Praktikum Entwicklung von Mediensystemen mit iOS

WS 2011

Prof. Dr. Michael Rohs
michael.rohs@ifi.lmu.de
MHCI Lab, LMU München

Today

- Alerts, Action Sheets, text input
- Application architecture
- Table views
- Multiview applications
- Touch input
- Saving data
- Exercise 2

Timeline

#	Date	Topic
1	19.10.2011	Introduction and overview of iOS
2	26.10.2011	App architecture, touch input, saving data
3	2.11.2011	Location, networking, sensors
4	16.11.2011	Interviews, storyboarding; brainstorming
5	30.11.2011	Paper prototyping test, start of software prototype
6	14.12.2011	Heuristic evaluation of software prototype
7	11.1.2012	Think-aloud user study
8	25.1.2012	Completion of software prototype
9	1.2.2012	Final presentation

Organization

- 4 SWS
- (Bi-)Weekly meetings
  - Wednesday 16:10 – 17:40
  - Room 107, Amalienstraße 17
- Homepage:
  - http://www.medien.ifi.lmu.de/lehre/ws1112/pem/

iOS Developer Account

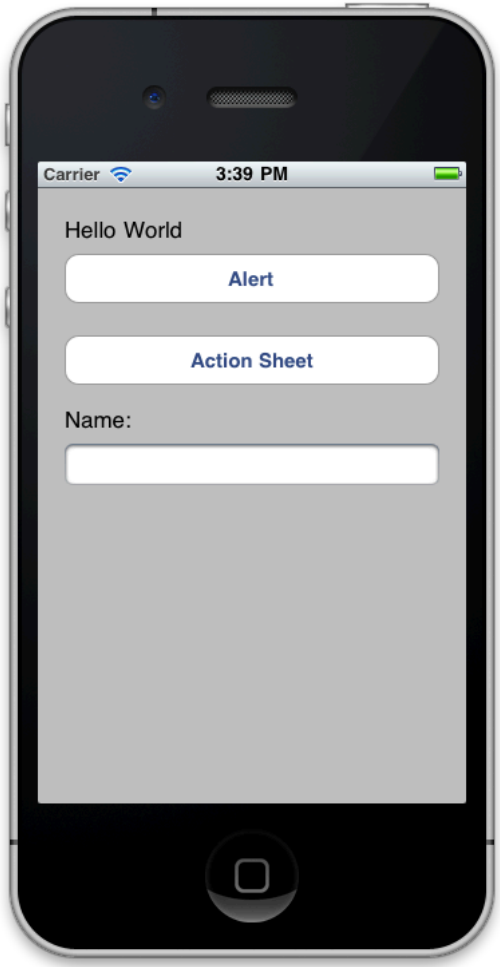
① University Account
② Send email, we invite you
③ Create certificate
④ Register as developer
⑤ We send provisioning profile

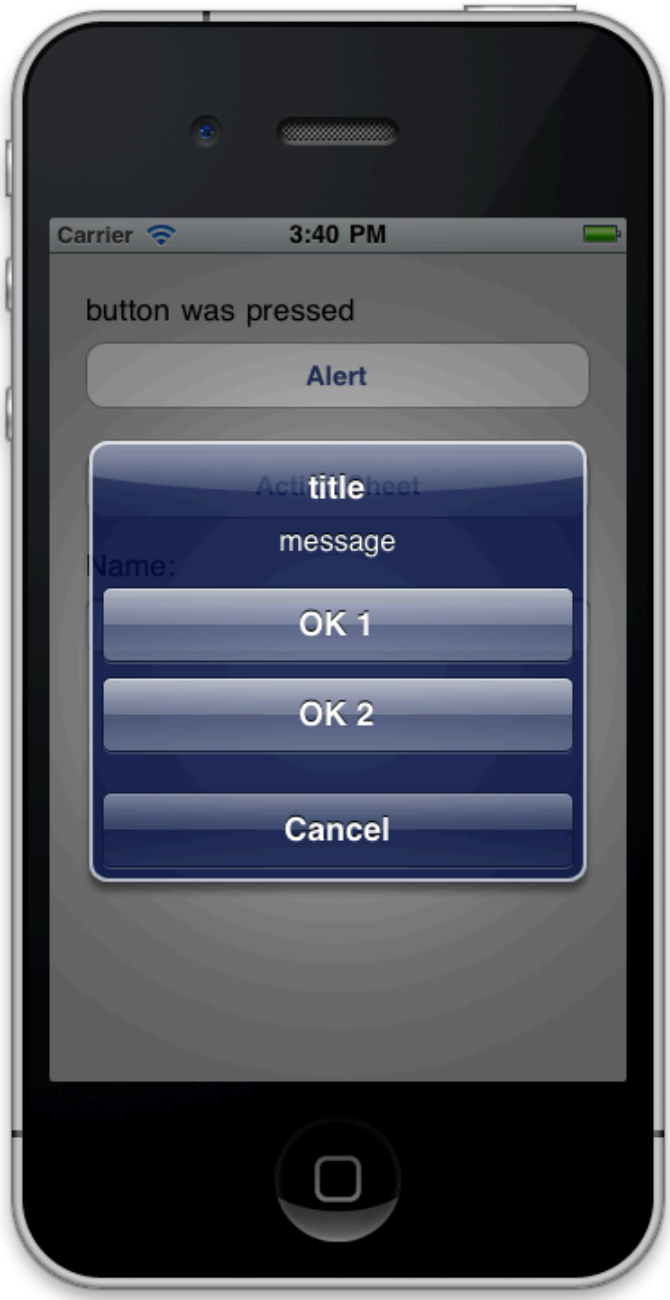
Organization

- UniWorX for individual exercises
  - https://uniworx.ifi.lmu.de
- SVN for teamwork
  - SVN accounts for each team
  - svn://tracsvn.medien.ifi.lmu.de/repos/pem_team[number] (e.g. svn://tracsvn.medien.ifi.lmu.de/repos/pem_team1)

“Hello World” Steps

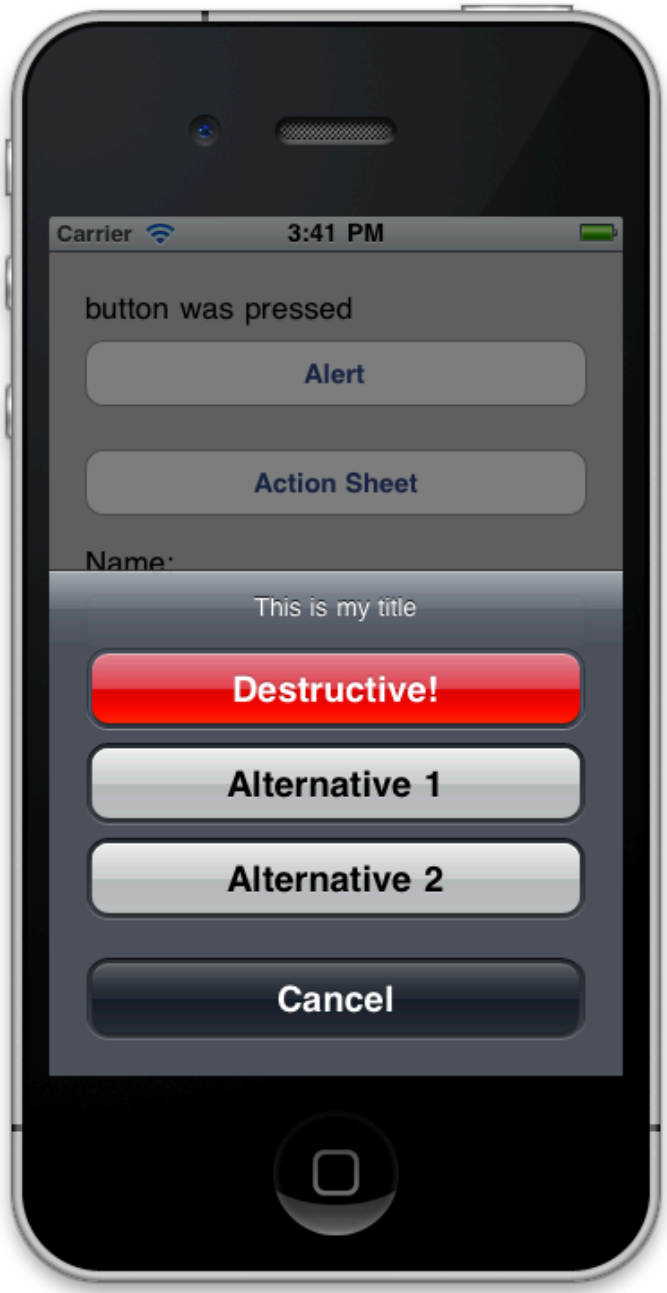
- Explain #pragma mark -
- Showing a UIAlertView





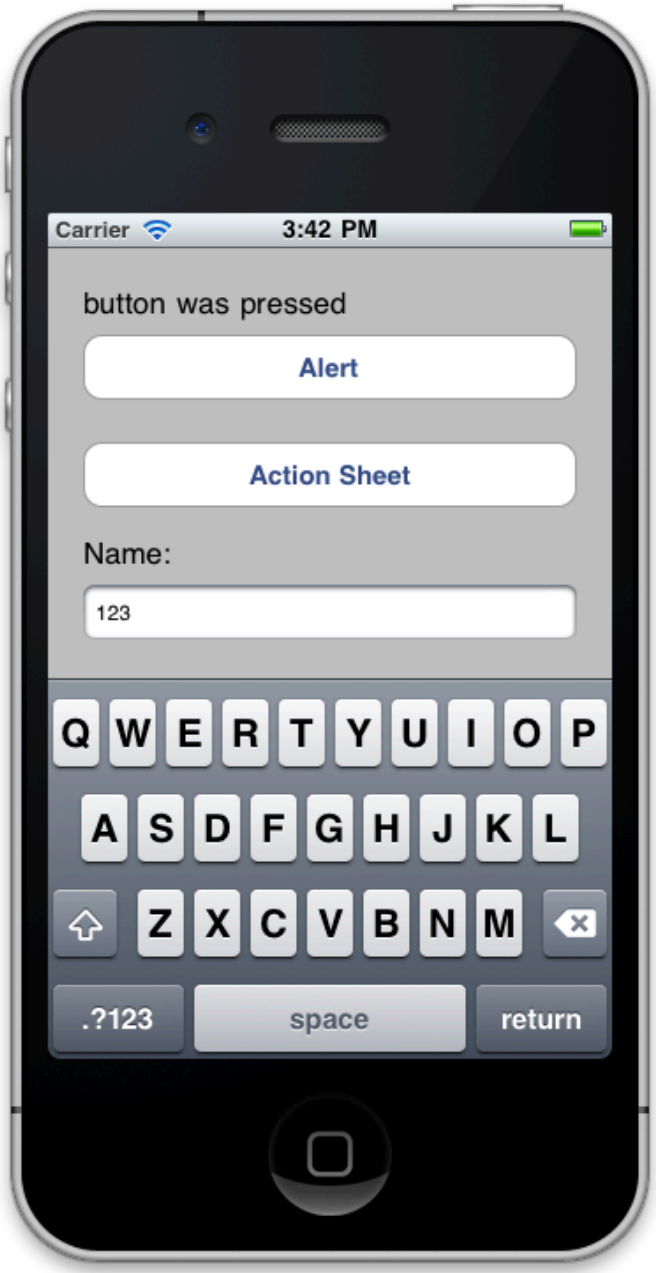
“Hello World” Steps

- Action sheets
  - Implement UIActionSheetDelegate in .h file
  - Construct, showInView, release
  - Implement delegate method clickedButtonAtIndex

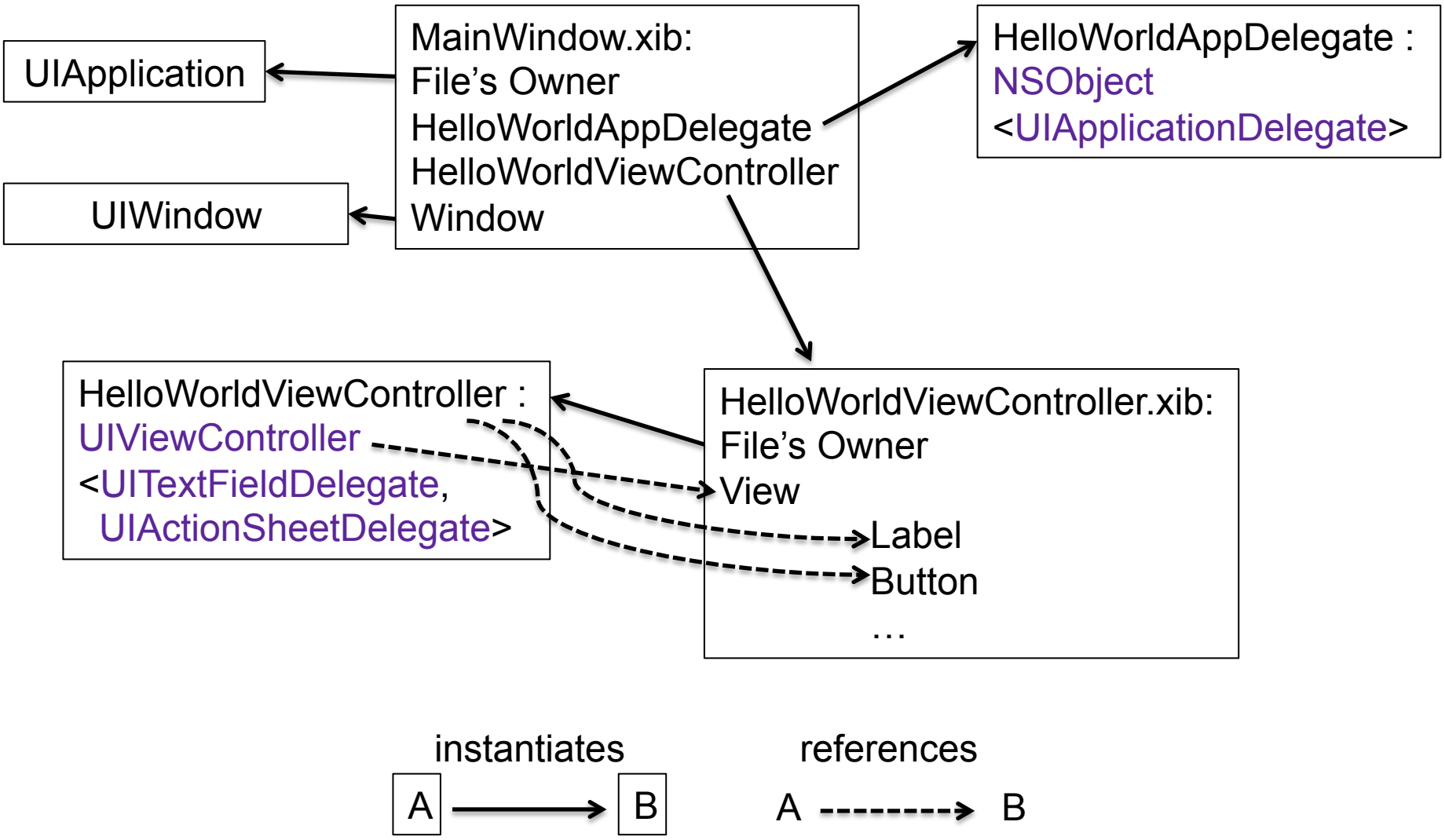


“Hello World” Steps

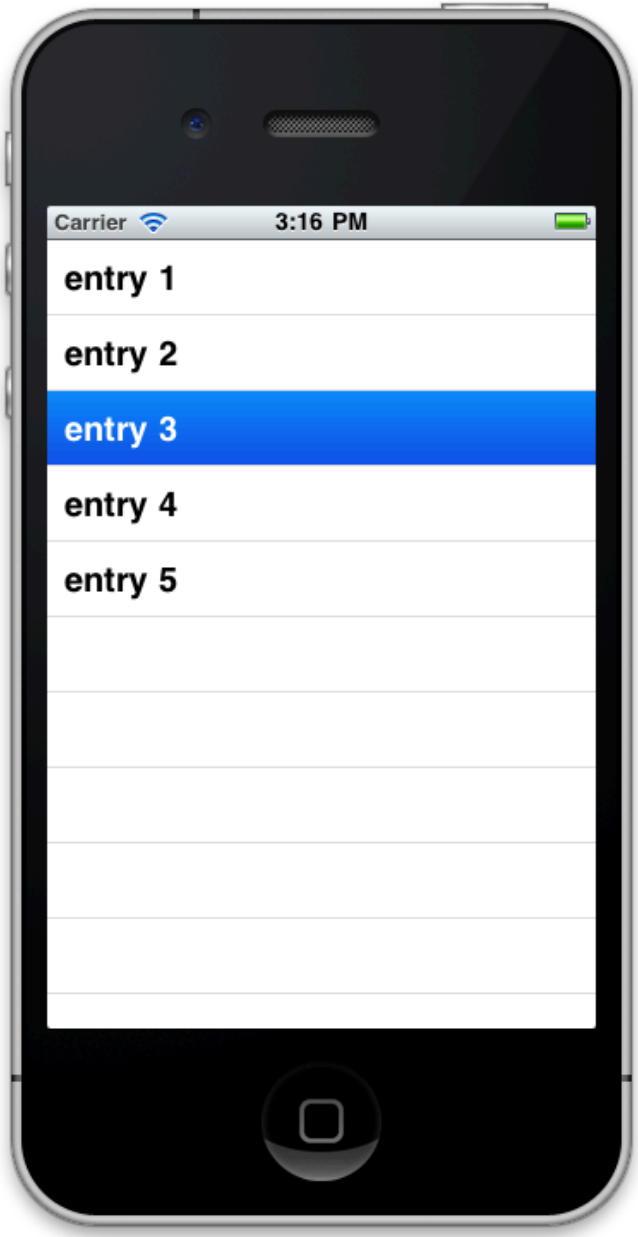
- Text input
  - Add UITextField in Interface Builder
  - Add member variable and property to .h, synthesize in .m
  - Declare UITextFieldDelegate in .h
  - Implement delegate methods in .m, set label text on end editing
  - Set delegate in viewDidLoad method



Hello World Application Architecture



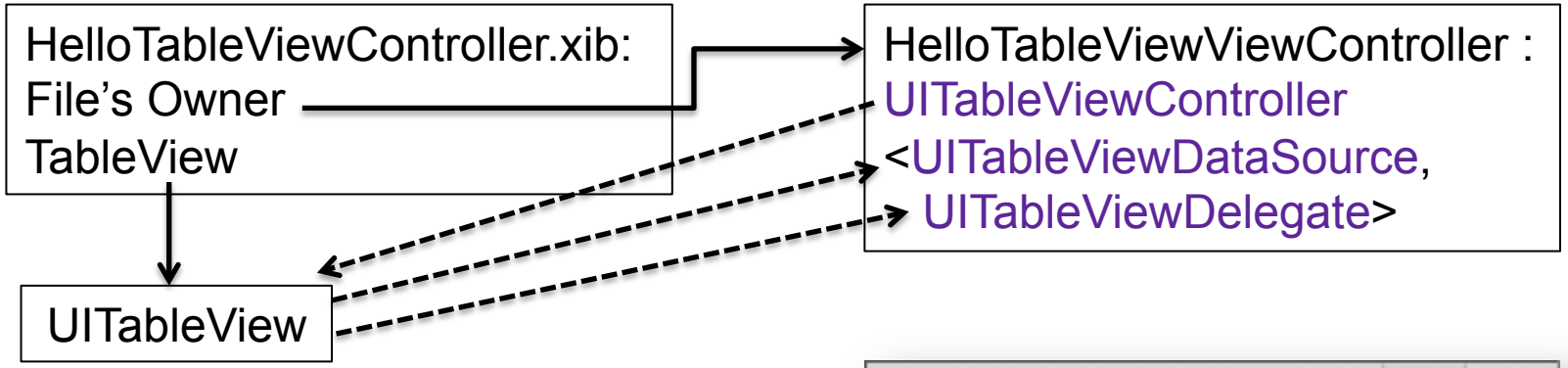
HELLO TABLEVIEW

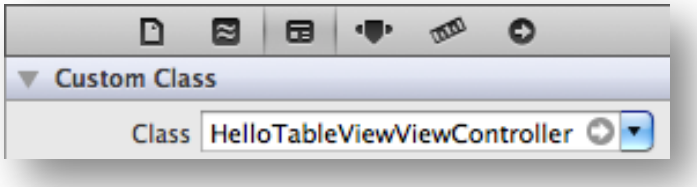


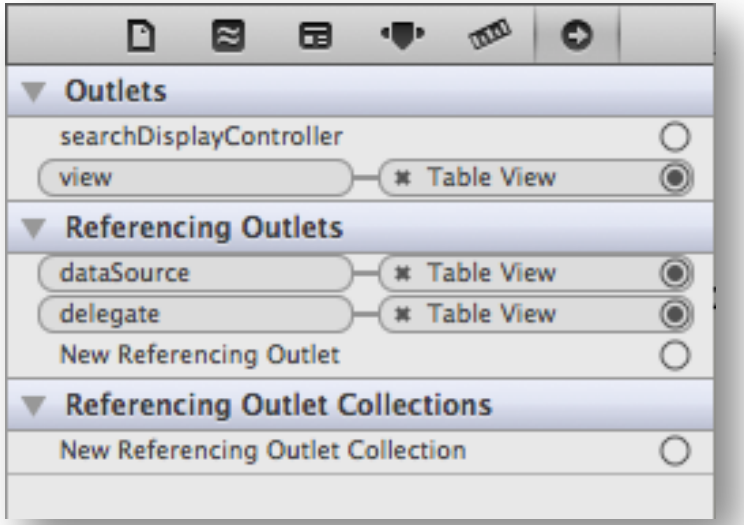
UITableView Example

- Create new project (“View-based application”)
- Change controller base class to UITableViewController
- Declare UITableViewDataSource, UITableViewDelegate
- Add data array to header file, release data in dealloc
- Change view in nib file to UITableView, connect File’s Owner (view, data source, delegate)
- Create arrayWithObjects in onViewDidLoad
- Implement table data source and delegate methods

Table Views


HelloTableViewController.xib:
File's Owner
TableView
UITableView
HelloTableViewViewController :
UITableViewController
<UITableViewDataSource,
UITableViewDelegate>


Custom Class
Class
HelloTableViewViewController
Custom Class
Class
UITableView


Outlets
searchDisplayController
view
Table View
Referencing Outlets
dataSource
Table View
delegate
Table View
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection


instantiates
A
B
references
A
B

UIViewController subclasses

- View lifecycle
  - (void)viewDidLoad
  - (void)viewDidUnload
- View events
  - (void) viewWillAppear:(BOOL)animated
  - (void) viewWillDisappear:(BOOL)animated
  - (void) viewDidAppear:(BOOL)animated
  - (void) viewDidDisappear:(BOOL)animated
- Rotation settings and events

  interfaceOrientation property

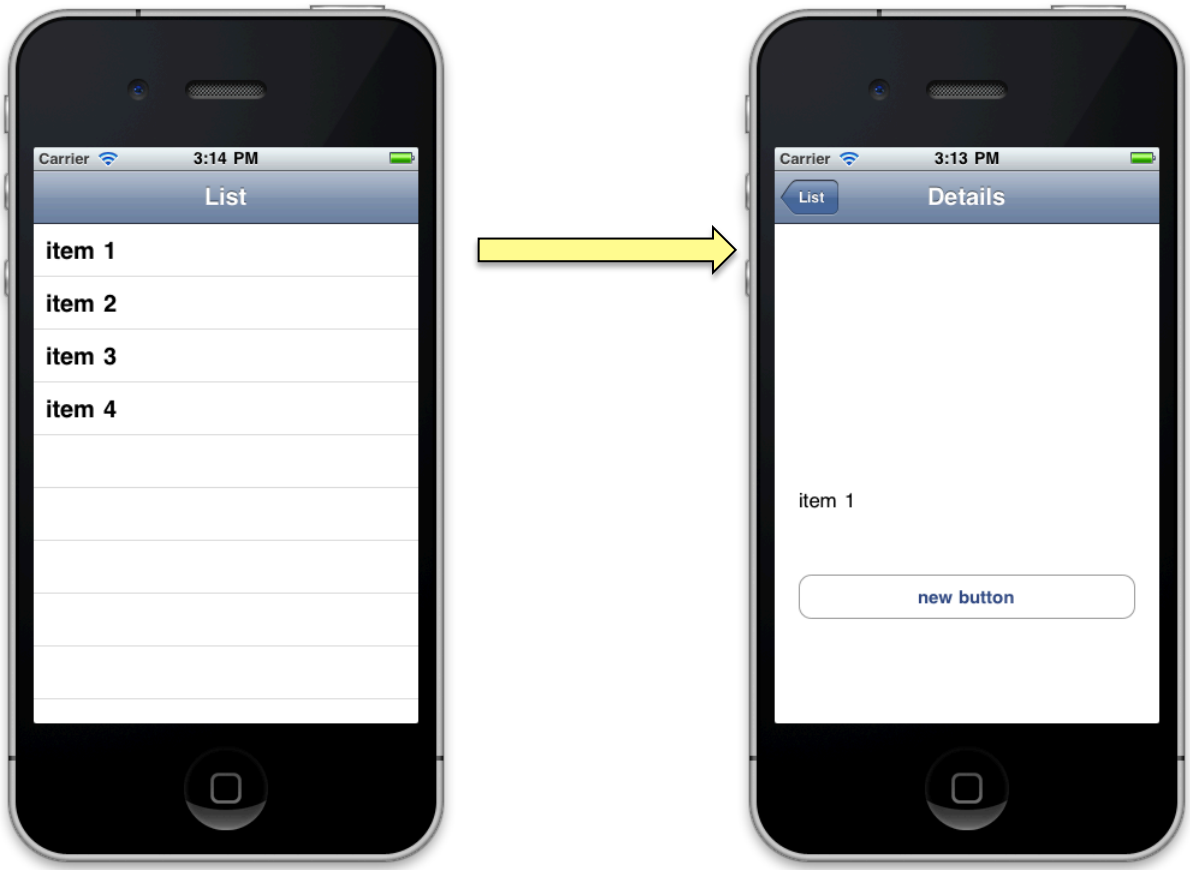
  – shouldAutorotateToInterfaceOrientation:
- many more… → see documentation

UITableViewDataSource (Protocol)

- Configuring a Table View
  - tableView:cellForRowAtIndexPath: required method
  - numberOfSectionsInTableView:
  - tableView:numberOfRowsInSection: required method
  - sectionIndexTitlesForTableView:
  - tableView:sectionForSectionIndexTitle:atIndex:
  - tableView:titleForHeaderInSection:
  - tableView:titleForFooterInSection:
- Inserting or Deleting Table Rows
  - tableView:commitEditingStyle:forRowAtIndexPath:
  - tableView:canEditRowAtIndexPath:
- Reordering Table Rows
  - tableView:canMoveRowAtIndexPath:
  - tableView:moveRowAtIndexPath:toIndexPath:

UITableViewDelegate (Protocol)

- Configuring Rows for the Table View
  - tableView:heightForRowAtIndexPath:
- Managing Accessory Views
  - tableView:accessoryButtonTappedForRowWithIndexPath:
- Managing Selections
  - tableView:{will,did}SelectRowAtIndexPath:
  - tableView:{will,did}DeselectRowAtIndexPath:
- Modifying the Header and Footer of Sections
  - tableView:viewFor{Header,Footer}InSection:
  - tableView:heightFor{Header,Footer}InSection:
- Editing Table Rows
- Reordering Table Rows



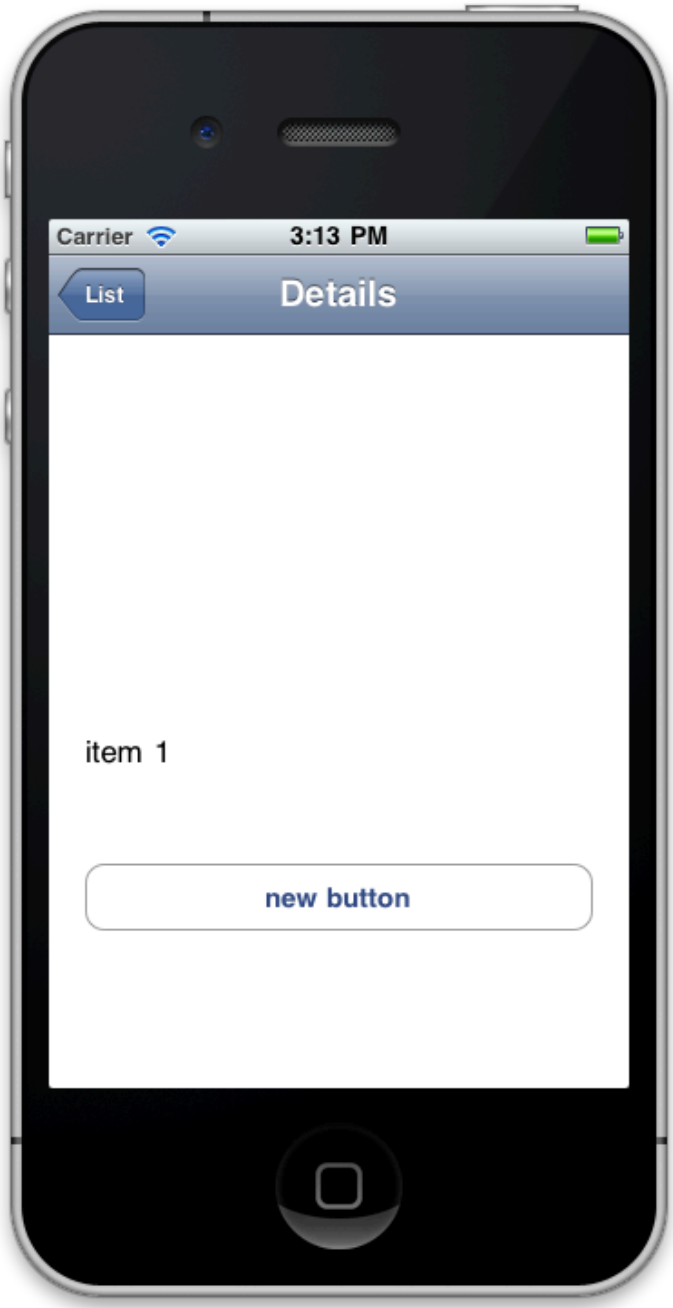
HELLO MULTIVIEW

View Navigation Example

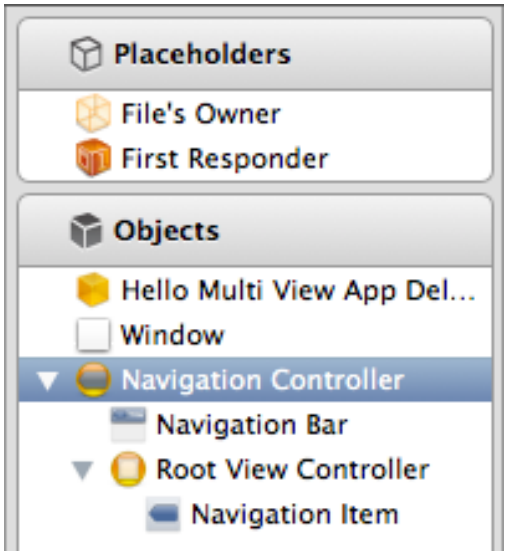
- Create a “Navigation-Based Application”
- Add NSArray *data to RootViewController
  - Add some data in onViewLoad, retain!
- New File… → UIViewController subclass (with nib file) → “MyDetailViewController”
  - Add UILabel to nib file and to .h file (IBOutlet, @property) and to .m file (@synthesize)
- #import "MyDetailViewController.h”
- Implement didSelectRowAtIndexPath, set selected item
- Show that it does not work ☺ → Debugger
- Show that label is still nil → use member variable, set label in viewDidLoad

View Navigation Example

- Add back button:
  self.navigationItem.title = @"List";



MultiView Application Architecture



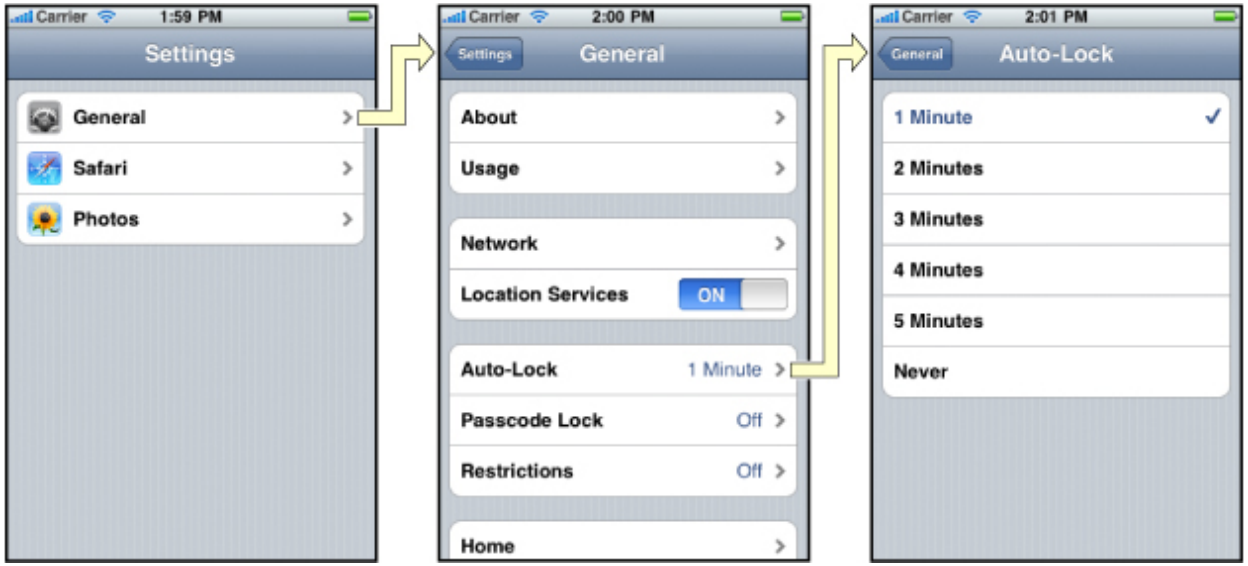
Navigation Controller Views

Source: http://developer.apple.com/library/ios/#documentation/uikit/reference/UINavigationController_Class/Reference/Reference.html

Pushing a new View onto the View Stack

- Loading and pushing the new view controller

```
MyDetailViewController *d = [[MyDetailViewController alloc]
initWithNibName:@"MyDetailViewController" bundle:nil];
d.labelText = [data objectAtIndex:indexPath.row];
[self.navigationController pushViewController:d animated:YES];
[d release];
```



Source: http://developer.apple.com/library/ios/#documentation/uikit/reference/UINavigationController_Class/Reference/Reference.html

How to Exchange Data between Views

- Set data before invoking new view controller

```
MyDetailViewController *dvc = [[MyDetailViewController alloc]
                    initWithNibName:@"MyDetailViewController" bundle:nil];
dvc.labelText = [data objectAtIndex:indexPath.row];
[self.navigationController pushViewController:dvc animated:YES];
```

- Use application delegate for “global” data

```
#import "HelloMultiViewAppDelegate.h”
…
HelloMultiViewAppDelegate *delegate = [UIApplication
                                        sharedApplication].delegate;
NSData *data = delegate.myGlobalData;
```

Touch Input

- Overwrite methods in UIView or UIImageView:

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    UITouch *touch = [touches anyObject];
    CGPoint p = [touch locationInView:self];
    traceCount = 0;
    trace[traceCount++] = p;
    [self updateDisplay];
}
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event;
```