

CS193P - Lecture 7

iPhone Application Development

Navigation & Tab Bar Controllers

ナビゲーションコントローラ と タブバーコントローラ

Announcements

- Assignment 3 is due tomorrow
- Paparazzi 1 is due on Wednesday February 3rd

Today's Topics

今日のトピックス

- Navigation Controllers
 - Application Data Flow
- Customizing Navigation
- Tab Bar Controllers
- Combining Approaches

ナビゲーションコントローラ
アプリケーションのデータの流れ

ナビゲーションコントローラのカスタム化

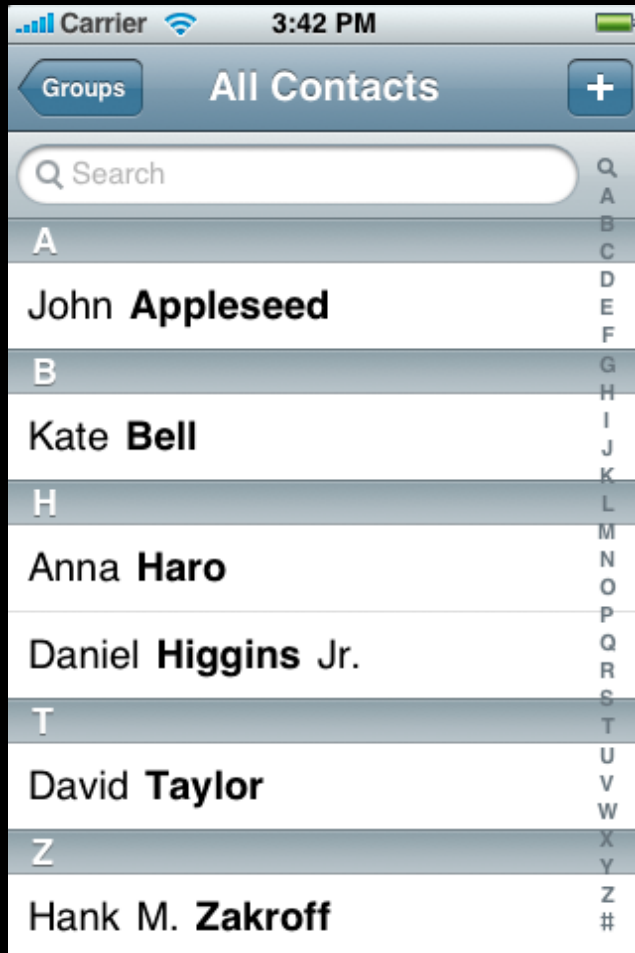
タブバーコントローラ

これらを複合させるアプローチ

Navigation Controllers

ナビゲーションコントローラ

UINavigationController

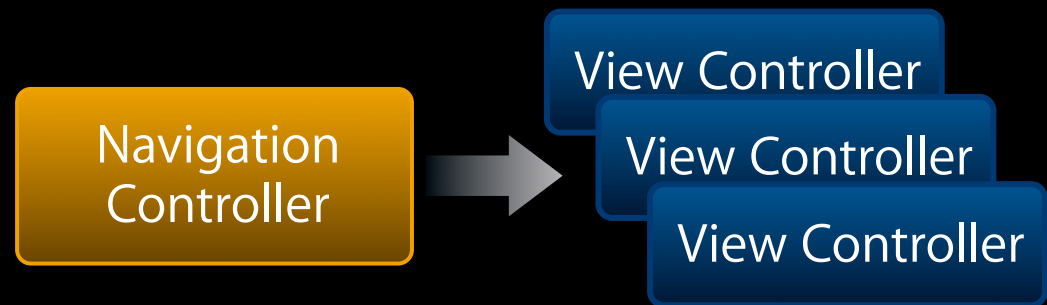
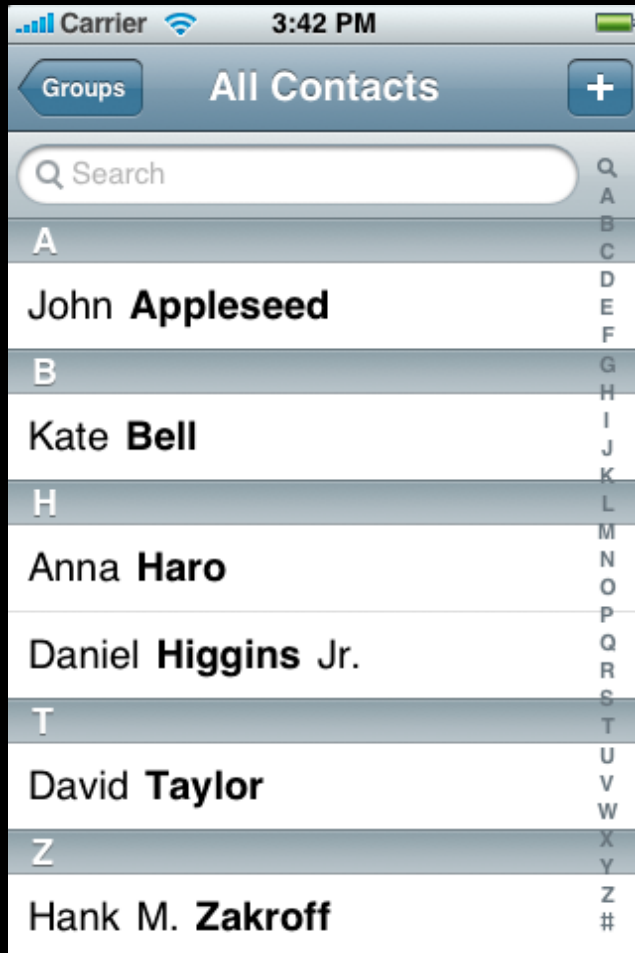


- Stack of view controllers
 - Navigation bar
ナビゲーションバー
- 重層化された
View Controller

Navigation
Controller

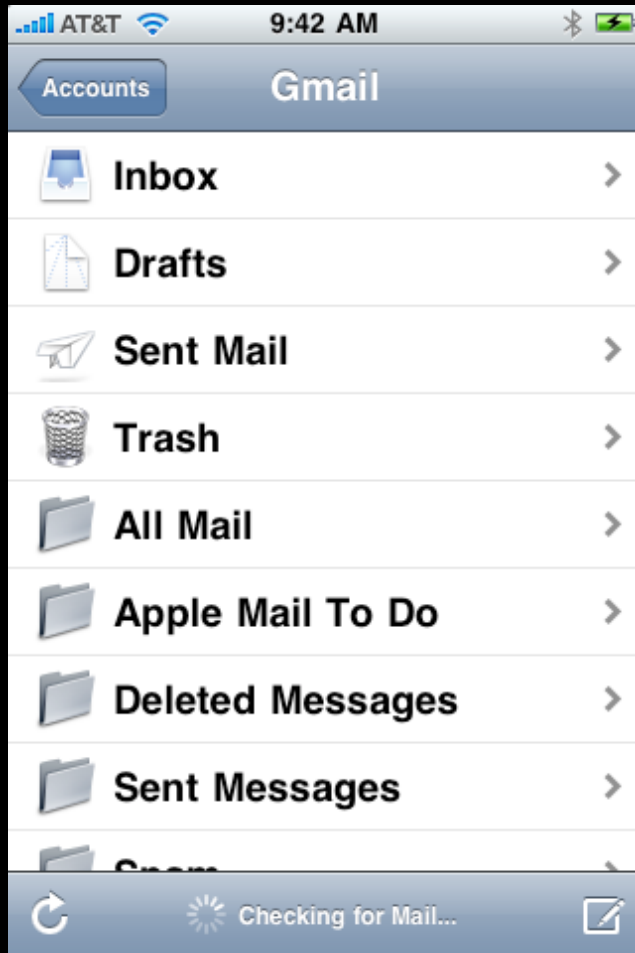
UINavigationController

- Stack of view controllers
- Navigation bar



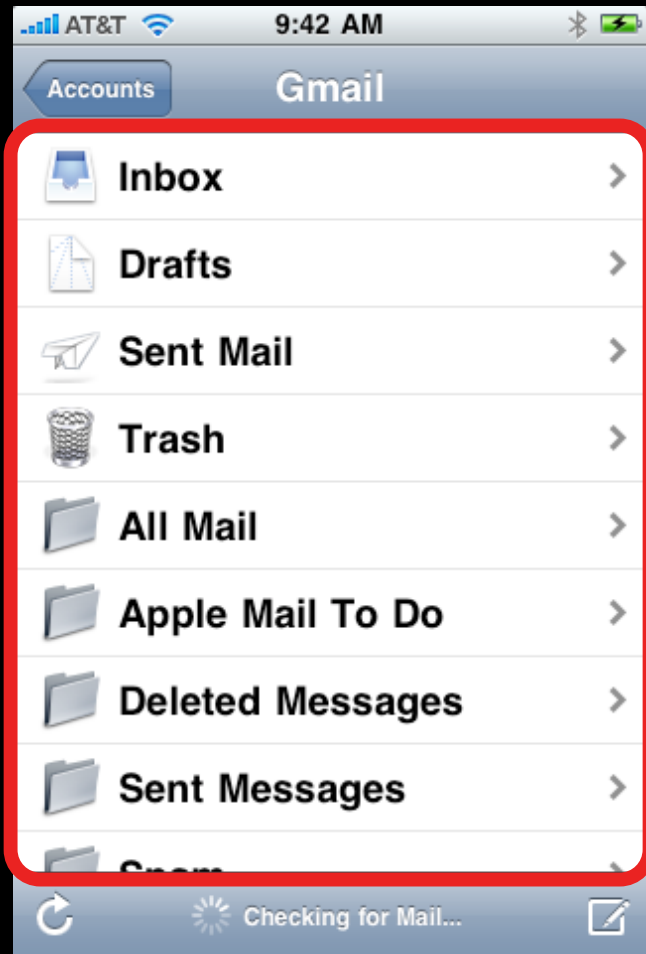
How It Fits Together

これら要素がどのように合体するのか

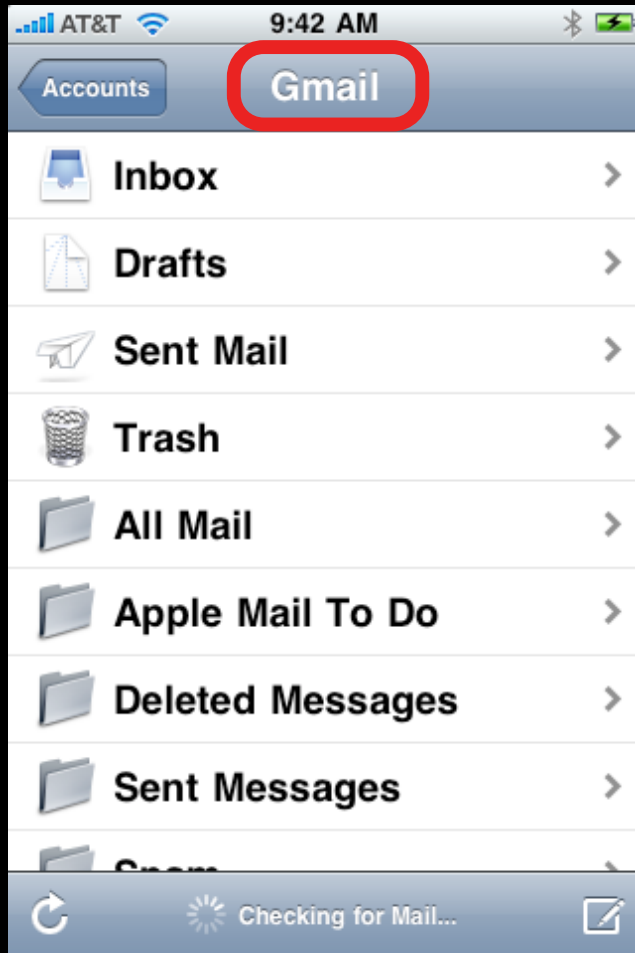


How It Fits Together

- Top view controller's view
最前面にある View Controller の View



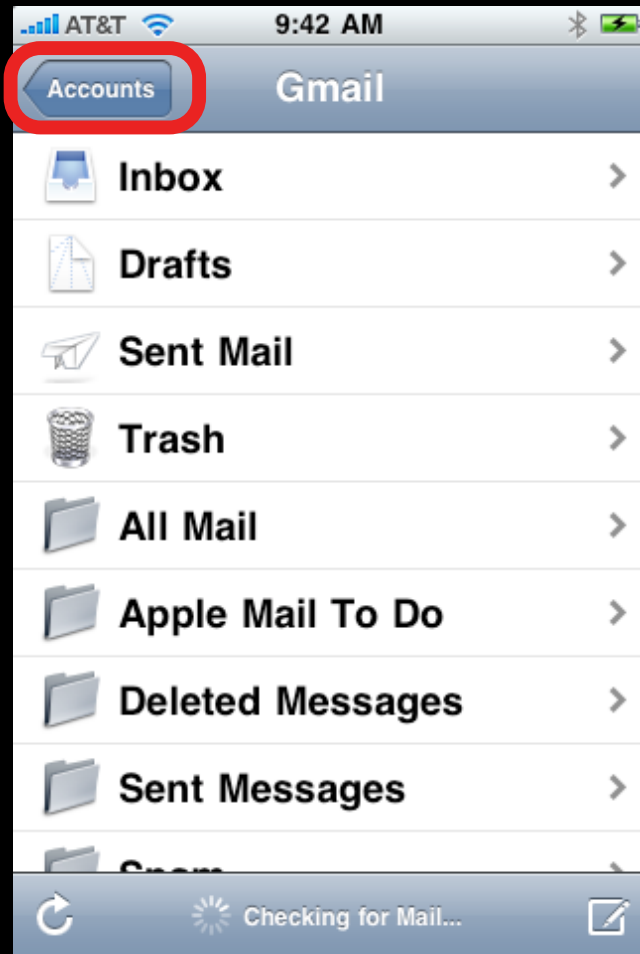
How It Fits Together



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

最前面にある View Controller の
タイトル

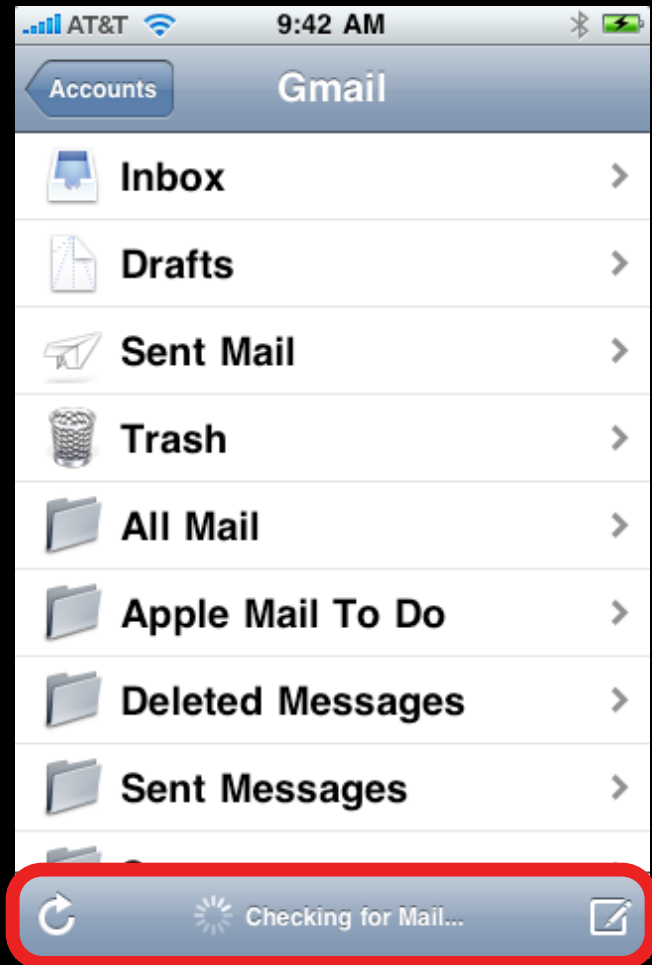
How It Fits Together



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

ひとつ前の View Controller の
タイトル (「戻る」ボタン)

How It Fits Together



- Top view controller's view
- Top view controller's title
- Previous view controller's title
- Top view controller's toolbar items (iPhone OS 3.0)

最前面にある View Controller の
ツールバー

Modifying the Navigation Stack

ナビゲーション
スタックの操作

- **Push** to add a view controller **View Controller を加える**
 - (void)**pushViewController**:(UIViewController *)viewController
animated:(BOOL)animated;
- **Pop** to remove a view controller **View Controller をはずす**
 - (UIViewController *)**popViewControllerAnimated**:(BOOL)animated;
- **Set** to change the entire stack of view controllers
(iPhone OS 3.0) **View Controller のスタック (積み上げ) 全体を別のスタックと入れ替える**
 - (void)**setViewControllers**:(NSArray *)viewControllers
animated:(BOOL)animated;

最初の View Controller を Push するには

Pushing Your First View Controller

アプリ起動直後に呼び出されるメソッド

- (void)applicationDidFinishLaunching

// Create a navigation controller Navigation Controller を生成

```
navController = [[UINavigationController alloc] init];
```

// Push the first view controller on the stack 最初の VC を
スタックに Push
[navController pushViewController:firstViewController
animated:NO];

// Add the navigation controller's view to the window

```
[window addSubview:navController.view]; 大元の Window に
```

```
} Navigation Controller をセット
```

In Response to User Actions

ユーザのアクションに
応答するには

- Push from within a view controller on the stack **スタック上の VC から
さらに奥側に Push する**
 - (void)someAction:(id)sender
{
 // Potentially create another view controller **必要に応じて
新しい VC を生成**
 UIViewController *viewController = ...;

 [self.navigationController pushViewController:viewcontroller
 自分の属する Navigation Controller animated:YES];
}
- Almost **never** call pop directly! **直接 Pop を呼ぶことはない**
 - Automatically invoked by the back button **「戻る」ボタンで
自動的に呼び出される**

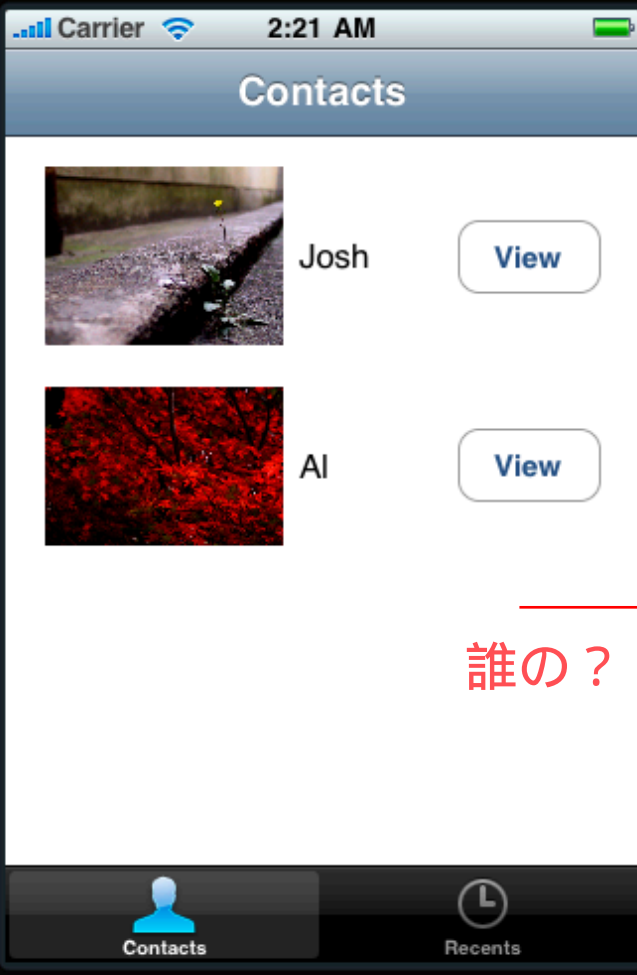
Demo: Pushing & Popping

Application Data Flow

View の階層を行き来するとき
どのようなデータをどのようにやりとりするか

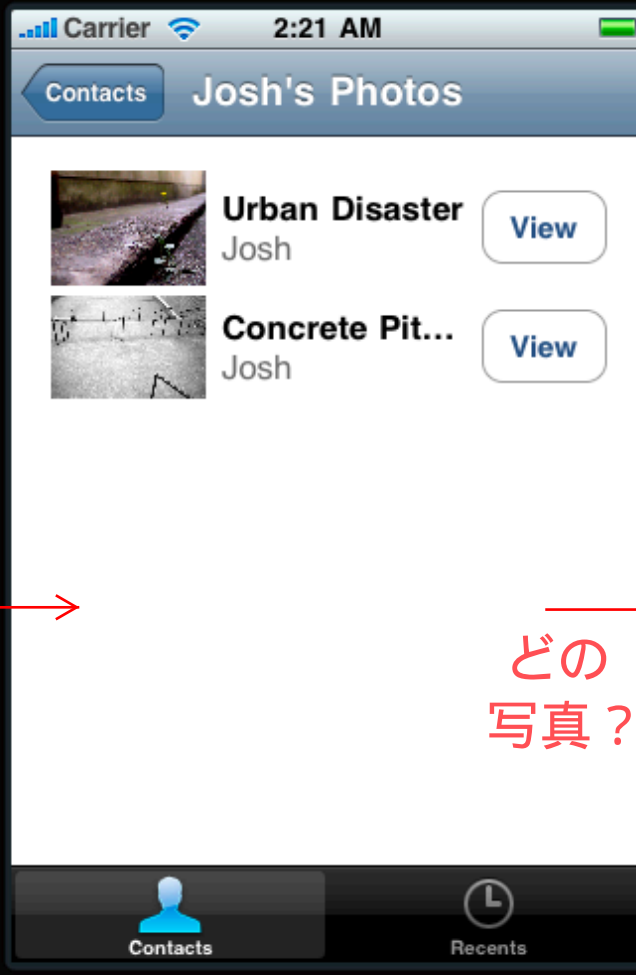
Paparazzi

この場面に
1つの View Controller



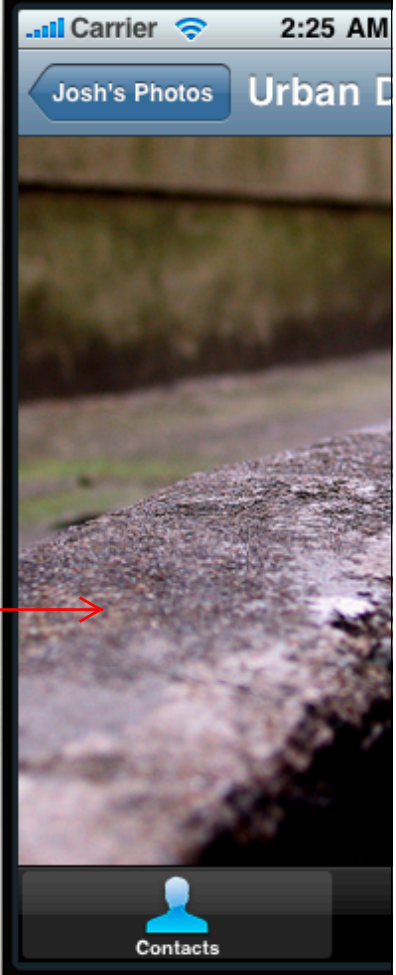
誰の？

この場面に
1つの View Controller



どの
写真？

この場面に
1つの View C



A Controller for Each Screen

スクリーンごとのコントローラ



View Controller のつなげかた

Connecting View Controllers

- Multiple view controllers may need to **share data**
複数の VC 間で共有データが必要
- One may need to know about what another is doing
 - Watch for added, removed or edited data
 - Other interesting events

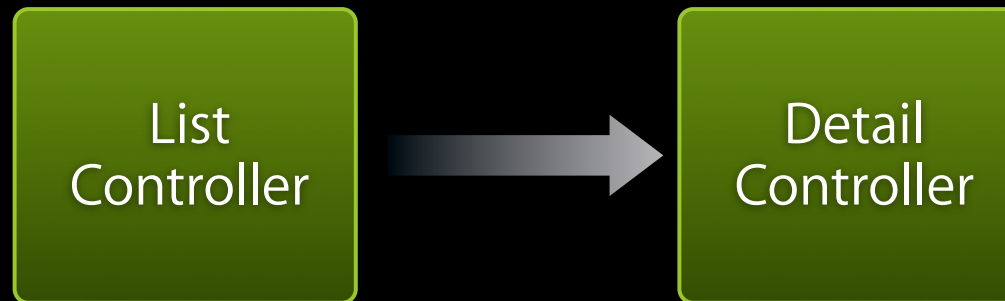
他の VC が何をしているのかを知りたい

- データの追加/削除/変更
- その他のイベント

How Not To Share Data

データを共有しない
ようにするには...

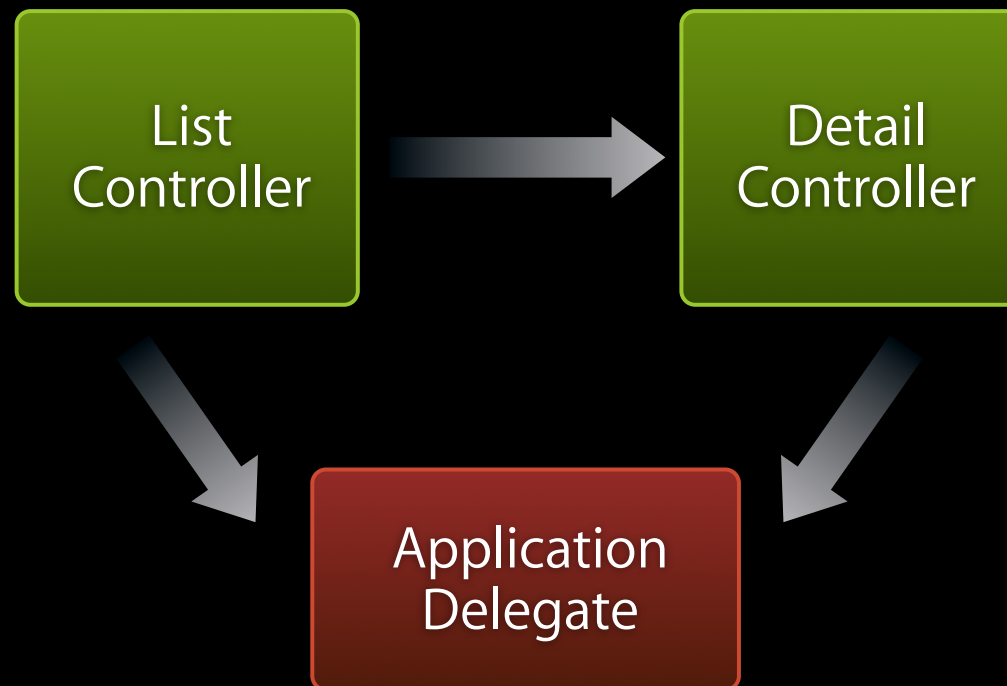
- Global variables or singletons グローバル変数や唯一オブジェクト？
 - This includes your **application delegate!**
- Direct dependencies make your code less reusable
 - And more difficult to debug & test



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

このような依存性をつくると
プログラムの再利用性が
なくなり
デバッグなどが
やりづらくなる



How Not To Share Data

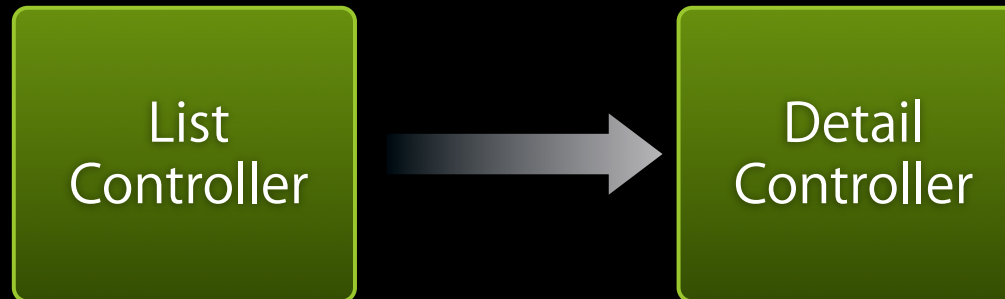
- Global variables or singletons
 - This includes your **application delegate**!
- Direct dependencies make your code less reusable
 - And more difficult to debug & test



では、どのようにデータの流れるか

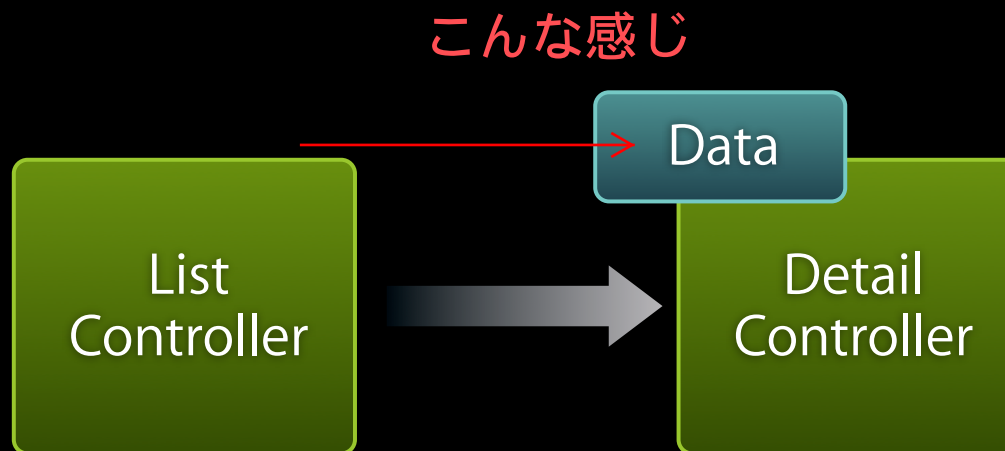
Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated
必要最低限のコミュニケーションを見極め...
- **Define input parameters** for your view controller
View Controller への入力パラメータとして設定する



Best Practices for Data Flow

- Figure out **exactly** what needs to be communicated
- **Define input parameters** for your view controller

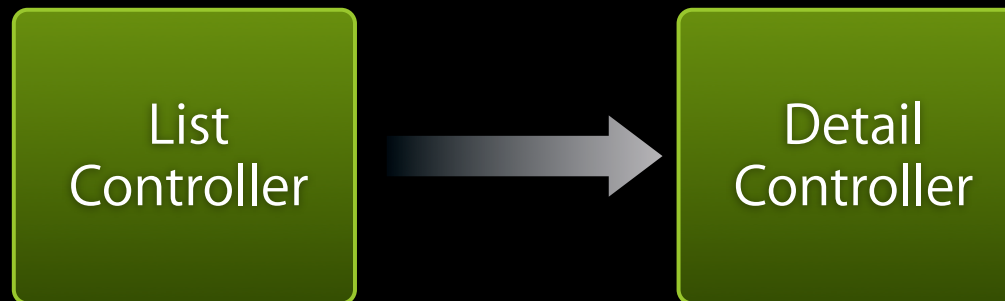


Best Practices for Data Flow

- 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)

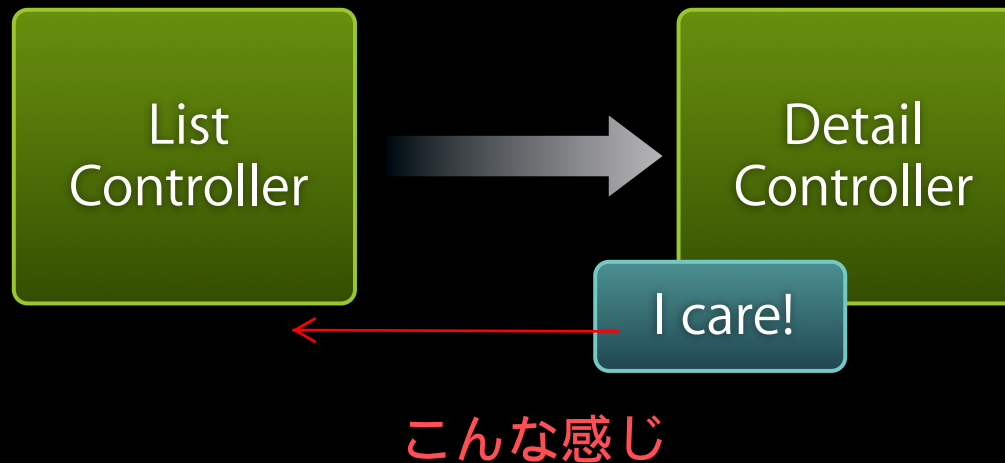
逆向きのコミュニケーションには、ルースカップリングを使う。

- デリゲーションを使う



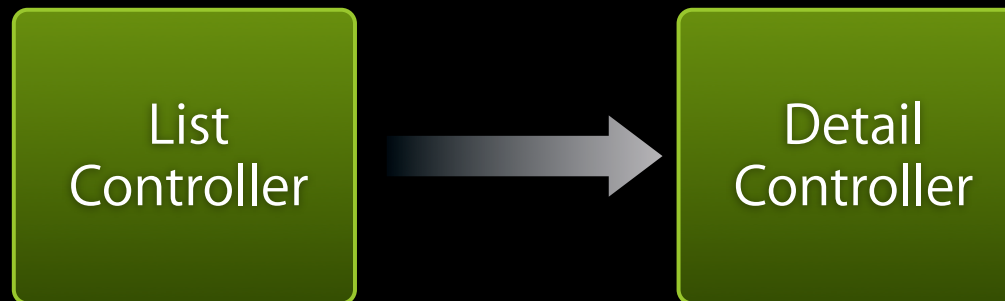
Best Practices for Data Flow

- 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)



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)



Example: UIImagePickerController

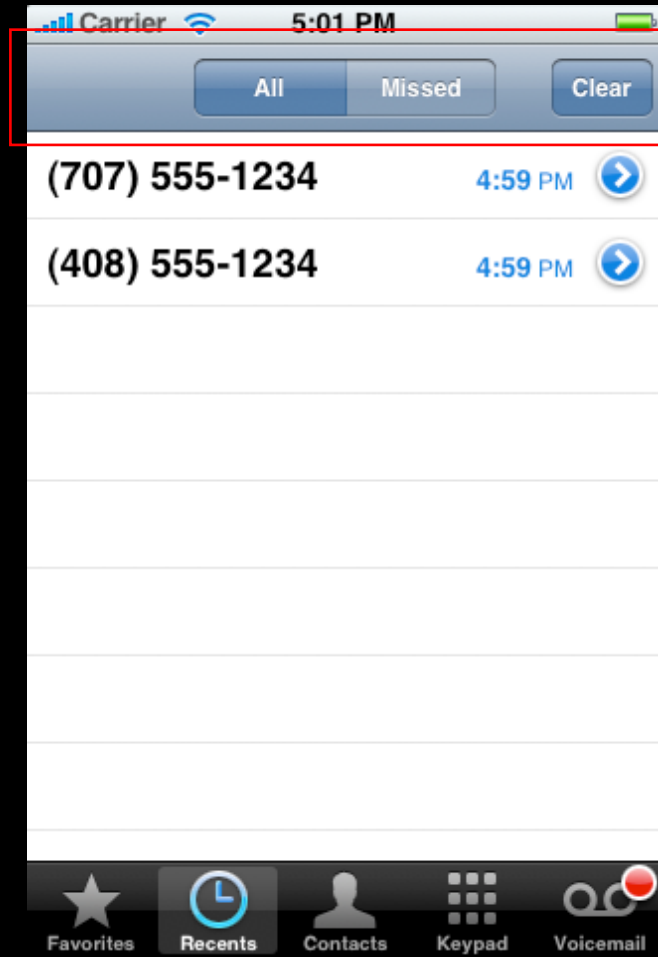
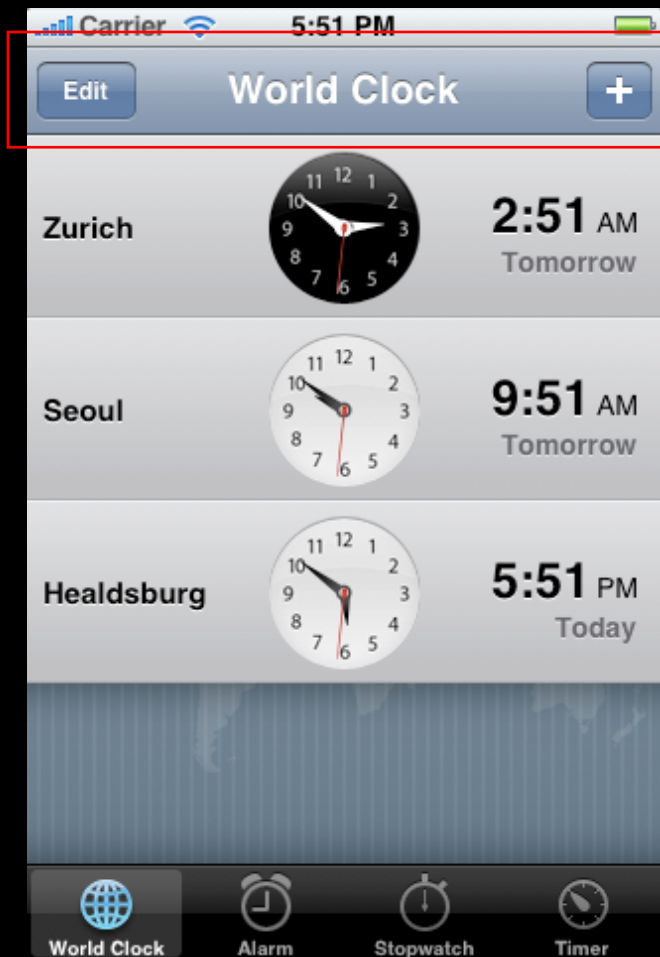
Demo: Passing Data Along

Customizing Navigation

ナビゲーションのカスタム化

Customizing Navigation ナビゲーションのカスタム化

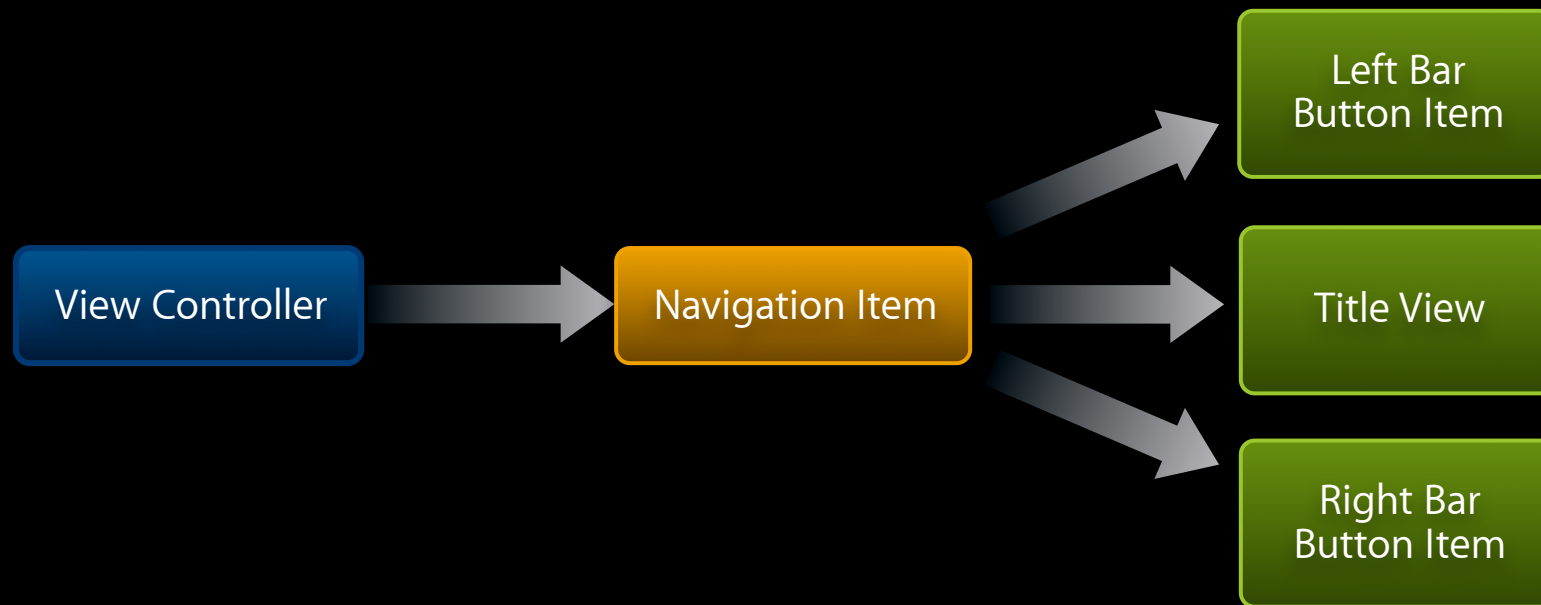
- Buttons or custom controls ボタンや他のコントロール（操作対象）
- Interact with the entire screen スクリーン全体でインタラクション



UINavigationController すべての VC が内蔵するアイテム

- Describes appearance of the navigation bar バーの外見を決める
 - Title string or custom title view タイトル
 - Left & right bar buttons 左右のボタン
 - More properties defined in UINavigationController.h その他
- **Every view controller has a navigation item** for customizing
 - Displayed when view controller is on **top of the stack**
これらアイテムはナビゲーションの中に自動的に表示される

Navigation Item Ownership



Displaying a Title

- UINavigationController already has a title property VC は元々
タイトルを持っている
 - @property(nonatomic, copy) NSString *title;
- Navigation item inherits automatically ナビゲーションアイテム
はそれを
自動的に
継承する
 - Previous view controller's title is displayed in back button 前の VC のタイトルが「戻る」ボタンに表示される



```
viewController.title = @"Detail";
```

Left & Right Buttons 左右のボタン

- UIBarButtonItem
 - Special object, defines appearance & behavior for items in navigation bars and toolbars
 - Display a string, image or predefined system item
 - Target + action (like a regular button) 「新規メール作成」など
アップル標準のピクトグラムも
用意されている
- 普通のボタンのように
ターゲット + アクションを
設定することもできる .

Text Bar Button Item

テキスト バー ボタン



```
- (void)viewDidLoad
{
    UIBarButtonItem *fooButton = [[UIBarItem alloc]
        initWithTitle:@"Foo"           ボタンのタイトル
        style:UIBarItemStyleBordered   スタイル
        target:self                     ターゲット
        action:@selector(foo:)]       アクション];

    self.navigationItem.leftBarButtonItem = fooButton;
        左ボタンに設定
    [fooButton release];
}
        ここから解放 (設定先の VC 側で retain される)
```

System Bar Button Item システム バー ボタン

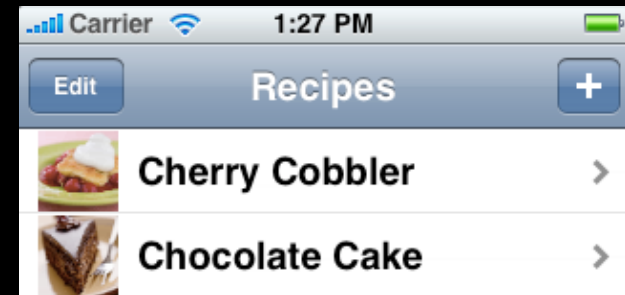


```
- (void)viewDidLoad
{
    UIBarButtonItem *addButton = [[UIBarButtonItem alloc]
        initWithBarButtonSystemItem:UIBarButtonSystemItemAdd
        style:UIBarButtonItemStyleBordered           「追加ボタン」
        target:self
        action:@selector(add:)];

    self.navigationItem.rightBarButtonItem = addButton;
        右ボタンに設定
    [addButton release];
}
```

Edit/Done Button 「編集」「(編集)終了」ボタン

- Very common pattern よく使うので...
- Every view controller has one available
 - Target/action already set up
あらかじめ VC に用意されている



用意されているボタン

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

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

```
- (void)setEditing:(BOOL)editing animated:(BOOL)animated
```

```
{ このメソッドが呼ばれる (editing=YES/NO で Edit/Done を区別)  
  // Update appearance of views
```

```
}
```

Custom Title View タイトルのカスタム化

- Arbitrary view in place of the title 任意の画面要素をタイトル部分に入れてよい



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

Back Button 「戻る」ボタン

- Sometimes a shorter back button is needed
- デフォルトでは直前のタイトルだが
もっと短いボタンに
したい場合がある



```
self.title = @"Hello there, CS193P!";
```


Back Button

- Sometimes a shorter back button is needed



```
self.title = @"Hello there, CS193P!";
```

```
UIBarButtonItem *heyButton = [[UIBarButtonItem alloc]  
                             initWithTitle:@"Hey!"  
                             ...];
```

```
self.navigationItem.backButtonItem = heyButton;  
新しい UIBarButtonItem を設定  
[heyButton release];
```

Back Button

- Sometimes a shorter back button is needed



すると, こうなる .

```
self.title = @"Hello there, CS193P!";
```

```
UIBarButtonItem *heyButton = [[UIBarButtonItem alloc]  
                               initWithTitle:@"Hey!"  
                               ...];
```

```
self.navigationItem.backButtonItem = heyButton;
```

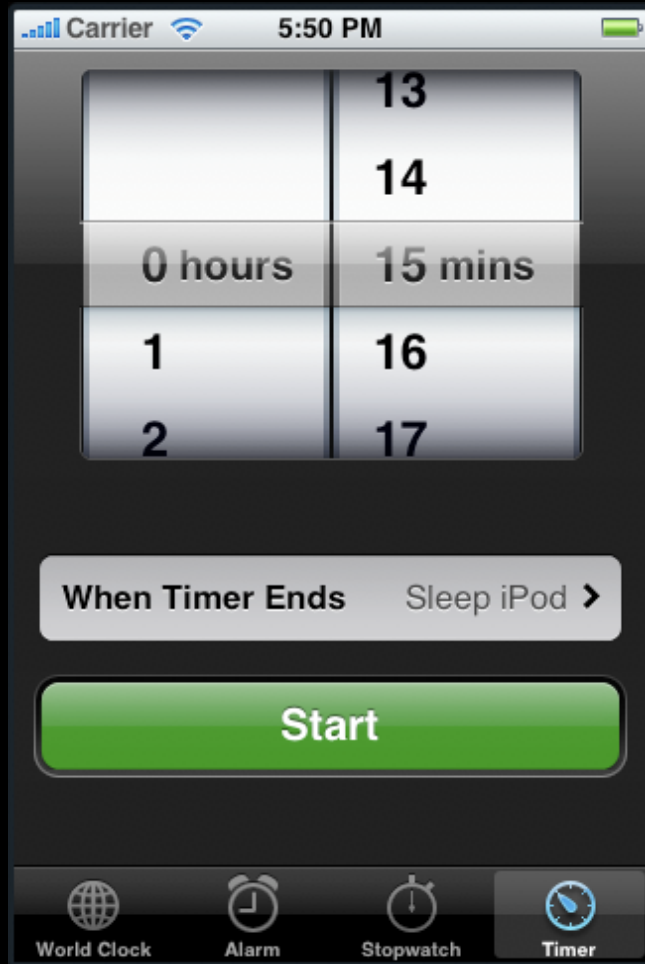
```
[heyButton release];
```

Demo: Customizing Buttons

Tab Bar Controllers

タブバーコントローラ

UITabBarController



機能切替ボタン

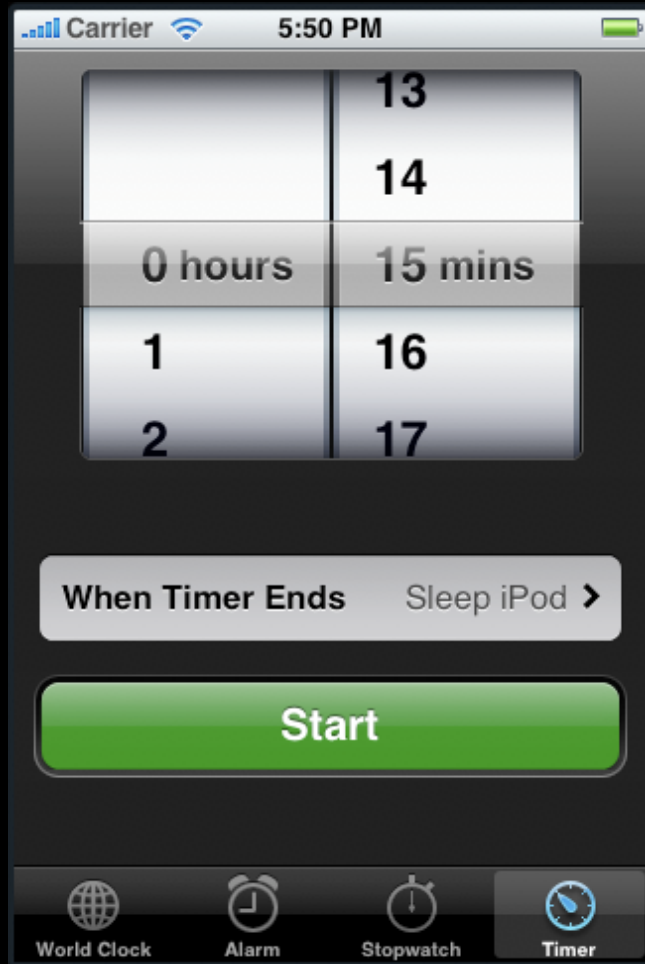
- Array of view controllers

- Tab bar

スタックではなく
ただの VC 配列

Tab Bar
Controller

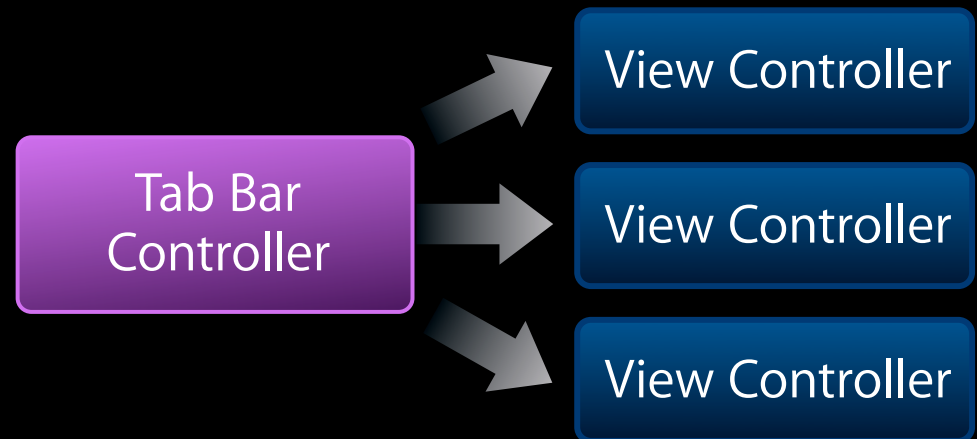
UITabBarController



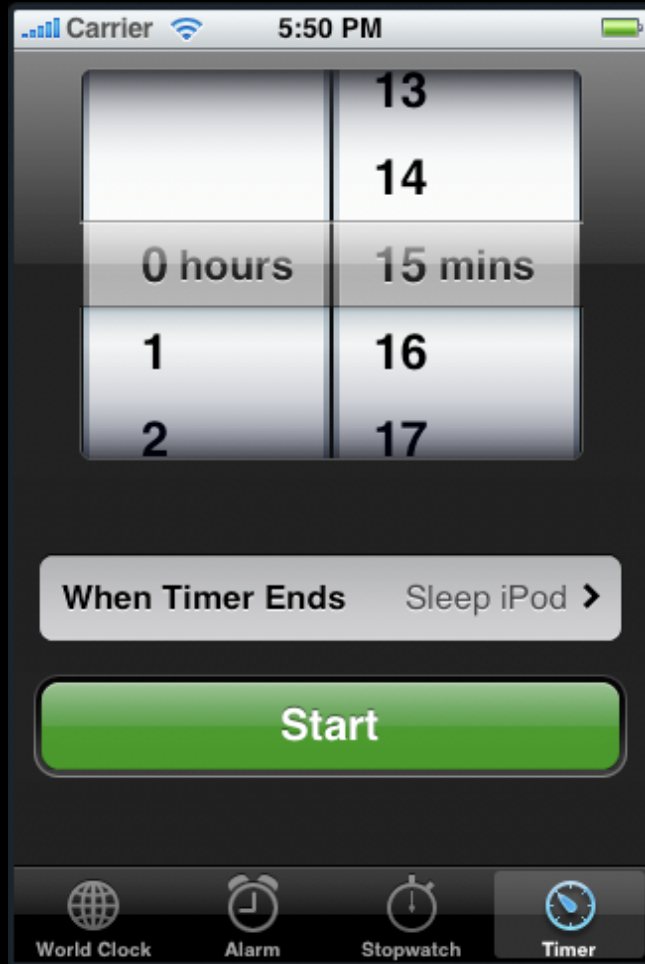
- Array of view controllers

- Tab bar

スタックではなく
ただの VC 配列

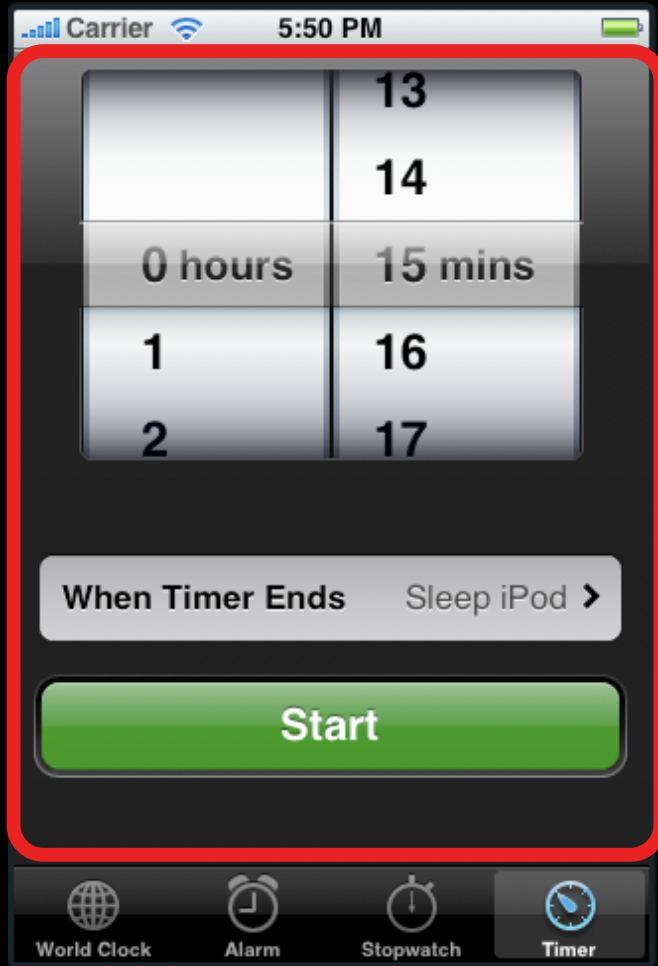


How It Fits Together

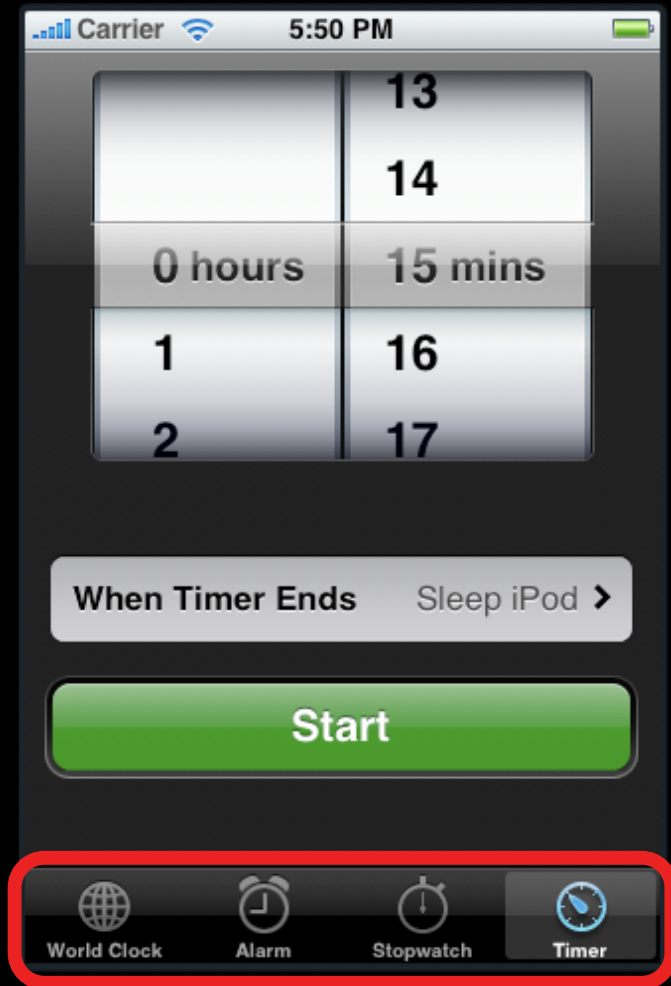


How It Fits Together

- Selected view controller's view



How It Fits Together



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

タブバーには
各 VC のタイトルが入る
(イメージも入る)

アップル標準のアイコン

タブバー

Setting Up a Tab Bar Controller セットアップ方法

アプリ起動直後に呼び出されるメソッド

- (void)applicationDidFinishLaunching

// Create a tab bar controller タブバーコントローラの生成

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

// Set the array of view controllers NSArray (of VC) をセット

```
tabBarController.viewControllers = myViewControllers;
```

// Add the tab bar controller's view to the window

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

```
} ベース window にタブバーコントローラをセット
```

Tab Bar Appearance タブバーの外見

- View controllers can define their appearance in the tab bar



- UITabBarItem オリジナルのタイトル + イメージ
 - Title + image or system item あるいはシステム標準アイテム
- Each view controller comes with a tab bar item for customizing すべての VC はタブバーアイテムを持っている .
これをカスタム化すればよい .

Creating Tab Bar Items タブバーアイテムの生成

- Title and image オリジナルのアイテムの場合



Playlists

```
- (void)viewDidLoad
{
    UITabBarItem *item = [[UITabBarItem alloc]
                          initWithTitle:@"Playlists"
                          image:[UIImage imageNamed:@"music.png"]
                          tag:0];
    self.tabBarItem = item;
    [item release];
}
```

Creating Tab Bar Items

- System item システム標準のアイテムの場合



Bookmarks

```
- (void)viewDidLoad
{
    UITabBarItem *item = [[UITabBarItem alloc]
                          initWithTabBarItemSystemItem:
                            UITabBarItemSystemItemBookmarks
                            tag:0]
    self.tabBarItem = item;
    [item release];
}
```

Demo: Using a Tab Bar Controller

More View Controllers

たくさんのアイテムを
タブバーに並べるには...

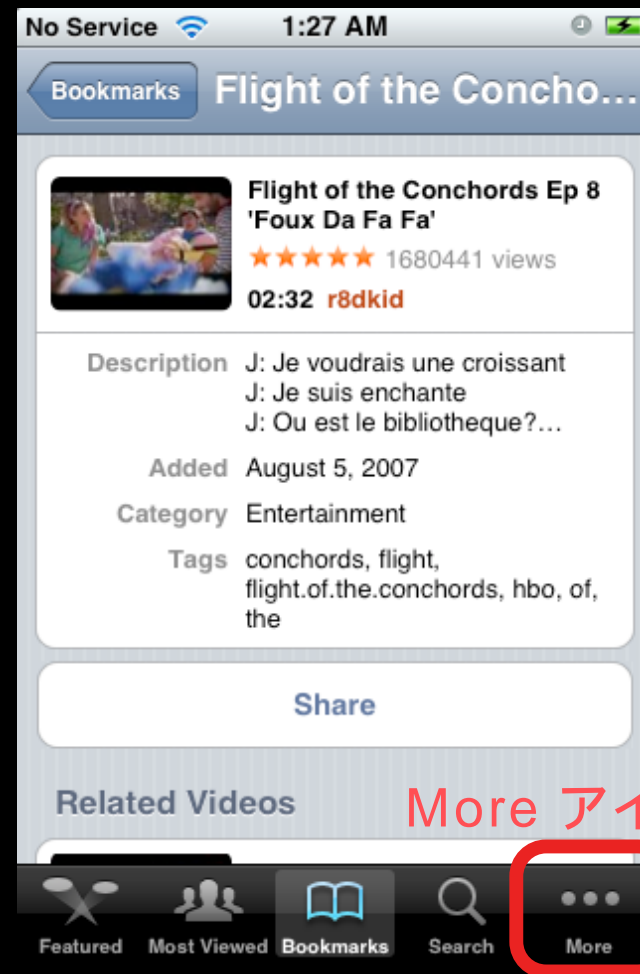
- What happens when a tab bar controller has too many view controllers to display at once?



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

More アイテムが自動的に入れられる

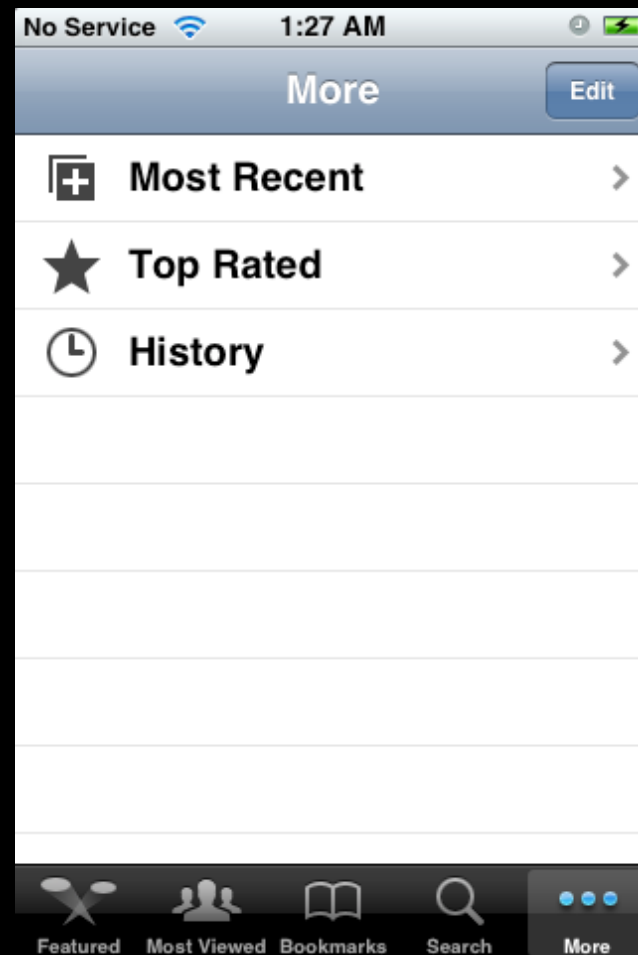


More アイテム

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

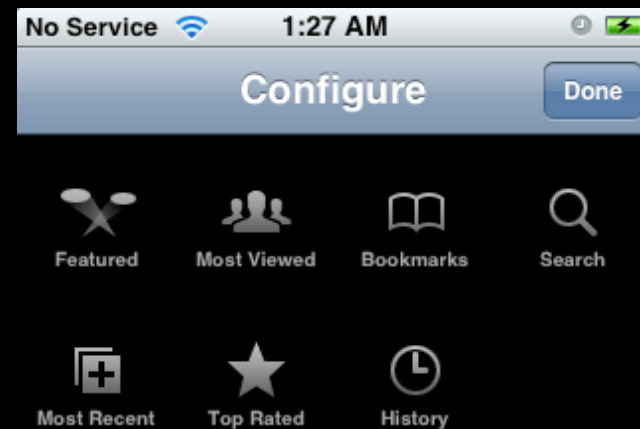
More をタップすれば
残りのアイテムの一覧が出る
さらに右上の Edit を押すと...



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
 - Customize order

アイテムの順序を
カスタマイズできる
(すべて Tab Bar Controller に
内蔵された機能)



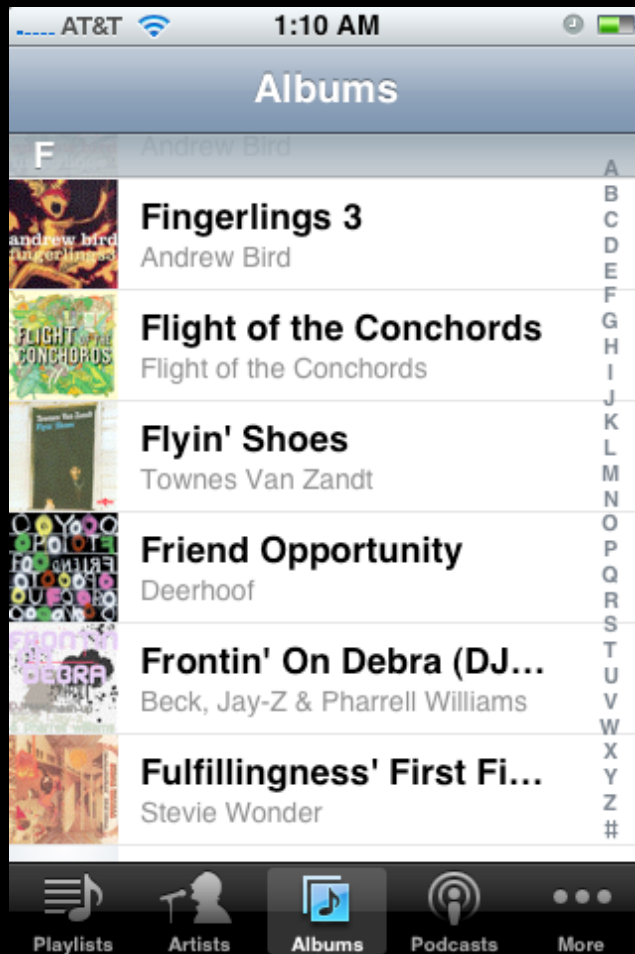
Combining Approaches

ナビゲーションとタブバーを組み合わせると...

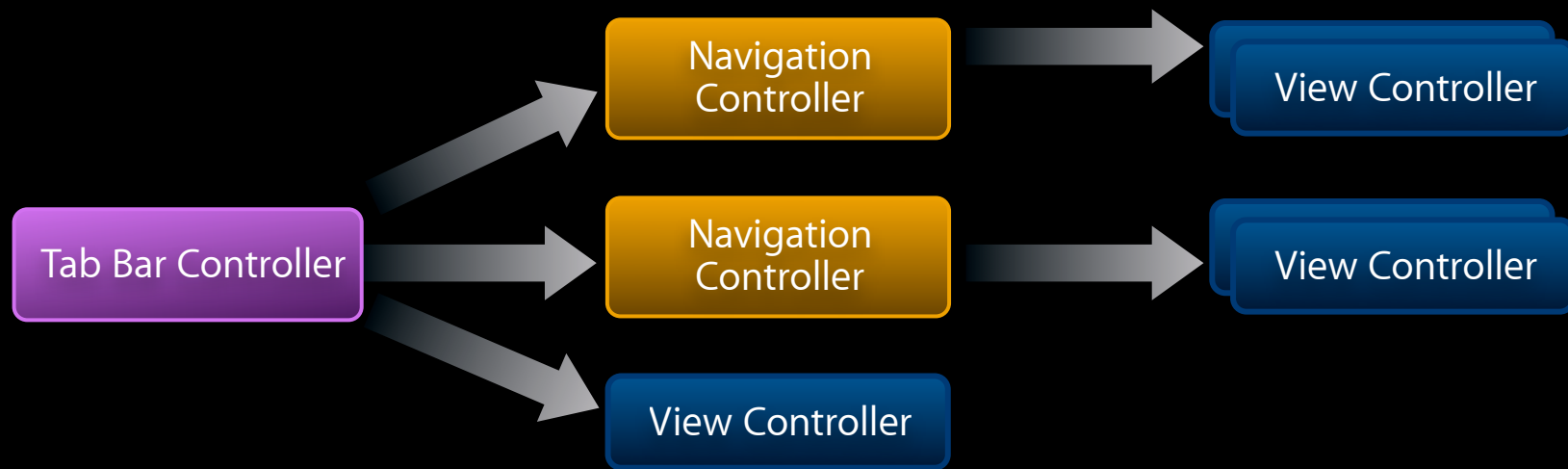
Tab Bar + Navigation Controllers

Multiple parallel hierarchies

並列された階層構造



Tab Bar + Navigation Controllers



入れ子になったナビゲーションコントローラ

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];
```

Questions?