Lesson 3: Storyboards and Segues

INTRODUCTION

Apple added The Storyboard tool with Xcode 4.2 which allows visual building of views. It also allows the developer to easily link views in Interface Builder.

LESSON OBJECTIVES

By the end of this lesson, the student will be able to:

1. Define: xib file, Interface Builder, storyboard, segue, and transition effect.
2. Explain the use of xib files in iOS applications.
3. Describe how a transition effect is added on a series of storyboard views using Interface Builder.
4. Describe how a storyboard view is set as the first scene in an app.
5. Explain issues caused by zooming when using storyboards.
6. Create an application that uses storyboards with multiple scenes and uses transition effects between scenes.

LEARNING SEQUENCE

Required Reading	Read the following: • Lesson 3: Storyboards and Segues
Resources	View the following: • *1/3 – Create a ViewController Using the .XIB File in Xcode Using Interface Builder and Xcode 5* (8:08) Other resources: • Xcode Tutorial 3: Storyboards, MapKit, View Navigation
Assignments	Complete the following: 1. Practice – Storyboards 2. Quiz 3

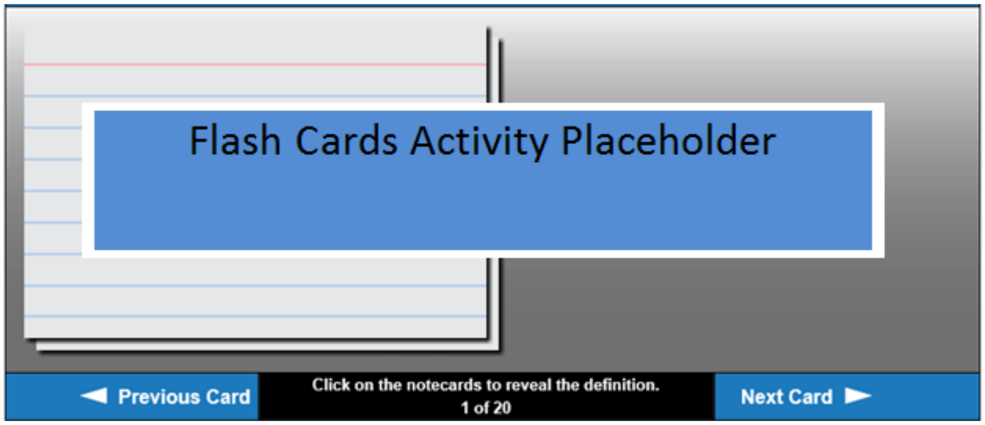
KEY TERMS

As you read your lesson, pay close attention to the key terms and phrases listed throughout the lesson. These terms and concepts are important to your understanding of the information provided in the lesson.

Karina Whetstine 2/17/2015 2:59 PM
Comment [1]: Please link to document titled:

L8_Key_Terms

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands


Flash Cards Activity Placeholder

Click on the notecards to reveal the definition.
1 of 20

INSTRUCTION

The Basics

Screens, or scenes, are designed and then linked with transitions between the scenes. Xcode generates the appropriate code.

A **segue** is the transition that defines the flow of an app by defining the relationship between scenes. A segue also sets transition effects such as a page fold, cross dissolve, or closing door, for example. A segue can be triggered with code in situations where the connection or transition cannot be drawn.

There are three directions that a developer can take when implementing a user interface:

- Storyboards: A visual tool for laying out multiple application views
- **Nibs** (or Xibs): Each nib file corresponds to a single view element that can be laid out in Interface Builder
- Code: No GUI tools used; everything is handled programmatically

Storyboard Files

Storyboard code, scenes and segues are saved in a .storyboard file. There is usually one **storyboard** file per app, but there are exceptions to this standard. For example, a universal app may have two different storyboard files; there may be one file for the iPhone and one for the iPad.

Storyboard files are edited in **Interface Builder**.

There are several advantages to using a Storyboard file:

- It streamlines switching between view controllers.
- It provides an overview of views (and view controllers) and shows how they connect to each other.

- It sets up UITableView content so that table creation is easier.

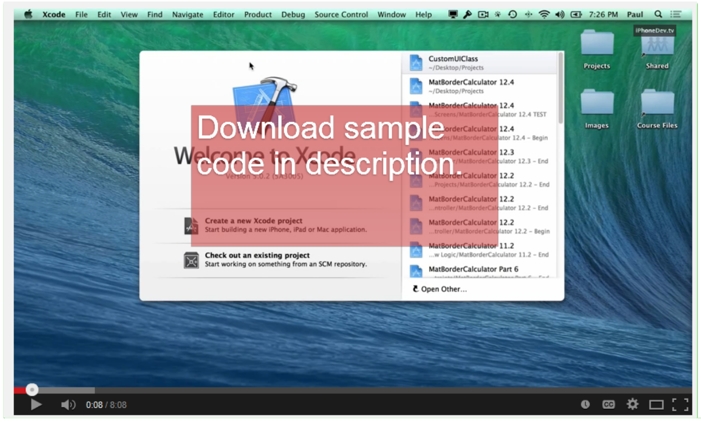
Storyboards are a complement to both nibs and custom code. Storyboards provide the best choice when an app has multiple interconnected view controllers. Conversely, storyboards are not the best choice when the view has a complicated or dynamic layout. In this type of app, code would work best.

A Nib File

A nib file stores the user interfaces of an iOS app and is an Interface Builder document. iOS uses nibs as an implementation detail that supports storyboards. When an app is built using **UIKit** frameworks, the framework supports the use of nib files for laying out windows, views, and controls. Those items can now be integrated with the app's event handling code. UIKit is the framework that provides common classes for user interface design.

Nibs should be used with **modal views** (a view that appears as a task related to the app's main function), simple login and registration views, popup windows, reusable view templates and reusable table cell templates. Avoid using nibs for views with dynamic content or view controllers with complicated transitions that can be simplified with storyboarding.

View the video, 1/3 – *Create a ViewController Using the .XIB File in Xcode Using Interface Builder and Xcode 5* (8:08), to learn how to make a new scene for an app using the .xib file in Xcode. The ViewController code will be connected to the .xib file.


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
7:26 PM Paul
Welcome to Xcode
Download sample code in description
Create a new Xcode project
Start building a new iPhone, iPad or Mac application.
Check out an existing project
Start working on something from an SCM repository.
CustomUIClass
MatBorderCalculator 12.4
MatBorderCalculator 12.3
MatBorderCalculator 12.2
MatBorderCalculator 11.2
MatBorderCalculator Part 6
Open Other...
Projects
Shared
Images
Course Files
0:08 / 8:08

Karina Whetstine 2/17/2015 2:59 PM
Comment [2]:

http://www.youtube.com/watch?v=bPLAz51EP2A

<iframe width="560" height="315" src="//www.youtube.com/embed/bPLAz51EP2A" frameborder="0" allowfullscreen></iframe>

What is in a Scene?

The view and **View Controller** make up a scene. The view is what displays, and the View Controller is the coding arc that goes along with it. The icons at the bottom of Figure 1 represent the **First Responder** and the View Controller. The first responder is the object that receives an event such as a motion event or action message first.

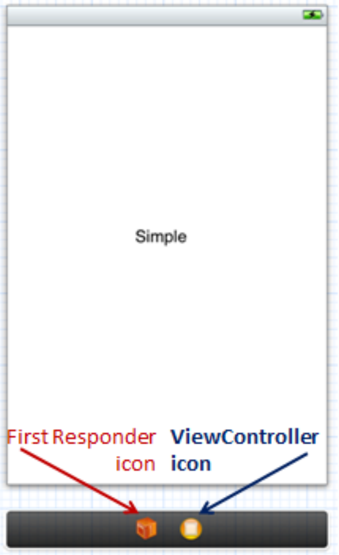


Figure 1: The View showing the First Responder and View Controller icons

The first scene in the app will have the start arrow to the left as shown in Figure 2.

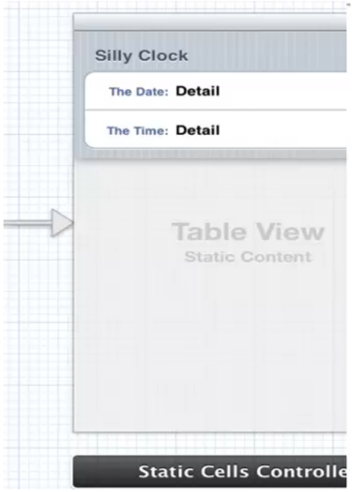


Figure 2: The first scene is indicated by the start arrow

The first scene can be changed by moving the arrow to a different view.

Zooming

The capability to zoom in to see the details of what is being worked on is available. Zoom in to edit the objects in the storyboard; zoom out to see an overview of how everything relates to each other. Objects cannot be selected or manipulated, though, when the developer zooms out.

Building a Push Segue

A **push segue** helps with passing data between views since it adds another view controller to the navigation stack. A back button is automatically added to the pushed view controller to get back to the original view controller.

To begin, start a single view controller project. Select the single view controller on the storyboard and delete it.

To build a scene, do the following:

1. Drag in a Navigation Controller from the objects explorer pane in the bottom right corner (see Figure 3). The navigation controller that was added to the screen has a table view controller as its root view controller.
2. Select and delete the table view controller.
3. Drag a standard view controller on the storyboard to replace the table view controller.
4. Hold down control and drag from the navigation controller to the new view controller.
5. Choose **root view controller** from the prompt. This makes the new view controller the root view of the navigation controller. Add any additional view controllers required to the project.

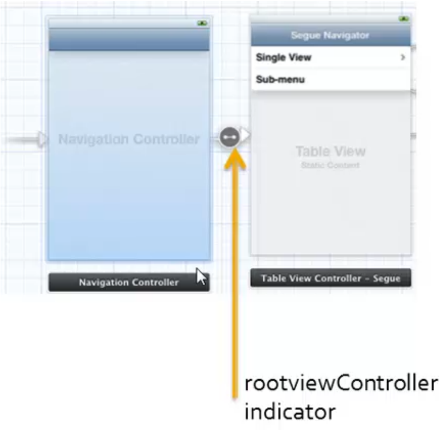


Figure 3: Building a Push Segue

A Push Segue, illustrated in Figure 3, pushes views onto a memory stack. As the user navigates, the Navigation controller pushes and pulls view controllers on and off the stack.

Remember, segues are one-way connections between scenes. There are several types of segues:

- Push segue: push is used with navigation to push onto a stack which will keep track of views
- **Modal segue**: puts one scene on top. The top scene must be dismissed to go to the original scene
- **Custom segue:** creates a transition between scenes when a push segue or modal segue will not achieve the desired effect

Read Xcode Tutorial 3: Storyboards, MapKit, View Navigation . Watch the included video, *How to Make an iPhone App – Multiple Views, Mapkit, Subclassing [Practice App #3]* (23:56) to see the concepts applied in Xcode.

Karina Whetstine 2/17/2015 2:59 PM

Comment [3]:

https://www.youtube.com/watch?v=FIkC_vGEkoQ

embed code

<iframe width="560" height="315" src="//www.youtube.com/embed/FIkC_vGEkoQ" frameborder="0" allowfullscreen></iframe>

SUMMARY

This lesson discussed storyboards (a tool for visually building views), nibs (an Interface Builder document), and segues (the connector between scenes).

ASSIGNMENTS

1. Practice – Storyboards
2. Quiz 3

Authoring Organization: Collin College
Written by: Original Author, Elizabeth Pannell; Edited Version, Susan Sands