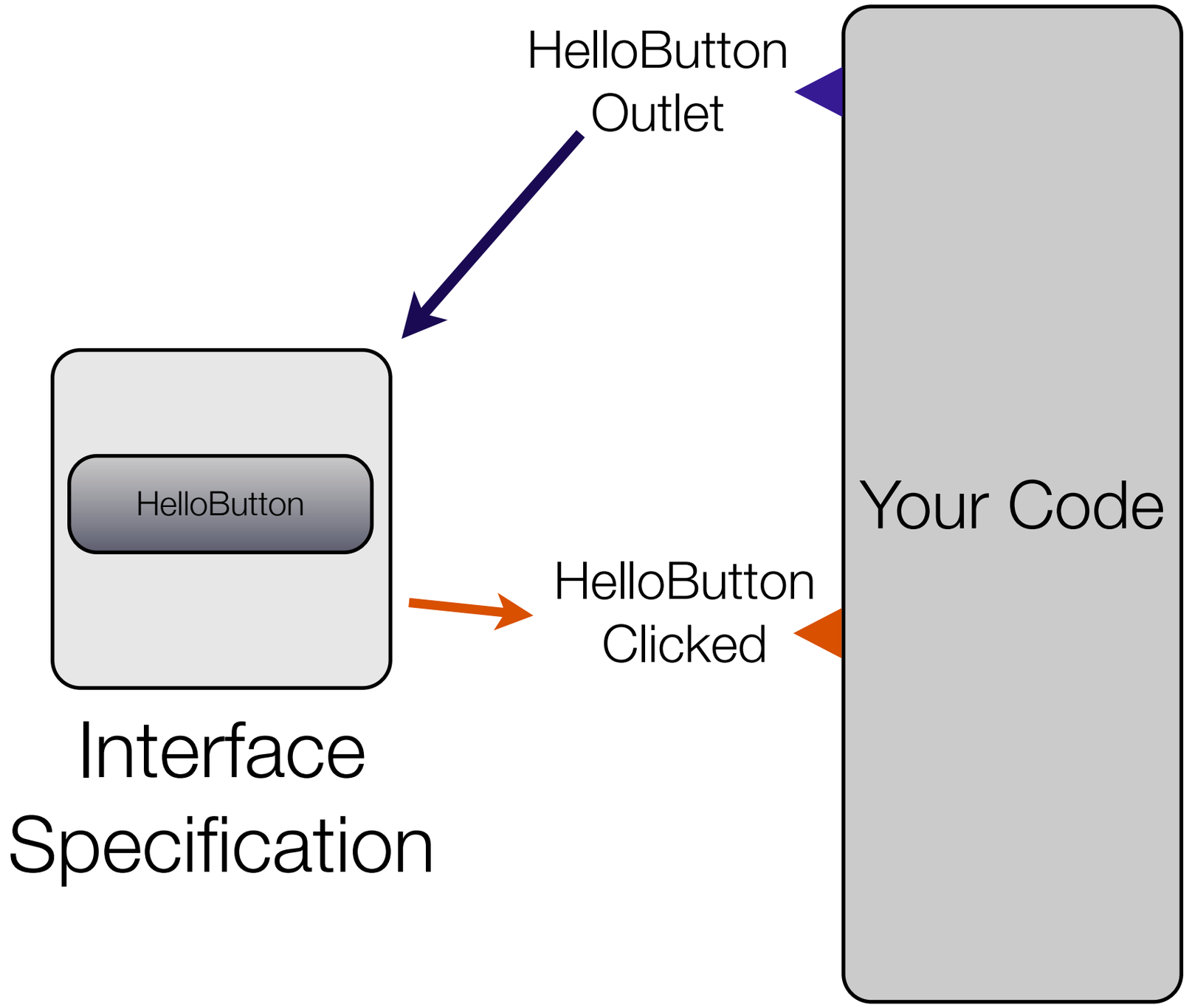
Interface Builder

.xib

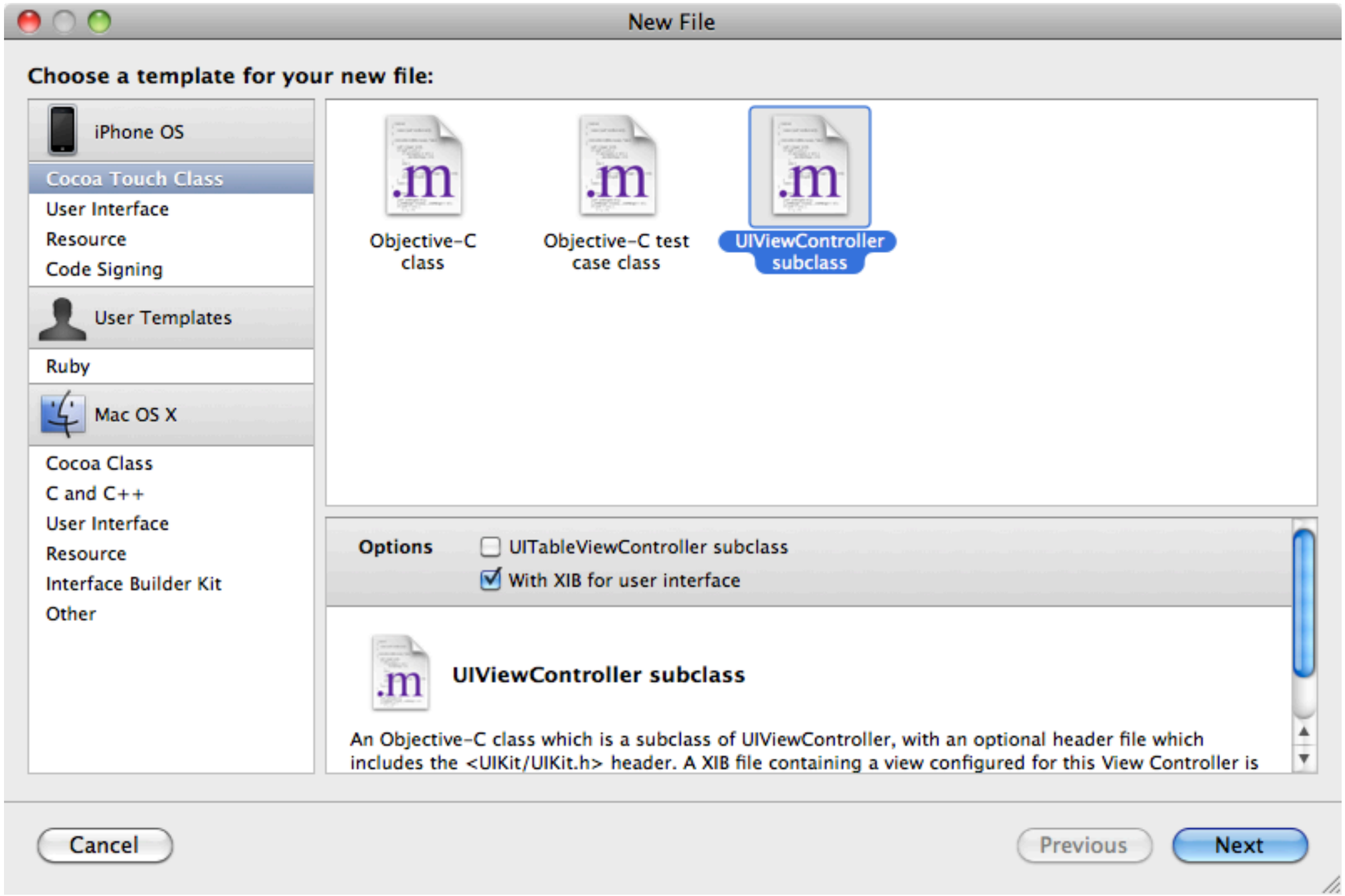
Interface Specification

Your Code

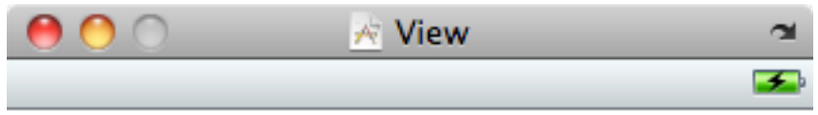

Outlet
Outlet
Outlet
Outlet
.xib
Your Code
Action
Action
Action
Interface
Specification


HelloButton
Outlet
HelloButton
HelloButton
Clicked
Your Code
Interface
Specification

Exercise 2


New File
Choose a template for your new file:
iPhone OS
Cocoa Touch Class
User Interface
Resource
Code Signing
User Templates
Ruby
Mac OS X
Cocoa Class
C and C++
User Interface
Resource
Interface Builder Kit
Other
Objective-C class
Objective-C test case class
UIViewController subclass
Options
UITableViewController subclass
With XIB for user interface
UIViewController subclass
An Objective-C class which is a subclass of UIViewController, with an optional header file which includes the <UIKit/UIKit.h> header. A XIB file containing a view configured for this View Controller is
Cancel
Previous
Next

Exercise 2



Exercise 2

Now swap the Table View Controller you used before for the RPSGameViewController you just created

Exercise 3 - Outlets and Actions

.h

```
@interface RPSGameViewController : UIViewController
{
  IBOutlet UIButton *rockButton;
  IBOutlet UIButton *paperButton;
  IBOutlet UIButton *scissorsButton;
  IBOutlet UILabel *responseLabel;
}

- (IBAction)rockClicked:(id)sender;
- (IBAction)paperClicked:(id)sender;
- (IBAction)scissorsClicked:(id)sender;

@end
```

Exercise 3 - Outlets and Actions

In interface builder, wire them together

Then Run it

Why does the app crash when you click a button?

Debugging Tips

Exercise 3 - Outlets and Actions

```
@implementation RPSGameViewController

- (IBAction)rockClicked:(id)sender {
}

- (IBAction)paperClicked:(id)sender {
}

- (IBAction)scissorsClicked:(id)sender {
}
```

...(implementation continues)...

Exercise 3 - Outlets and Actions

```
- (IBAction)rockClicked:(id)sender {
  responseLabel.text = @"The strongest of foes!";
}

- (IBAction)paperClicked:(id)sender {
  responseLabel.text = @"Cunning and underrated!";
}

- (IBAction)scissorsClicked:(id)sender {
  responseLabel.text = @"Deadly and quick!";
}
```

...(implementation continues)...

Categories

One last Objective-C Feature

Categories provide a way to extend a class you did (or didn't!) write

Be careful -- overuse can get you into trouble

@interface NSString

@interface NSString(XMLSerialization)

-(NSData *)toXML

@interface NSString(PigLatin)

-(NSString *)toPigLatin

Ex 4

NSString+RPS.h

```
@interface NSString(RPS)
-(BOOL)rpsBeats:(NSString *)other;
@end
```

NSString+RPS.m

```
-(BOOL)rpsBeats:(NSString *)other {
   if (([self isEqualToString:@"rock"] && [other isEqualToString:@"scissors"]) ||
       ([self isEqualToString:@"scissors"] && [other isEqualToString:@"paper"]) ||
       ([self isEqualToString:@"paper"] && [other isEqualToString:@"rock"])) {
       return YES;
   }
   return NO;
}
```

Main App Delegate

```
if ([@"paper" rpsBeats:@"rock"]) {
   NSLog(@"All is right in the world!");
}
```

Lab

Extend your program so it has **three** labels.

The first button click sets the first label,
The second button click sets the second label,

And then the winner is declared in the third button click