CS 696 Mobile Application Development
Fall Semester, 2010
Doc 10 Tab Bars & Pickers
Sep 30, 2010

Some Code Organization Issues

Auto generated Commented out Methods

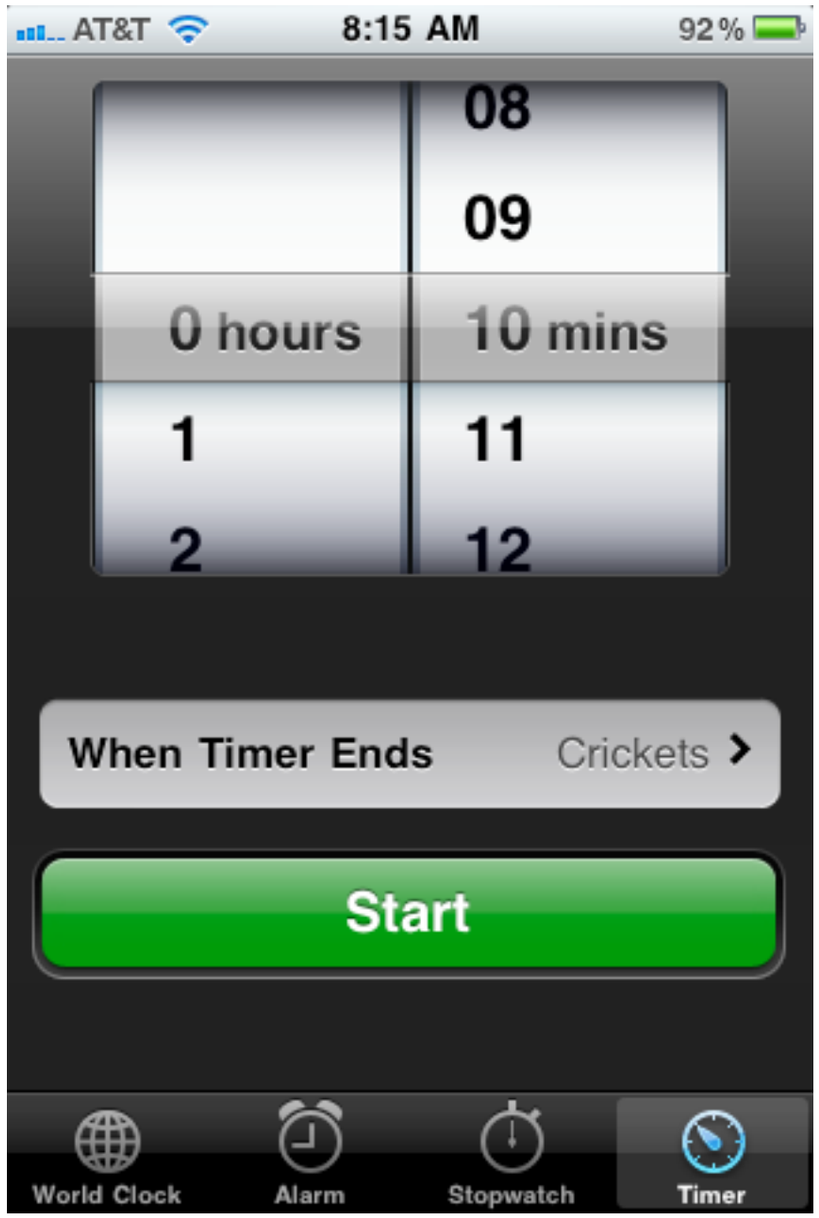
```
/*
// Override to allow orientations other than the default portrait orientation.
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}
*/
```

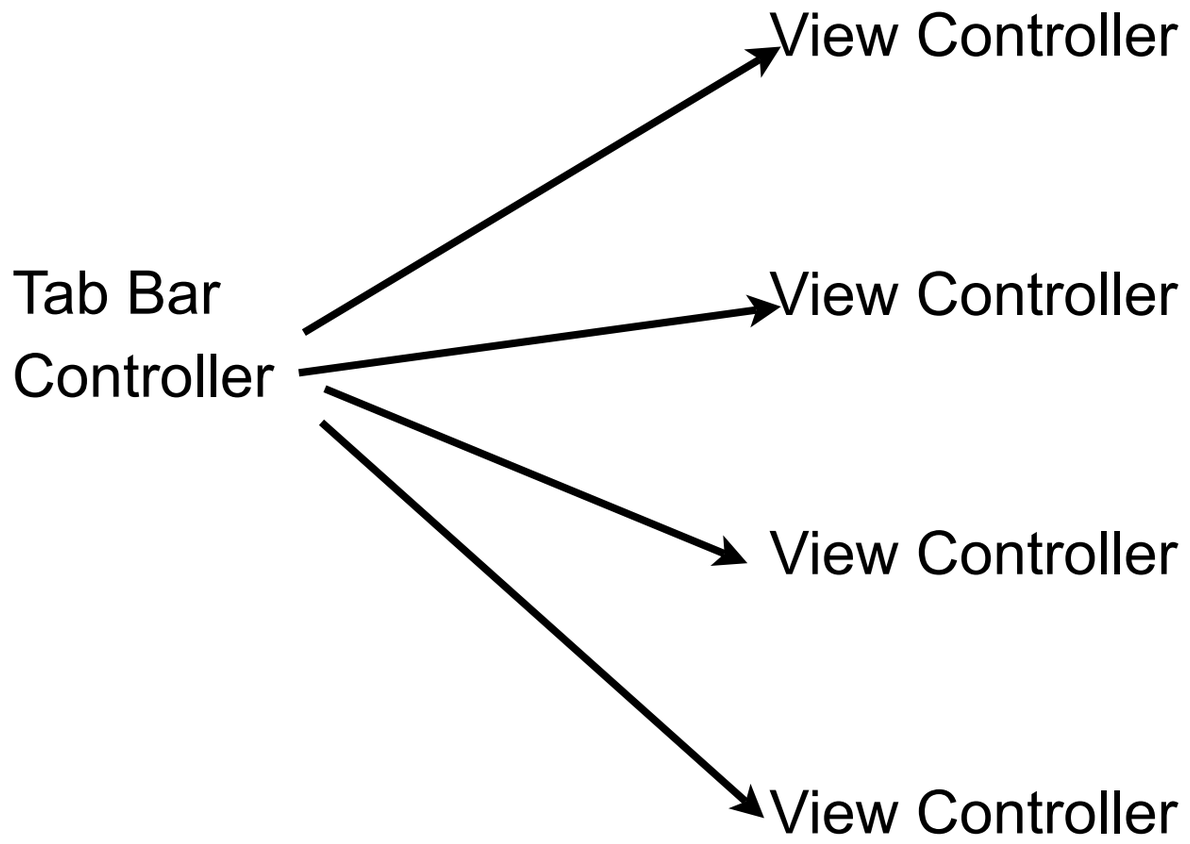
Code pragma's

```
#pragma mark -
#pragma mark Picker Data Source Methods

- (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView
{
    return 2;
}
```

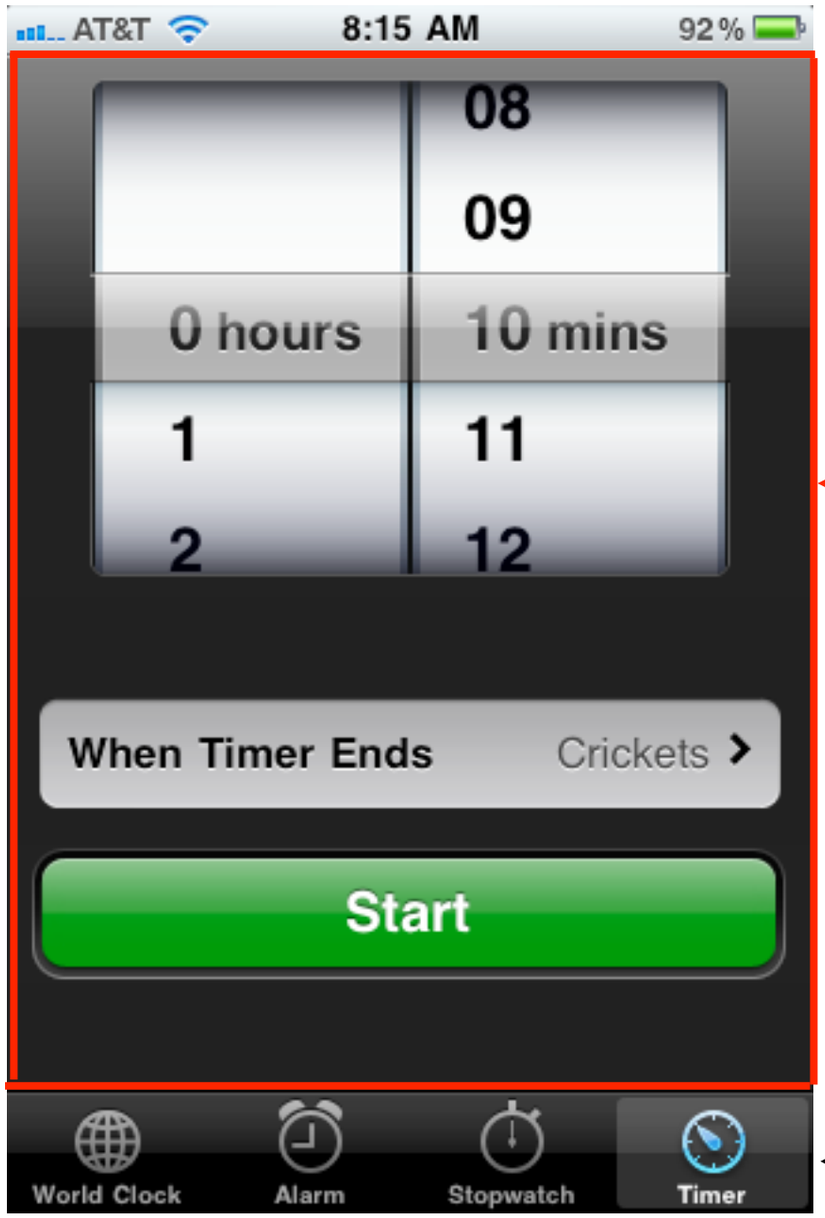
Tab Bars

Tab Bars


AT&T
8:15 AM
92%
08
09
0 hours
10 mins
1
11
2
12
When Timer Ends
Crickets
Start
World Clock
Alarm
Stopwatch
Timer


Tab Bar Controller
View Controller
View Controller
View Controller
View Controller

Tab Bars

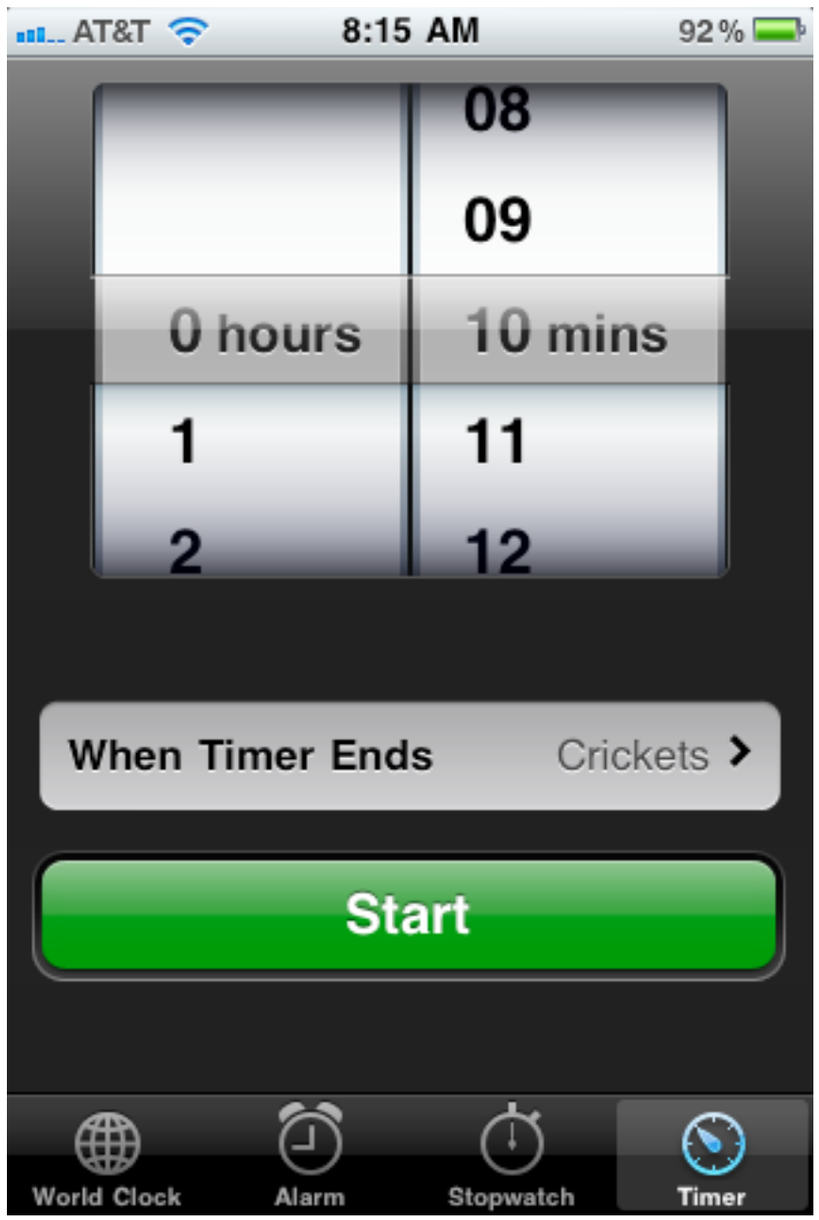


Selected view controller's view

Tab bar view controller view

Titles of each view controller

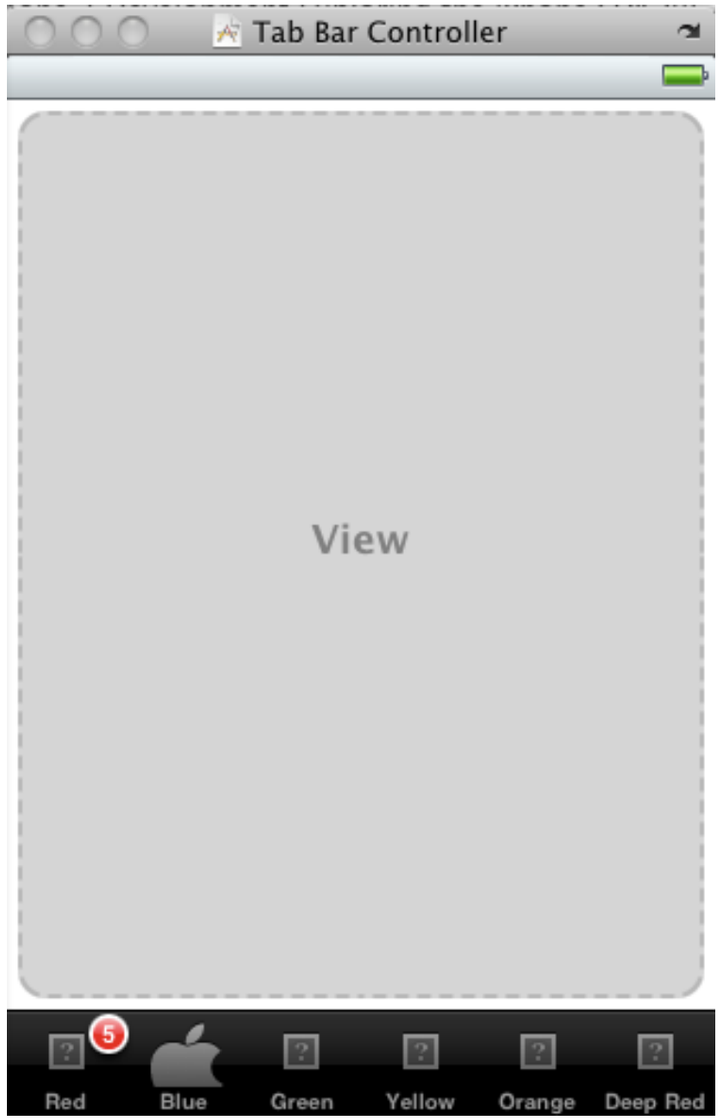
Tab Bars

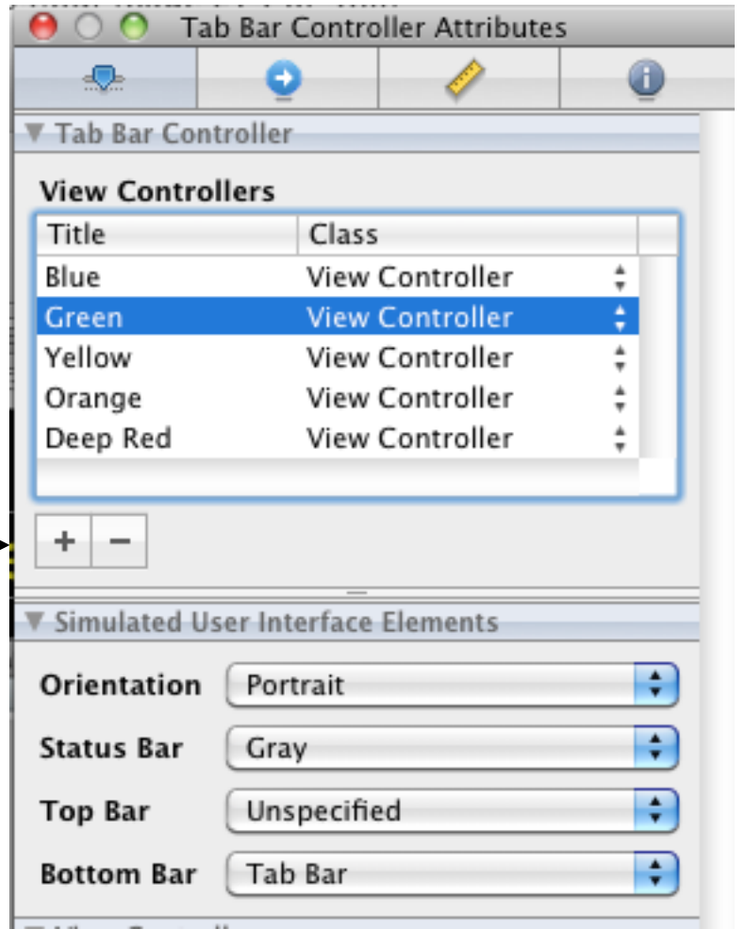


Separate xib files for
TabController
Each subview

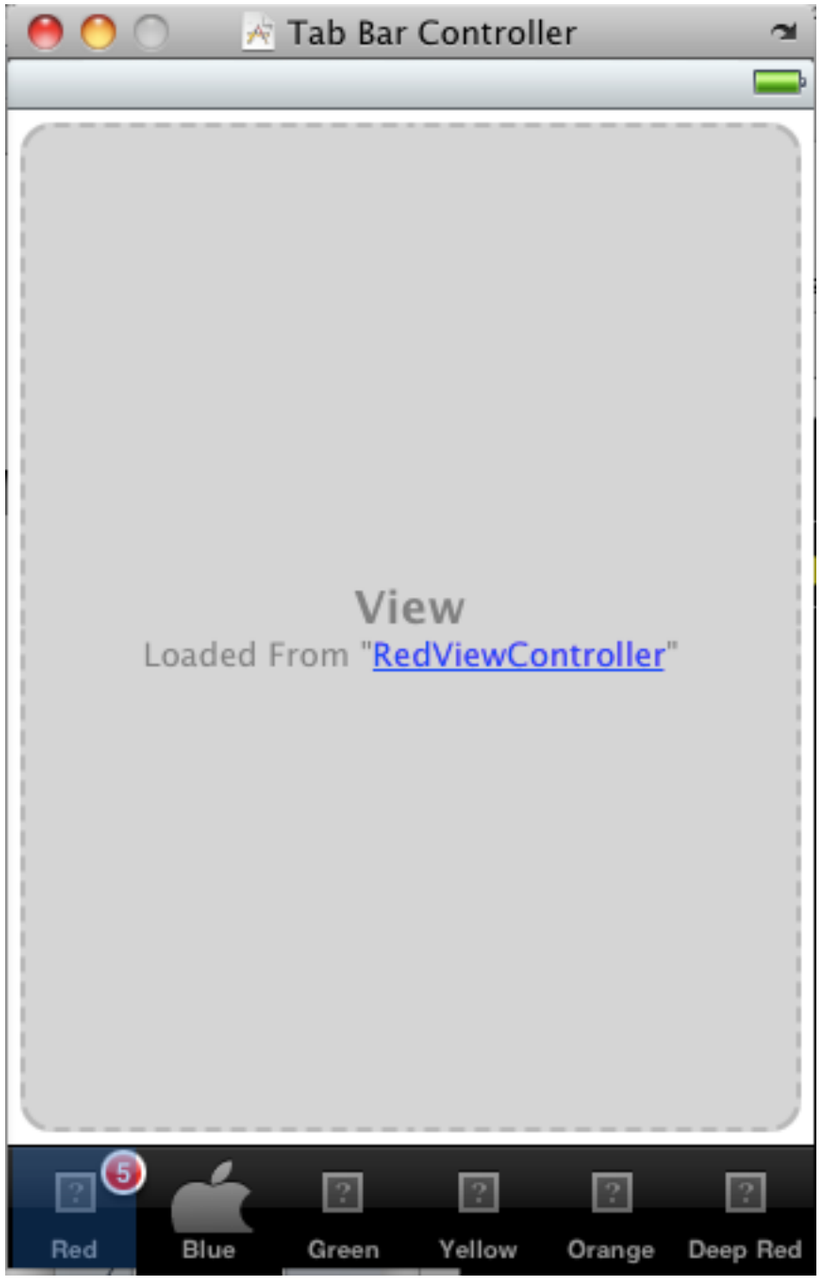
Tab bar controller handles
displaying subviews

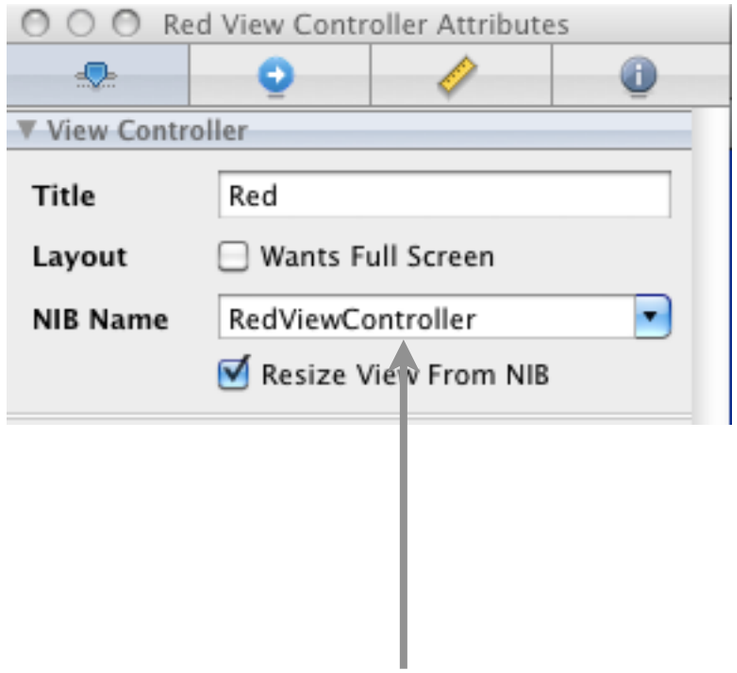
Adding More Tabs

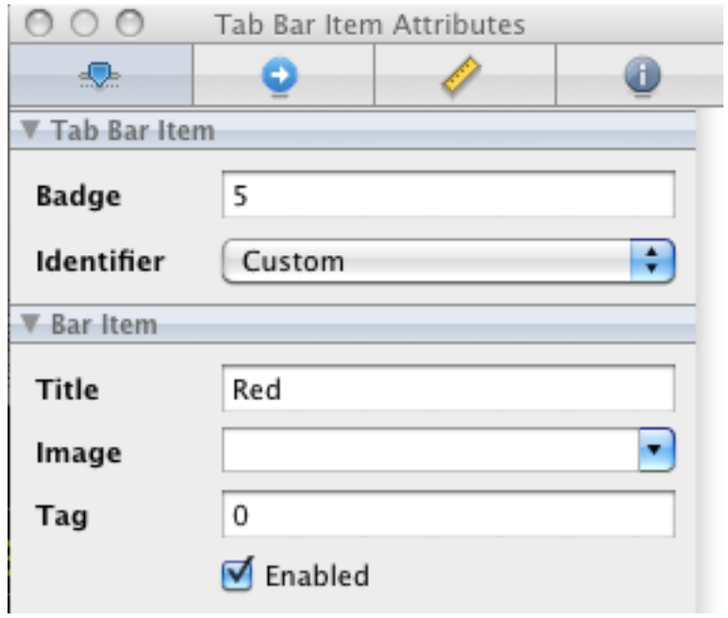




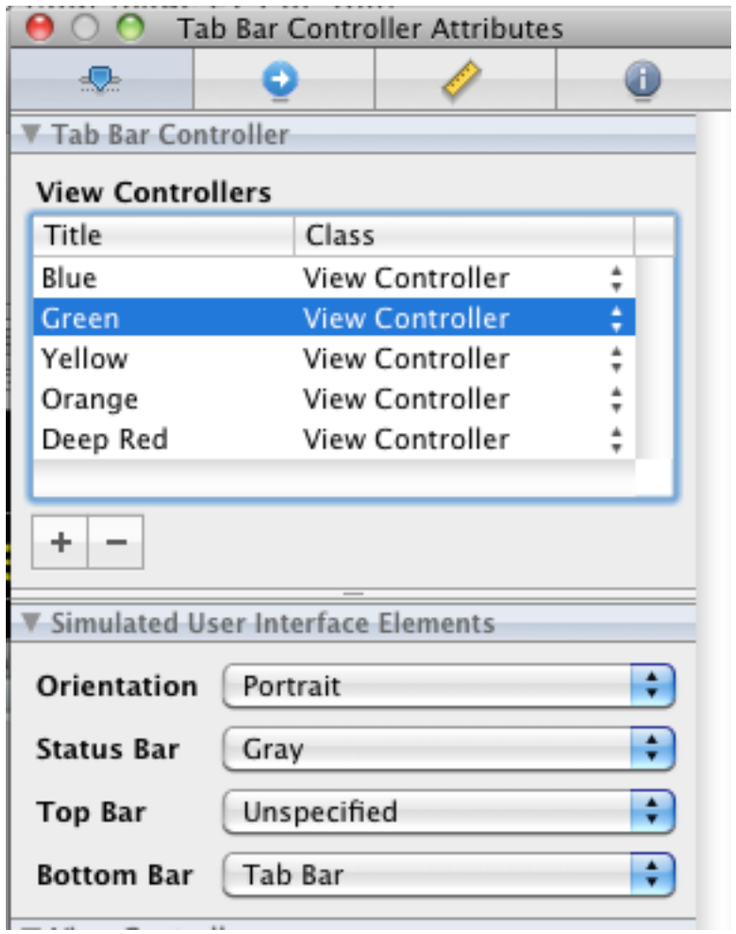
Linking Tab to XIB file





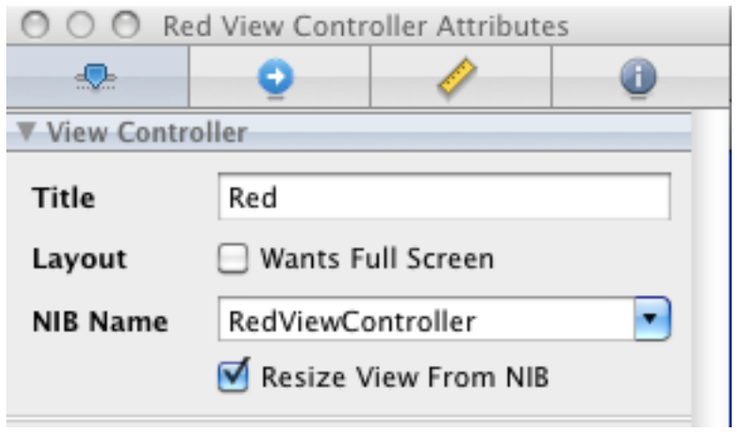


Three Inspector Views of Tabbar

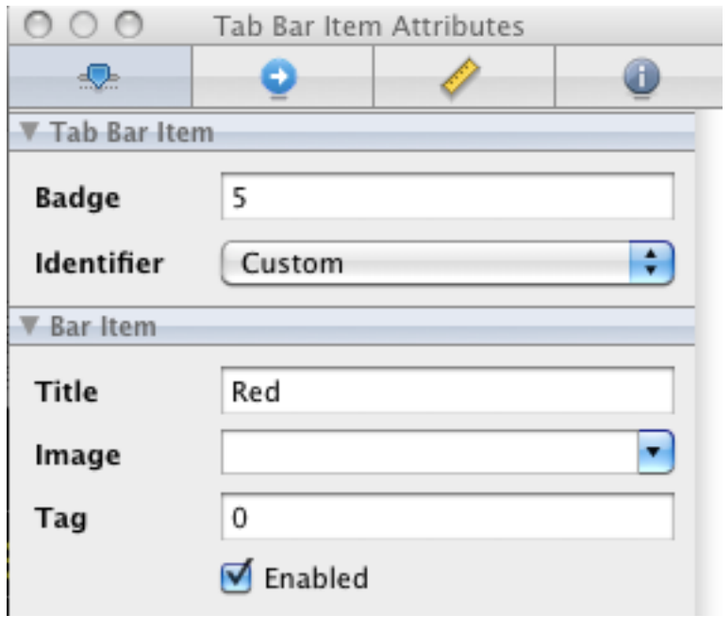


first click on tab bar

command (apple) click on tab item to deselect it

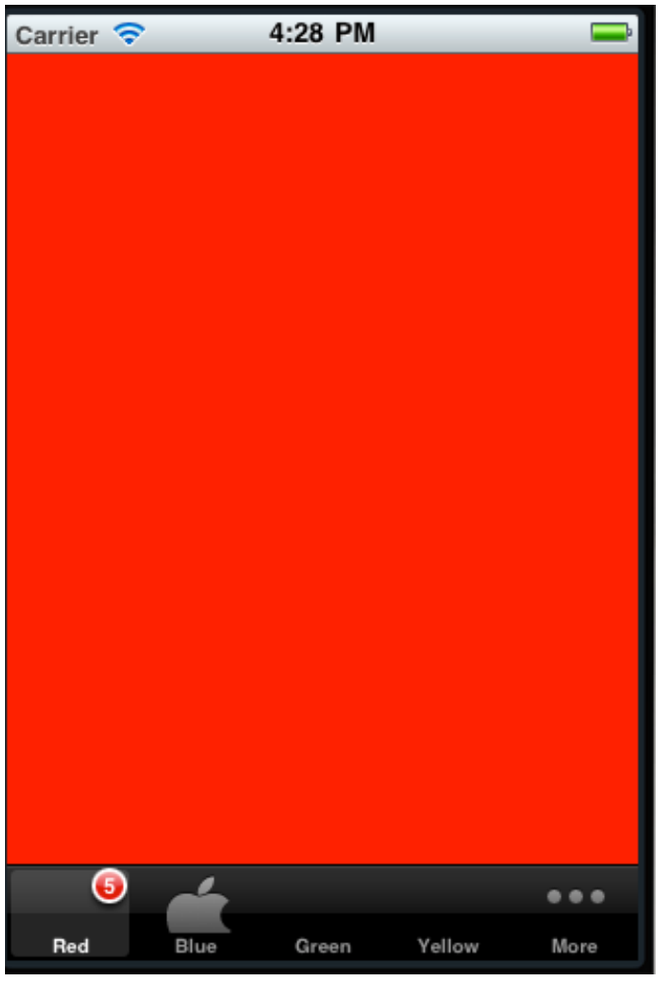


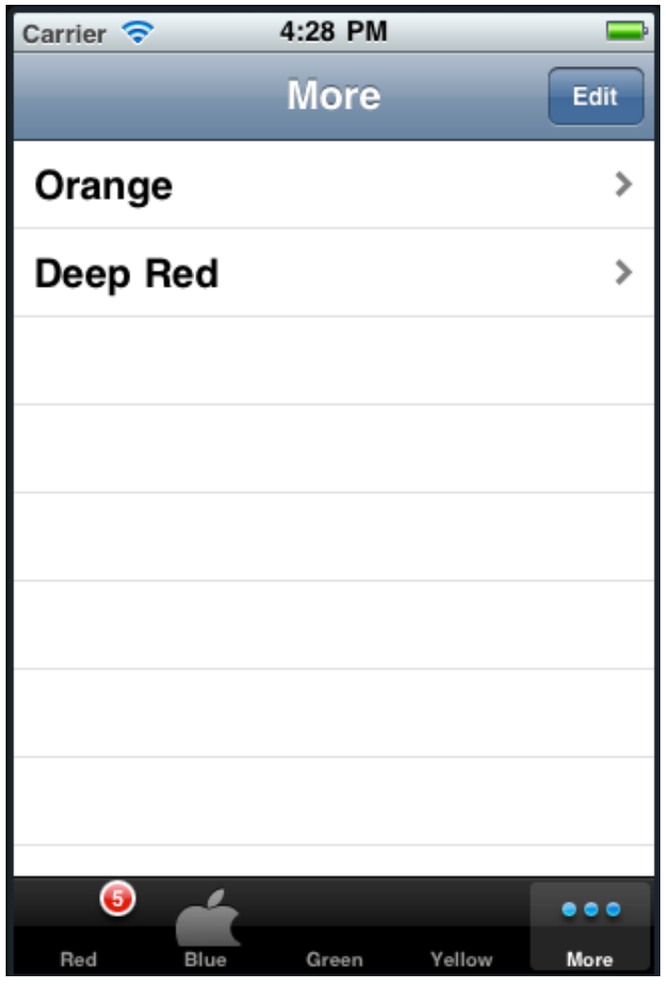
first click on tab item

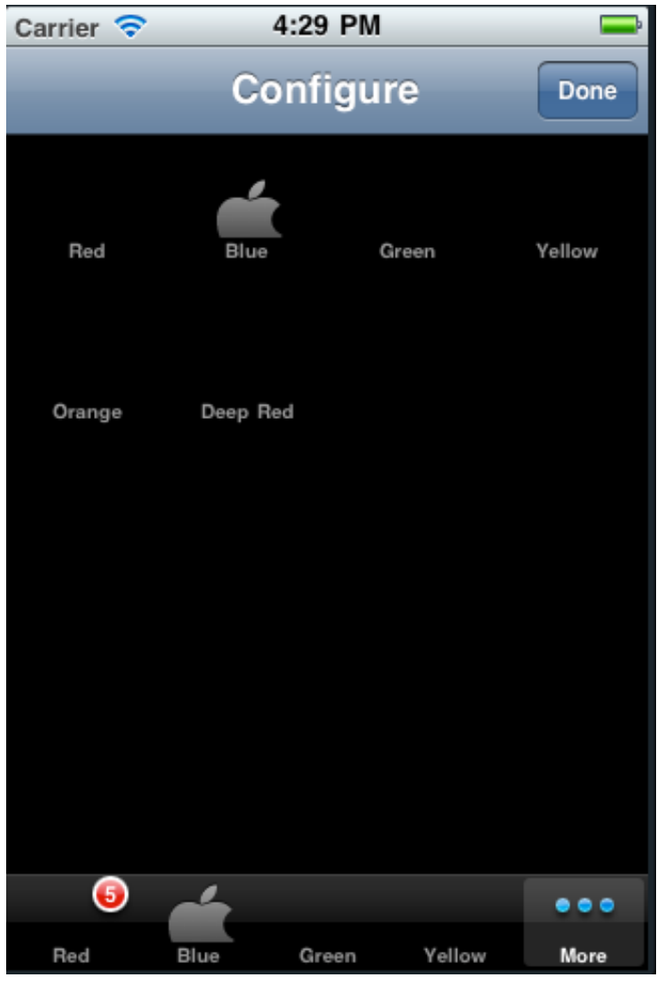


second click on tab item

More


Carrier
4:28 PM
5
Red
Blue
Green
Yellow
More


Carrier
4:28 PM
More
Edit
Orange
Deep Red
5
Red
Blue
Green
Yellow
More


Carrier
4:29 PM
Configure
Done
Red
Blue
Green
Yellow
Orange
Deep Red
5
Red
Blue
Green
Yellow
More

Classes & Protocols

UITabBar

View, contains UITabBarItems

UITabBarDelegate

Used to interact with users changing tab order

UITabBarController

Contains UITabBar, delegate, subview controllers

UITabBarControllerDelegate

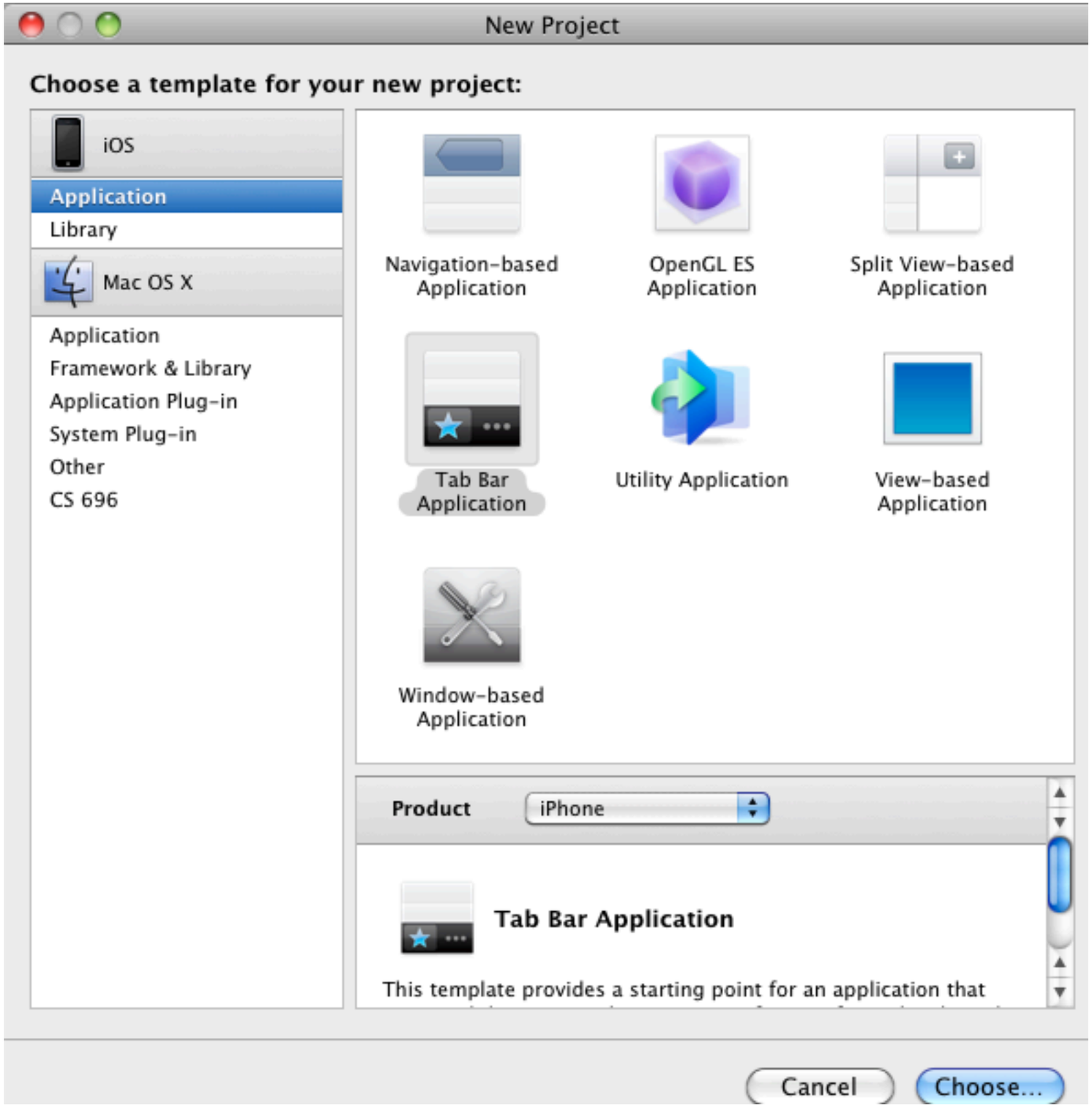
Selecting items

Starting, stopping editing of tab order

UITabBarItem

title, image, badge, tag

Tabbar & Xcode Templates



Textbook has you use Window-based app. Try doing it so you understand how main window xib file works

Interface Builder Blues

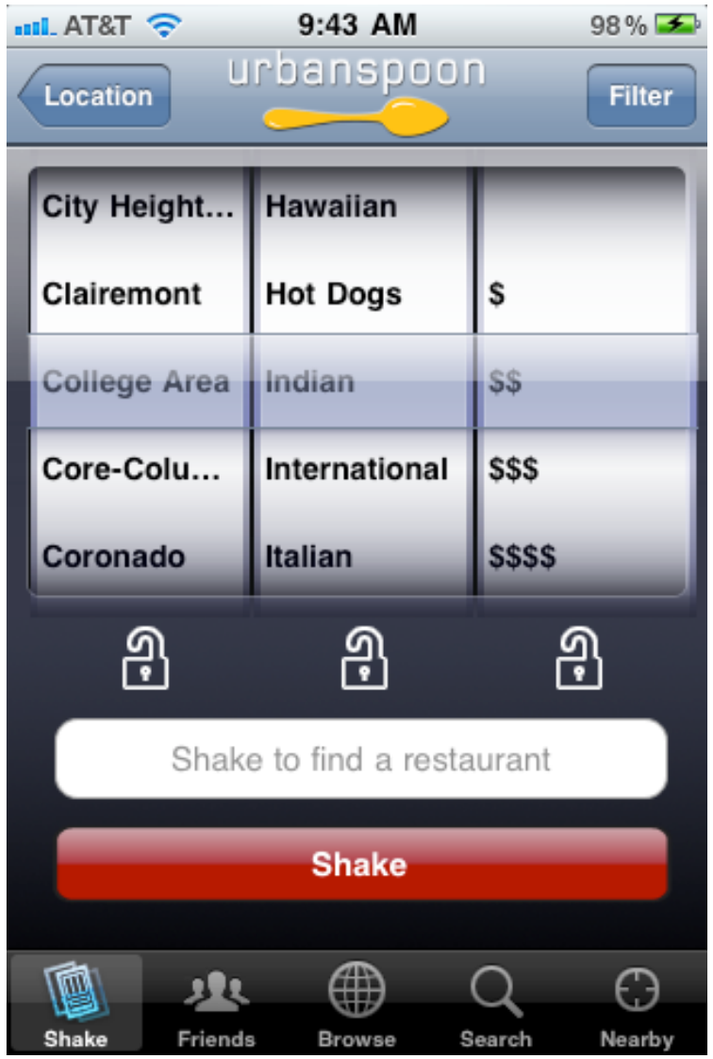
You will make a mistake in the interface builder

App will not work/crash

Error can be hard to find

You need to understand how the xib file works & what is needed

Tab bar + Navigation



Tab bar Creation in code

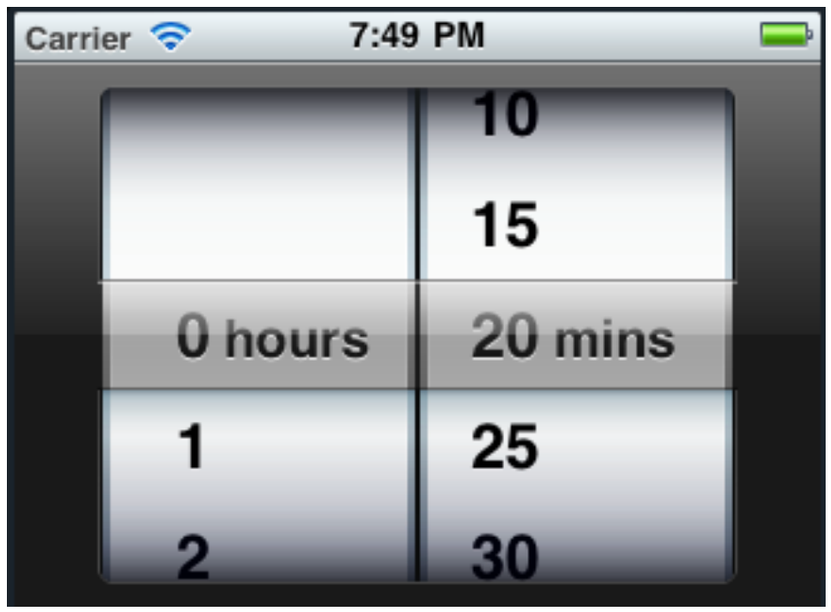
```
- (void) applicationDidFinishLaunching {
    tabBarController = [[UITabBarController alloc] init];

    tabBarController.viewControllers = arrayOfControllers;

    [window addSubview: tabBarController.view];
}
```

Each Controller has a tab bar item

```
- (void)viewDidLoad {
    [super viewDidLoad];
      UITabBarItem *item = [[UITabBarItem alloc] initWithTitle: @"Red" image:nil tag:0];
      self.tabBarItem = item;
      [item release];
```


Dont do this if you are using the interface builder to define tab bar

Pickers

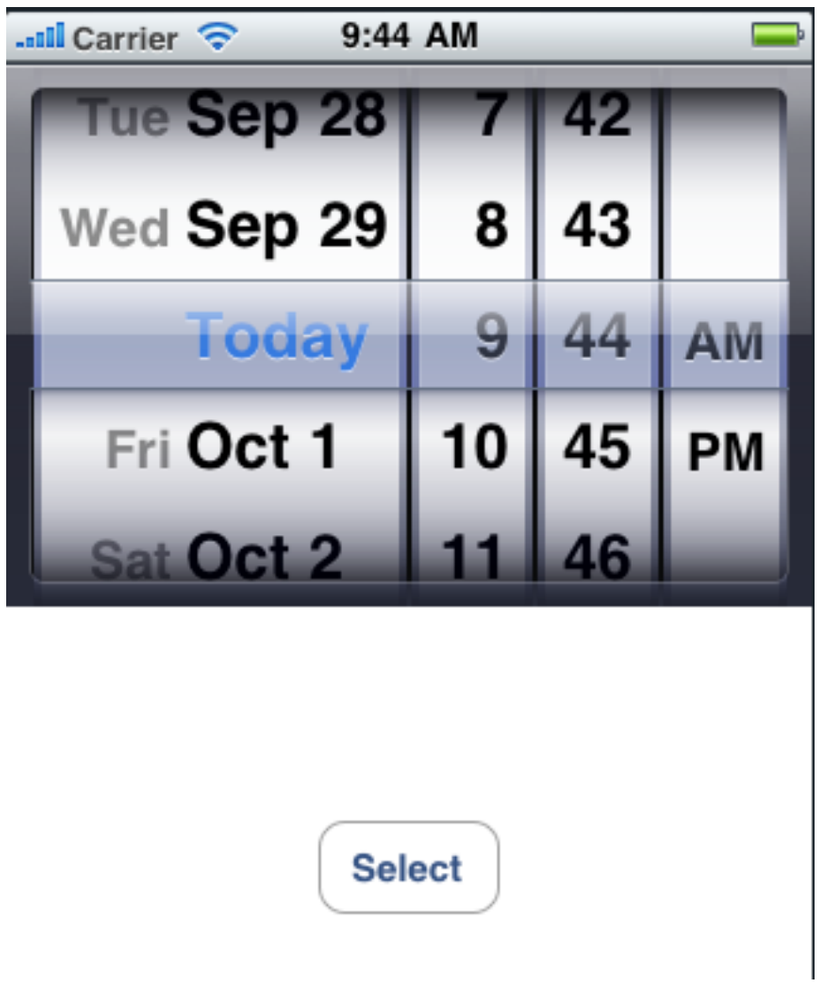


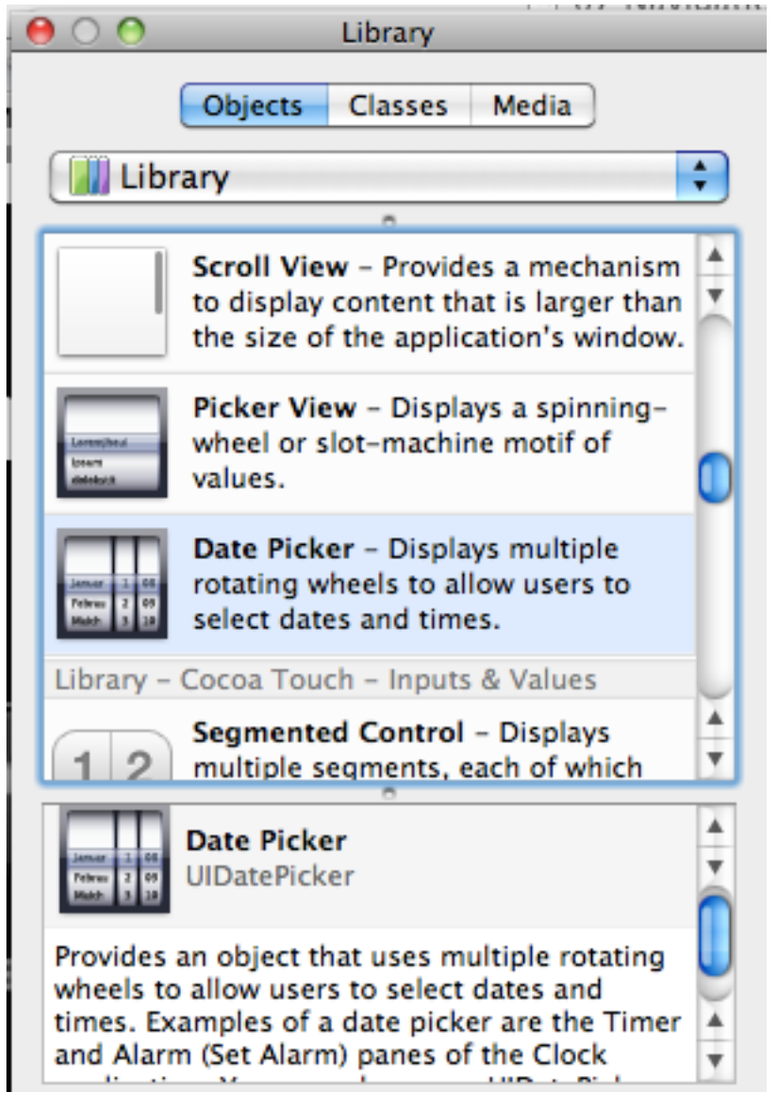
DatePicker

General Picker

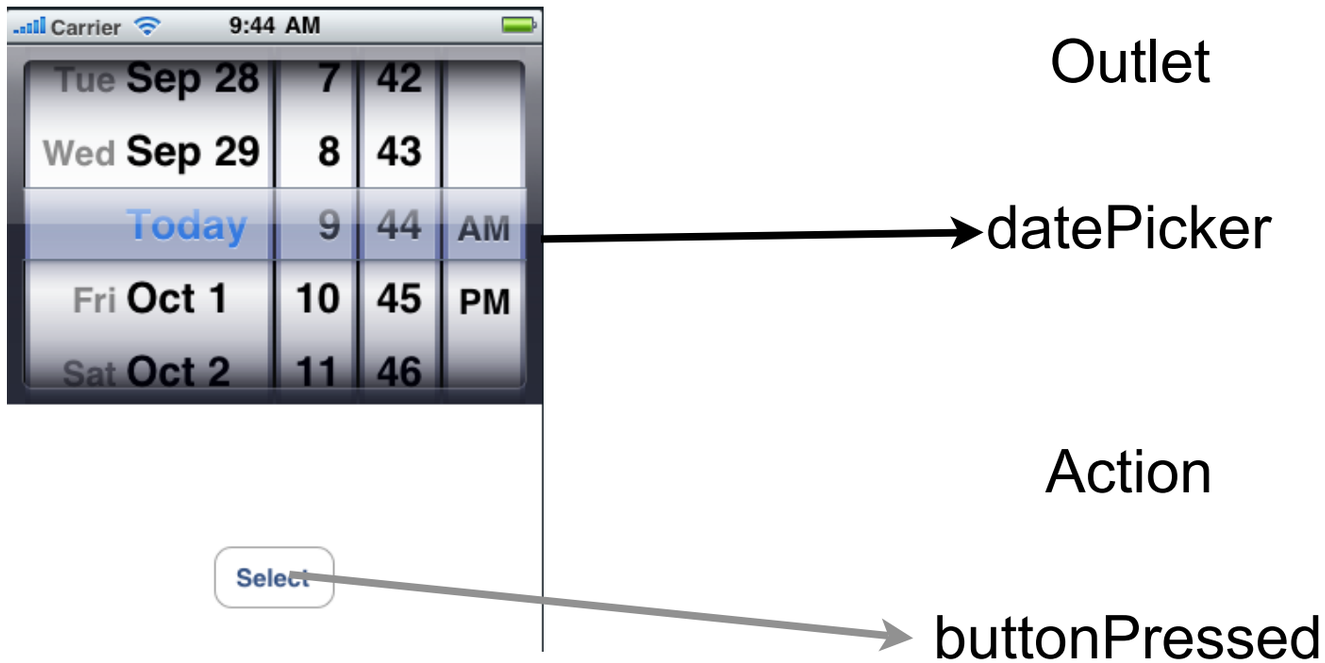
DatePicker

Simple





Connections




```
@interface DateViewController : UIViewController {
}
@property (nonatomic, retain) IBOutlet UIDatePicker *datePicker;
-(IBAction)buttonPressed;
@end
```

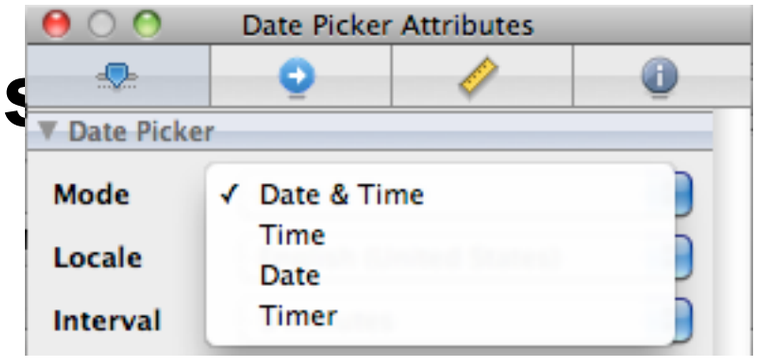
Setting the initial selection

```
- (void)viewDidLoad {
    [super viewDidLoad];
       NSDate *now = [[NSDate alloc] init];
       [datePicker setDate:now animated:NO];
       [now release];
}
```

Getting the selected date

```
- (IBAction) buttonPressed {
    NSDate *selected = [datePicker date];
    NSString *message = [[NSString alloc] initWithFormat:@"%@", selected];
    UIAlertView *alert = [[UIAlertView alloc]
                          initWithTitle:@"Date and Time Selected"
                          message:message
                          delegate:nil
                          cancelButtonTitle:@"OK"
                          otherButtonTitles:nil];
    [alert show];
    [alert release];
    [message release];
}
```

Display options

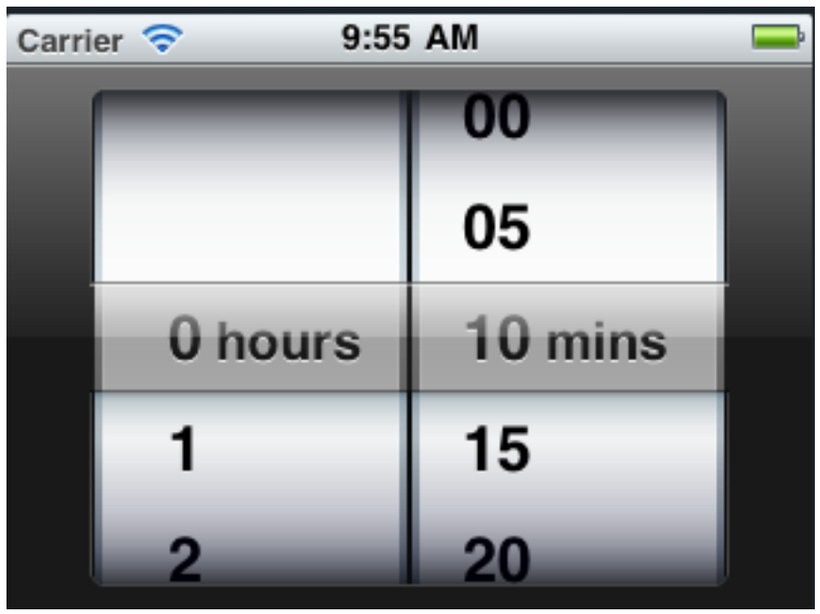


Date & Time

Time

Date

Timer



Date Picker Events

Value Changed is sent when user changes any dial

Picker

UIPickerView

Few Options

UIPickerViewDataSource

Number of columns

Number of rows per column

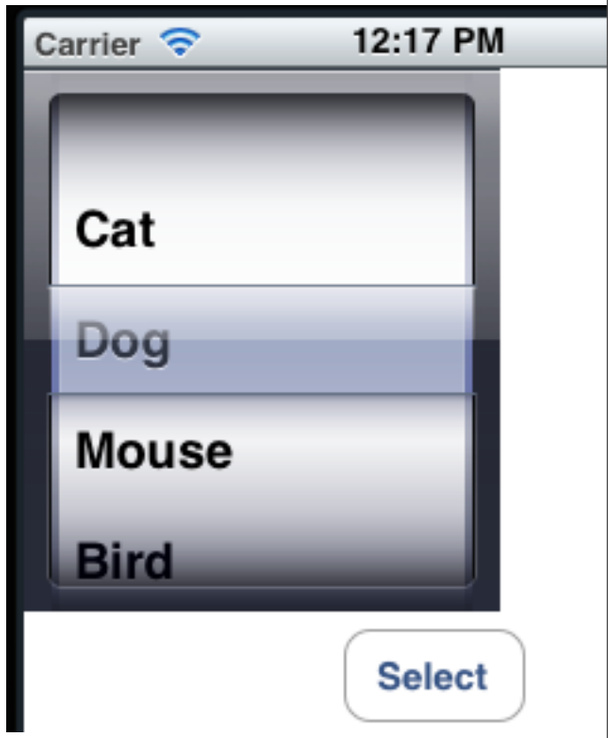
UIPickerViewDelegate

Sets the dimensions of columns & rows

Provides data to display

Responses to selection

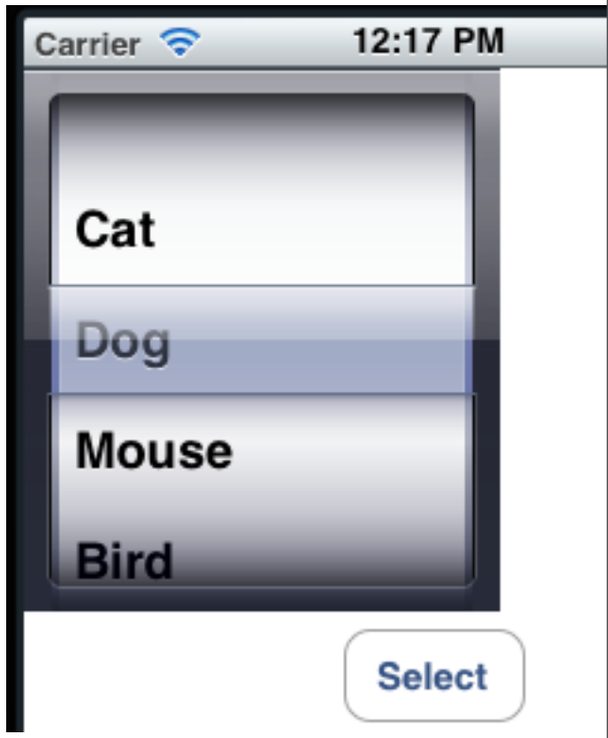
Single column Picker




```
@interface SingleComponentPicker : UIViewController
            <UIPickerViewDelegate, UIPickerViewDataSource> {
}

@property (nonatomic, retain) IBOutlet UIPickerView *singlePicker;

@property (nonatomic, retain) NSArray *pickerData;

- (IBAction)buttonPressed;
@end
```

Some Data

```
- (void)viewDidLoad {
    [super viewDidLoad];
      NSArray *array = [[NSArray alloc] initWithObjects:
          @"Cat", @"Dog", @"Mouse", @"Bird", @"Worm", nil];
      self.pickerData = array;
      [array release];
}
```



Used array for example as code is short,

All UIPickerViewDataSource methods

```
- (NSInteger) numberOfComponentsInPickerView: (UIPickerView *) pickerView {
      return 1;
}

- (NSInteger) pickerView: (UIPickerView *)
         pickerView numberOfRowsInComponent: (NSInteger)component {
      return [pickerData count];
}
```

Component = column = dial

Some UIPickerViewDelegate Methods

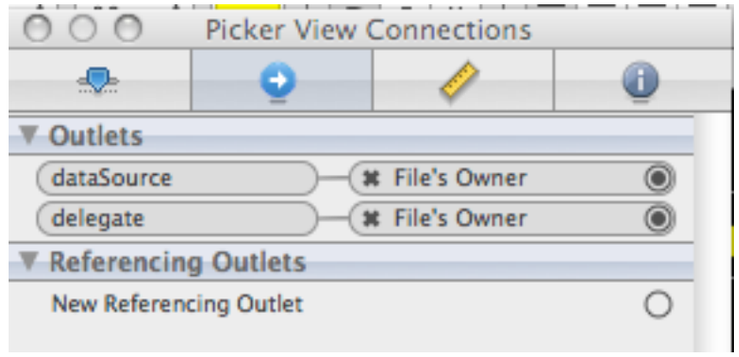
```
// Return the data
- (NSString *) pickerView: (UIPickerView *) pickerView
     titleForRow: (NSInteger) row
     forComponent: (NSInteger) component
{
      return [pickerData objectAtIndex:row];
}

//Handle change in dial (if really needed)
- (void) pickerView: (UIPickerView *) pickerView
     didSelectRow:(NSInteger) row
     inComponent: (NSInteger) component {
      NSLog(@"Column:%i Row: %i", component, row);
}
```

Connecting the Picker to source and delagete

In code

```
- (void)viewDidLoad {
    [super viewDidLoad];
    NSArray *array = [[NSArray alloc] initWithObjects:
        @"Cat", @"Dog", @"Mouse", @"Bird", @"Worm", nil];
    self.pickerData = array;
    self.singlePicker.delegate = self;
    self.singlePicker.dataSource = self;
    [array release];
}
```

In Interface Buidler



Don't need to do both

Dependent Components

Data file - food.plist

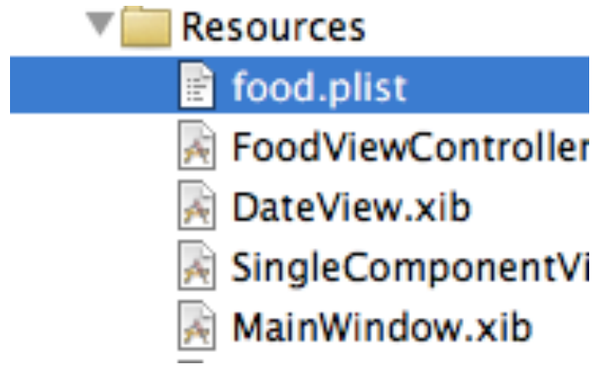
Key	Type	Value
▼ Root	Dictionary ▲▼	(2 items)
▼ Vegetarian	Array	(4 items)
Item 0	String	Mango
Item 1	String	Papaya
Item 2	String	Jackfruit
Item 3	String	Durian
▼ Carnivore	Array	(4 items)
Item 0	String	Chicken
Item 1	String	Pork
Item 2	String	Beef
Item 3	String	Fish

Creating plist

plist editor

Code

Reading food.plist




```
NSBundle *bundle = [NSBundle mainBundle];
NSString *plistPath = [bundle pathForResource:@"food" ofType:@"plist"];
NSDictionary *dictionary = [[NSDictionary alloc] initWithContentsOfFile:plistPath];
```

Some Controller instance variables

NSDictionary *typesAndFood

Dictionary read from plist file

NSArray *foodTypes

Sorted list of keys from typesAndFood

NSArray *food

Food for current selected food type

Initializing Controller

```
- (void)viewDidLoad {
    [super viewDidLoad];
      NSBundle *bundle = [NSBundle mainBundle];
      NSString *plistPath = [bundle pathForResource:@"food" ofType:@"plist"];
      NSDictionary *dictionary = [[NSDictionary alloc] initWithContentsOfFile:plistPath];
      self.typesAndFood = dictionary;
      [dictionary release];

      NSArray *components = [self.typesAndFood allKeys];
      NSArray *sorted = [components sortedArrayUsingSelector:@selector(compare:)];
      self.foodTypes = sorted;

      NSString *selectType = [self.foodTypes objectAtIndex:0];
      self.food = [typesAndFood objectForKey:selectType];

}
```

Number of row & columns

```
#define kFoodTypeComponent 0
#define kFoodComponent 1



 - (NSInteger) numberOfComponentsInPickerView: (UIPickerView *) pickerView
 {
        return 2;
 }
```

```
- (NSInteger) pickerView: (UIPickerView *) pickerView
     numberOfRowsInComponent: (NSInteger) component
{
       if (component == kFoodTypeComponent)
             return [self.foodTypes count];
       return [self.food count];
}
```

Returning the data for each column

```
- (NSString *)pickerView:(UIPickerView *)pickerView
     titleForRow:(NSInteger)row
     forComponent:(NSInteger)component
{
      if (component == kFoodTypeComponent)
            return [self.foodTypes objectAtIndex:row];
      return [self.food objectAtIndex:row];
}
```

Updating View when user selects Food type

```
- (void) pickerView: (UIPickerView *) pickerView
     didSelectRow: (NSInteger) row
     inComponent: (NSInteger) component
{
      if (component == kFoodTypeComponent)
            {
            NSString *selectedType = [self.foodTypes objectAtIndex:row];
            NSArray *array = [typesAndFood objectForKey:selectedType];
            self.food = array;
            [foodPicker selectRow:0 inComponent:kFoodComponent animated:YES];
            [foodPicker reloadComponent:kFoodComponent];
            }
}
```