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

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http:// www.opencontent.org/openpub/) license defines the copyright on this document.

Some Code Organization Issues

Auto generated Commented out Methods

/*

// Override to allow orientations other than the default portrait orientation.

} */

Code pragma's

#pragma mark #pragma mark Picker Data Source Methods

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





•••- AT&T 🗢 8:15	AM 92%			
	08			
	09			
0 hours	10 mins			
1	11			
2	12			
When Timer Ends Crickets >				
Start				
((\mathbf{i})			
World Clock Alarm	Stopwatch Timer			

Separate xib files for TabController Each subview

Tab bar controller handles displaying subviews

Adding More Tabs

▼ Tab Bar Con View Contro Title Blue	troller Class	0
▼ Tab Bar Con View Contro Title Blue	llers	
View Contro Title Blue	Class	
Title Blue	Class	
Blue	Class	
	View Controller	÷
Green	View Controller	\$
Yellow	View Controller	\$
Orange	View Controller	÷
Deep Red	View Controller	÷
+ -		
▼ Simulated U	ser Interface Elements	
Orientation	Portrait	\$
Status Bar	Gray	\$
Top Bar	Unspecified	\$
Bottom Bar	Tab Bar	\$
	Yellow Orange Deep Red + - ✓ Simulated U Orientation Status Bar Top Bar Bottom Bar	Yellow View Controller Orange View Controller Deep Red View Controller + - * - * Simulated User Interface Elements Orientation Portrait Status Bar Gray Top Bar Unspecified Bottom Bar Tab Bar

Red

Green

Blue

Yellow

Orange Deep Red

Linking Tab to XIB file

😝 🔿 💿 🛛 😽 Tab Bar Controller 🛛 🛥	0 0 0 Re	d View Controller Attributes	000	Tab Bar Item Attributes
		O 🤌 🛈		O 🖉 🛈
	View Contr	oller	▼ Tab Bar Ite	m
	Title	Red	Badge	5
	Layout	Wants Full Screen	Identifier	Custom
	NIB Name	RedViewController	▼ Bar Item	
		🗹 Resize Viev From NIB	Title	Red
			Image	
			Tag	0
View				🗹 Enabled
Loaded From " <u>RedViewController</u> "				
? ? ? ?				
Red Blue Green Yellow Orange Deep Red				

Three Inspector Views of Tabbar

\varTheta 🔿 🔿 Tab Bar Controller Attributes				
- -	Q	1	0	
▼ Tab Bar Con	troller			
View Contro	llers			
Title	Class			
Blue	View	Controller	÷	
Green	View	Controller	÷	
Yellow	View	Controller	÷	
Orange	View	Controller	÷	
Deep Red	View	Controller	÷	
+ -				
▼ Simulated U	ser Interface	Elements		
Orientation	Portrait		\$	
Status Bar	Gray		\$	
Top Bar	Unspecifie	ed	\$	
Bottom Bar	Tab Bar		\$	

0 0 0 R	ed View Contro	d View Controller Attributes				
.	Q	1	0			
View Cont	View Controller					
Title	Red					
Layout	Wants Full Screen					
NIB Name	RedViewController					
	🗹 Resize V	iew From NIB				

first click on tab item

000	Tab Bar Iten	n Attributes			
	0	1	0		
▼ Tab Bar Iten	1				
Badge	5				
Identifier	Custom				
▼ Bar Item					
Title	Red				
Image			-		
Tag	0				
	🗹 Enabled				

second click on tab item

first click on tab bar

command (apple) click on tab item to deselect it

More



Classes & Protocols

UITabBar

View, contains UITabBarltems

UITabBarDelegate

Used to interact with users changing tab order

UITabBarController

Contains UITabBar, delegate, subview controllers

UITabBarControllerDelegate

Selecting items Starting, stopping editing of tab order

UITabBarltem title, image, badge, tag

Tabbar & Xcode Templates

0 0	New Proj	ect					
Choose a template for you	Choose a template for your new project:						
iOS Application Library Mac OS X	Navigation-based	OpenGL ES	Split View-based				
Application Framework & Library Application Plug-in System Plug-in Other CS 696	Application Tab Bar Application Window-based Application	Application Utility Application	Application View-based Application				
	Product iPhor	Application	n application that				
Cancel Choose							

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

	IL AT&T 🤇	?	9:43 AM	98 % 🎿
<	Location) []	rbanspoor	Filter
	City Heig	ght	Hawaiian	
	Clairemo	ont	Hot Dogs	\$
	College	Area	Indian	\$\$
	Core-Co	lu	International	\$\$\$
	Coronad	io	Italian	\$\$\$\$
	<u>م</u>	}	କ	ଟ୍ର
		Shake	e to find a resta	lurant
			Shake	
		191	Æ	0 0
	Shake	Friends	Browse S	earch Nearby

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];

Pickers

Carrier 🗢 7:49 PM 🔤			
	10		
	15		
0 hours	20 mins		
1	25		
2	30		

DatePicker General Picker

DatePicker

Simple

Il Carrier 🗢 9:44	AM		
Tue Sep 28	7	42	
Wed Sep 29	8	43	
Today	9	44	AM
Fri Oct 1	10	45	РМ
Sat Oct 2	11	46	

Select

00		Library		
	Objects	Classes	Media	
🛄 Libr	ary			¢
	Scroll Vie to display the size o	• w - Provid content th f the applic	es a mechanism at is larger than cation's window.	* *
Lannajbeal Ignam deleksit	Picker Vie wheel or s values.	aw – Displa Iot-machin	ays a spinning- ne motif of	0
Jamar 1 66 Petrus 2 69 Math 3 18	Date Pick rotating w select date	er – Displa /heels to al es and time	iys multiple low users to es.	
Library -	Cocoa Tou	ch – Inputs	s & Values	
12	Segmente multiple s	eqments, e	– Displays each of which	A Y
Jamar 1 08 Pelwar 2 09 Made 3 28	Date Picke UIDatePick	er er		
Provides wheels to times. Ex and Alarr	an object th allow user amples of a n (Set Alarn	hat uses m is to select a date pick n) panes of	ultiple rotating dates and er are the Timer f the Clock	▲ ▼

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

[message release];

}

	😝 🔿 🙆 🛛 Date Picker Attributes					
	Display option		Q	1	0	
	Display option	▼ Date Picke	er			
		Mode	✓ Date & Ti	me		
Date & Time		Locale	Time Date			
Time		Interval	Timer			
Date						

Timer

Date



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
 <ur>
 <UIPickerViewDelegate, UIPickerViewDataSource> {

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

@property (nonatomic, retain) NSArray *pickerData;

- (IBAction)buttonPressed; @end

}

Carrie	r ᅙ	12:17 PM
	-	
Ca	ıt	
Do	g	
Mo	ouse	
Bi	rd	
		Select

Some Data

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

Carrier 裦	12:17 PM
Cat	
Dog	
Mouse	
Bird	
	Select

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

O O O Picker View Connections				
.	Q	1	0	
▼ Outlets				
dataSource		File's Owner	0	
delegate		File's Owner	۲	
Referencing Outlets				
New Referencing Outlet			0	

Dependent Components

Carrier 🗢	3:52 PM	-
	_	
_	_	
	_	
Cornivoro	_	Chickon
Carnivore		Chicken
Vegetarian		Bork
vegetariari	_	POIK
_	_	Dest
		Веет



Data file - food.plist

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

Creating plist

plist editor

Code

Reading food.plist

Resources food.plist FoodViewController DateView.xib SingleComponentVi MainWindow.xib

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];
        }
}
```