Announcements

- Lab 3 is due Feb 23rd

Topics

- NSUserDefaults and Screen Rotation
- View Controllers
- Application Data Flow
- Customizing Navigation
- Tab Bar Controllers
- Combining Approaches

Loading and Saving Data

- **Lots of options out there, depends on what you need**
  - NSUserDefaults
  - Property lists
  - SQLite
  - Web services
- **Covering in greater depth in a few weeks**

Saving State Across App Launches

- **NSUserDefaults to read and write prefs & state**
- **Singleton object:**
  + (NSUserDefaults *)standardUserDefaults;
- **Methods for storing & fetching common types:**
  -(NSInteger)integerForKey:(NSString *)key;
  -(void)setInteger:(int)value forKey:(NSString *)key;

  -(id)objectForKey:(NSString *)key;
  -(void)setObject:(int)value forKey:(NSString *)key;•
- **Find an appropriate time to store and restore your state**

More View Controller Hooks

- Automatically rotating your user interface
- Low memory warnings

Supporting Interface Rotation

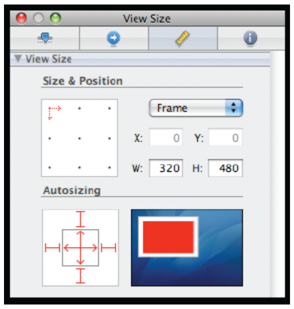
```
-(BOOL)shouldAutorotateToInterfaceOrientation:
              (UIInterfaceOrientation)interfaceOrientation
{

  // This view controller only supports portraits
     return (interfaceOrientation == UIInterfaceOrientationPortrait);

}
```

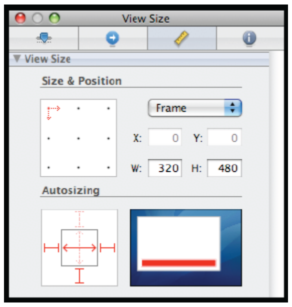
Supporting Interface Rotation

```
-(BOOL)shouldAutorotateToInterfaceOrientation:
            (UIInterfaceOrientation)interfaceOrientation
{

  // This view controller supports all orientations
  // except for upside-down.
    return (interfaceOrientation !=UIInterfaceOrientationPortraitUpsideDown);

}
```

Autoresizing Your Views

```
view.autoresizingMask = UIViewAutoresizingFlexibleWidth |
                        UIViewAutoresizingFlexibleHeight;


view.autoresizingMask = UIViewAutoresizingFlexibleWidth |
                        UIViewAutoresizingFlexibleTopMargin;
```





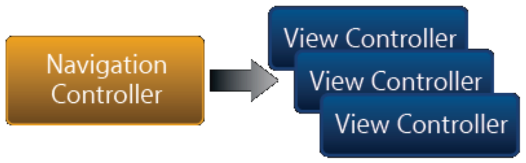
NSUserDefaults and Screen Orientation Demo

UIViewController

- Basic building block
- Manages a screenful of content
- Subclass to add your application logic

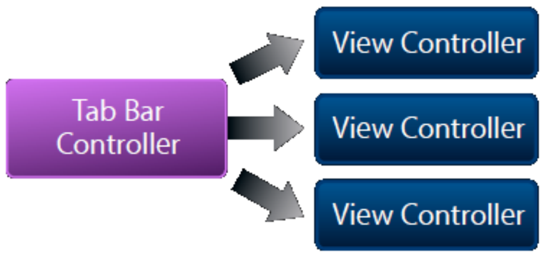
“Your” and Apple View Controllers

- Create your own UIViewController subclass for each screenful
- Plug them together using existing composite view controllers



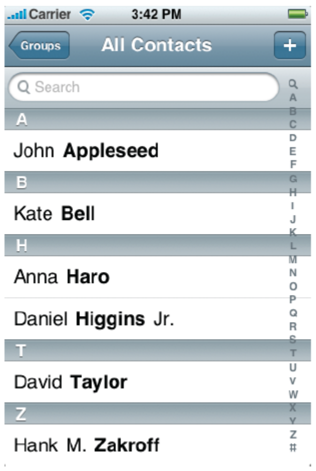
“Your” and “Our” View Controllers

- Create your own UIViewController subclass for each screenful
- Plug them together using existing composite view controllers

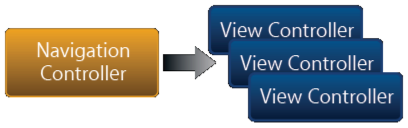


Navigation Controllers

UINavigationController

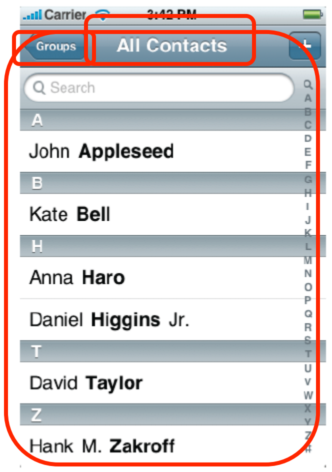


- Stack of view controllers
- Navigation bar



How It Fits Together

- Top view controller's view
- Top view controller's title
- Previous view controller's title



Modifying the Navigation Stack

- Push to add a view controller

```
-(void)pushViewController:(UIViewController *)viewController
        animated:(BOOL)animated;
```

- Pop to remove a view controller

```
-(UIViewController *)popViewControllerAnimated:(BOOL)animated;
```

Pushing Your First View Controller

```
- (void)applicationDidFinishLaunching {
// Create a navigation controller
navController = [[UINavigationController alloc] init];


// Push the first view controller on the stack
[navController pushViewController:firstViewController
animated:NO];

// Add the navigation controller's view to the window
[window addSubview:navController.view];
}
```

In Response to User Actions

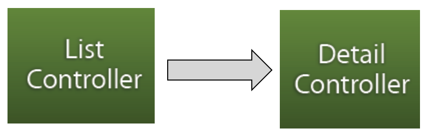
- Push from within a view controller on the stack

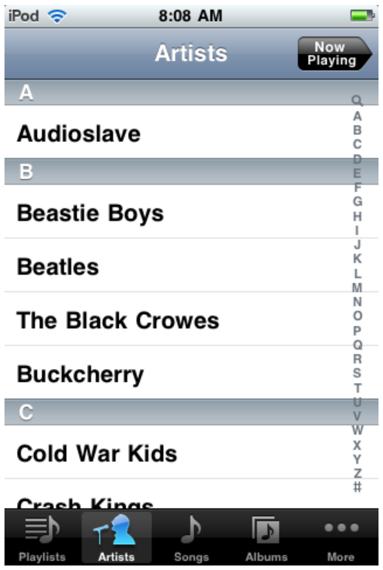
```
- (void)someAction:(id)sender
{
// Potentially create another view controller
UIViewController *viewController = ...;

[self.navigationController pushViewController:viewController
animated:YES];

}
```

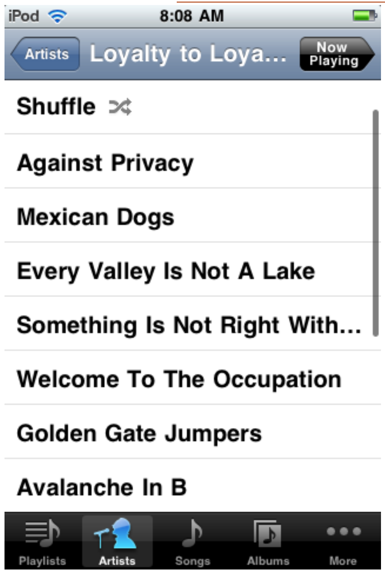
- Almost never call pop directly!
  - Automatically invoked by the back button

Application Data Flow

A Controller for Each Screen





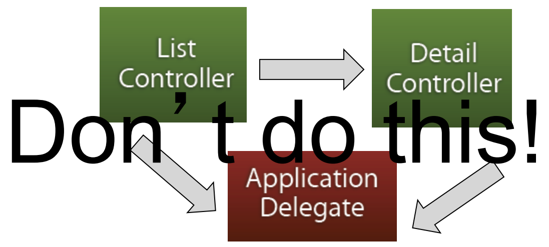


Connecting View Controllers

- Multiple view controllers may need to share data
- One may need to know about what another is doing
  - Watch for added, removed or edited data
  - Other interesting events

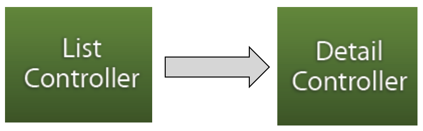
How Not To Share Data

- **Global variables or singletons**
  - This includes your application delegate!
- **Direct dependencies make your code less reusable**
  - And more difficult to debug & test



Best Practices for Data Flow

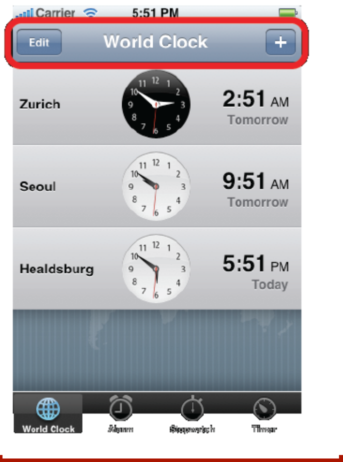
- **Figure out exactly what needs to be communicated**
- **Define input parameters for your view controller**
- **For communicating back up the hierarchy, use loose coupling**
  - Define a generic interface for observers (like delegation)

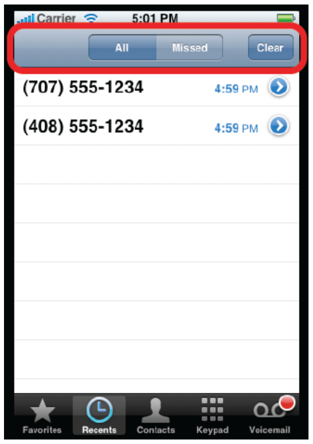


Customizing Navigation

Customizing Navigation

- Buttons or custom controls
- Interact with the entire screen

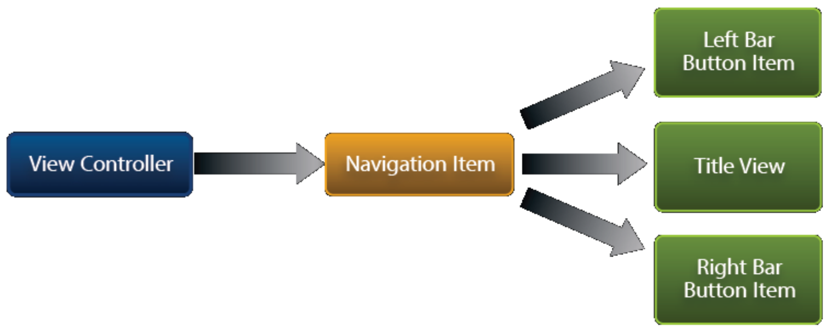




UINavigationItem

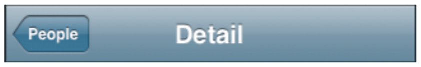
- **Describes appearance of the navigation bar**
  - Title string or custom title view
  - Left & right bar buttons
  - More properties defined in UINavigationBar.h
- **Every view controller has a navigation item for customizing**
  - Displayed when view controller is on top of the stack

Navigation Item Ownership



Displaying a Title

- **UIViewController already has a title property**
  - @property(nonatomic,copy) NSString *title;
- **Navigation item inherits automatically**
  - Previous view controller's title is displayed in back button

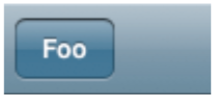


viewController.title = @"Detail";

Left & Right Buttons

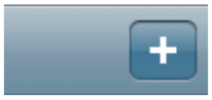
- **UIBarButtonItem**
  - Special object, defines appearance & behavior for items in navigation bars and toolbars
- **Display a string, image or predefine system item**
- **Target + action (like a regular button)**

Text Bar Button Item



```
- (void)viewDidLoad
{
  UIBarButtonItem *fooButton = [[UIBarButtonItem alloc]
      initWithTitle:@"Foo”
      style:UIBarButtonItemStyleBordered
      target:self
      action:@selector(foo:)];

   self.navigationItem.leftBarButtonItem = fooButton;

   [fooButton release];
}
```

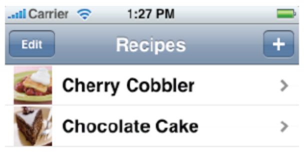
System Bar Button Item



```
(void)viewDidLoad
{
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc]
    initWithBarButtonSystemItem:UIBarButtonSystemItemAdd
    style:UIBarButtonItemStyleBordered
    target:self
    action:@selector(add:)];
    self.navigationItem.rightBarButtonItem = addButton;
    [addButton release];
}
```

Navigation Controller with Custom Button Demo

Edit/Done Button



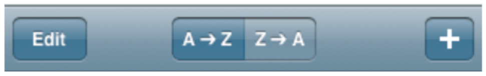
- Very common pattern
- Every view controller has one available
  - Target/action already set up
- Edit/Done Button

```
self.navigationItem.leftBarButtonItem = self.editButtonItem;
```

// Called when the user toggles the edit/done button

```
- (void)setEditing:(BOOL)editing animated:(BOOL)animated
{
   // Update appearance of views
}
```

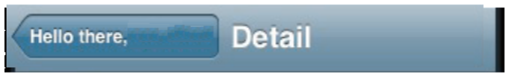
Custom Title View



- Arbitrary view in place of the title

```
UISegmentedControl *segmentedControl = ...
self.navigationItem.titleView = segmentedControl;
[segmentedControl release];
```

Back Button



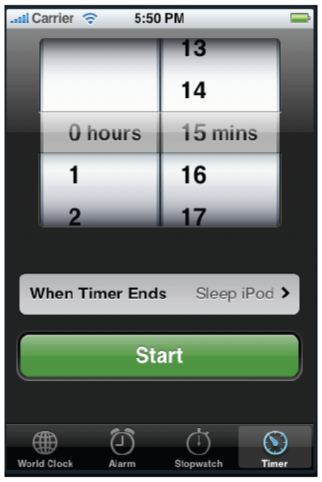
- Sometimes a shorter back button is needed

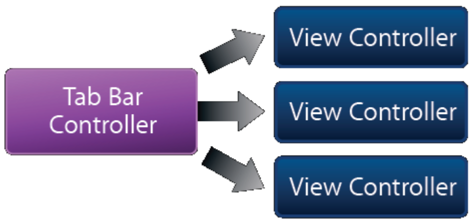
```
Self.title = @"Hello there,";
UIBarButtonItem *heyButton = [[UIBarButtonItem alloc]
initWithTitle:@"Hey!" ...];
self.navigationItem.backBarButtonItem = heyButton;
[heyButton release];
```

Tab Bar Controllers

UITabBarController

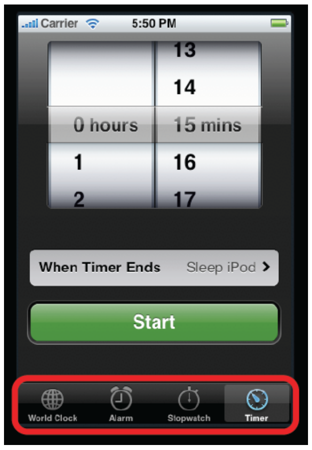
Array of view controllers





How it Fits Together

- Selected view controller's view
- All view controller's titles

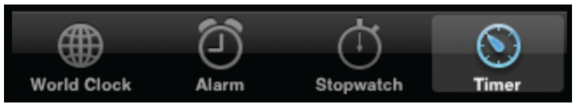


Setting Up a Tab Bar Controller

```
(void)applicationDidFinishLaunching
// Create a tab bar controller
tabBarController = [[UITabBarController alloc] init];


// Set the array of view controllers
tabBarController.viewControllers = [NSArray arrayWithObjects:
    firstVC, secondVC, nil];

// Add the tab bar controller's view to the window
[window addSubview:tabBarController.view];

}
```

Tab Bar Appearance

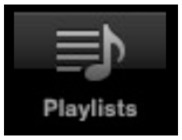
- **View controllers can define their appearance in the tab bar**
- **UITabBarItem**
  - Title + image or system item
- **Each view controller comes with a tab bar item for customizing**



Tab Bar Controller Demo

Creating Tab Bar Items

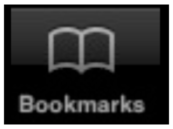
- Title and image



```
- (void)viewDidLoad
{
   self.tabBarItem = [[UITabBarItem alloc]
   initWithTitle:@“Playlists”
   image:[UIImage imageNamed:@“music.png”]
   tag:0]
}
```

Creating Tab Bar Items

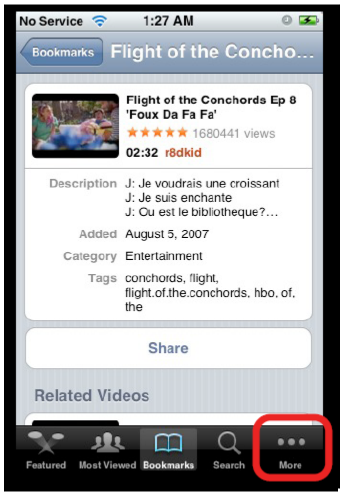
- System item



```
- (void)viewDidLoad
{
   self.tabBarItem = [[UITabBarItem alloc]
   initWithTabBarSystemItem:
     UITabBarSystemItemBookmarks tag:0]
}
```

More View Controllers

- **What happens when a tab bar controller has too many view controllers to display at once?**
  - "More" tab bar item displayed automatically
  - User can navigate to remaining view controllers

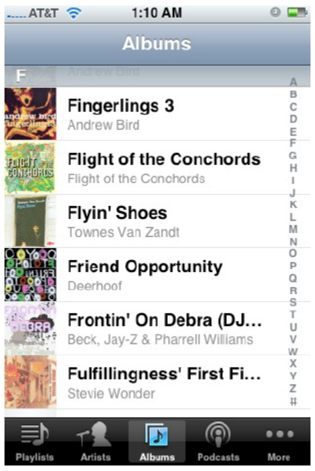


Flight of the Conchords Ep 8 'Foux Da Fa Fa'
★★★★★ 1680441 views
02:32 r8dkid

Description	J: Je voudrais une croissant J: Je suis enchante J: Ou est le bibliotheque?...
Added	August 5, 2007
Category	Entertainment
Tags	conchords, flight, flight.of.the.conchords, hbo, of, the

Combining Approaches

Tab Bar + Navigation Controllers


AT&T 1:10 AM

Albums

F

Fingerlings 3
Andrew Bird

Flight of the Conchords
Flight of the Conchords

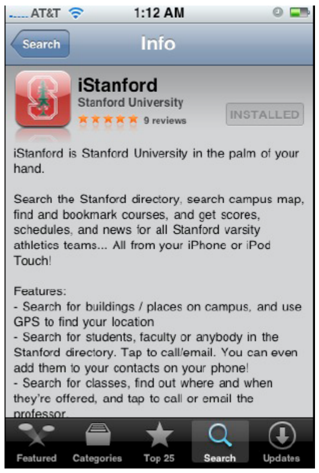
Flyin' Shoes
Townes Van Zandt

Friend Opportunity
Deerhoof

Frontin' On Debra (DJ...
Beck, Jay-Z & Pharrell Williams

Fulfillingness' First Fi...
Stevie Wonder

Playlists Artists Albums Podcasts More


AT&T 1:12 AM

Search Info

iStanford
Stanford University
★★★★★ 9 reviews
INSTALLED

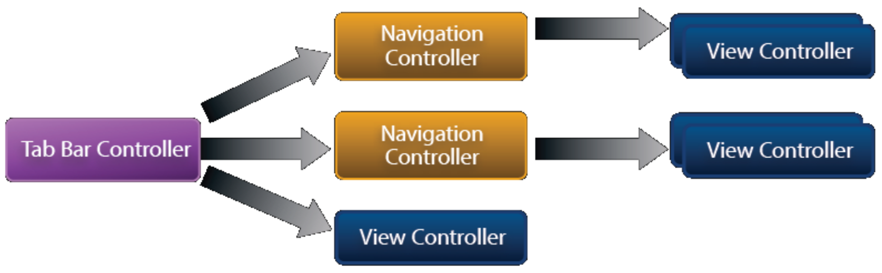
iStanford is Stanford University in the palm of your hand.

Search the Stanford directory, search campus map, find and bookmark courses, and get scores, schedules, and news for all Stanford varsity athletics teams... All from your iPhone or iPod Touch!

Features:
- Search for buildings / places on campus, and use GPS to find your location
- Search for students, faculty or anybody in the Stanford directory. Tap to call/email. You can even add them to your contacts on your phone!
- Search for classes, find out where and when they're offered, and tap to call or email the professor

Featured Categories Top 25 Search Updates

Tab Bar + Navigation Controllers



Nesting Navigation Controllers

- Create a tab bar controller

```
tabBarController = [[UITabBarController alloc] init];
```

- Create each navigation controller

```
navController = [[UINavigationController alloc] init];
  [navController pushViewController:firstViewController
                                   animated:NO];
```

- Add them to the tab bar controller

```
tabBarController.viewControllers = [NSArray arrayWithObjects:
                                         navController,
                                         anotherNavController,
                                         someViewController,
                                         nil];
```

Lab 4 Demo

Final Project Example