

CSC 471 / 371 Mobile Application Development for iOS



Prof. Xiaoping Jia
School of Computing, CDM
DePaul University
xjia@cdm.depaul.edu
 @DePaulSWEEng

1

Tabbed Views & Picker Views

2

Outline

- Tabbed views
- Picker views

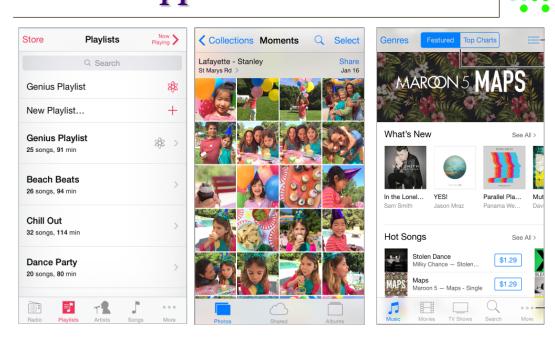


3

Tabbed Views

4

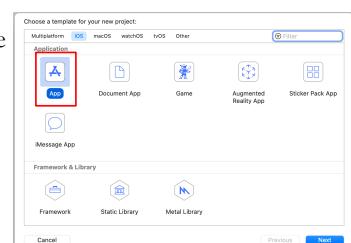
Tabbed Apps



5

Create a Tabbed App

- Start with the “App” template



6

Create a Tabbed App

- Add a Tab Bar Controller

DEPAUL UNIVERSITY

7

Create a Tabbed App

- Add a Tab Bar Controller

DEPAUL UNIVERSITY

8

Create a Tabbed App

- Add a Tab Bar Controller
- Move the Entry Point to the Tab Bar Controller

DEPAUL UNIVERSITY

9

Tabbed App

- Each tab can be customized
 - View
 - Item label and icon
- Add a view controller for each tab
- Adding new tabs
 - view controllers and
 - relationships

DEPAUL UNIVERSITY

10

Add a New Tab

- Add a new scene
 - Drag and drop a new View Controller in the storyboard
- Add a new Cocoa Touch Class
 - Class: `ThirdViewController`
 - Subclass of: `UIViewController`
 - Uncheck: *Also create XIB file*
 - Language: `Swift`
- In the storyboard
 - Select the new View Controller
 - Use the Identity Inspector to change the Class to: `ThirdViewController`

DEPAUL UNIVERSITY

11

Add a Relationship Segue

- Add a segue from the Tab Bar Controller to the new View Controller.

The Third scene

DEPAUL UNIVERSITY

13

The Storyboard with Three Tabs

- The *Third* tab is added and linked
- The tabs can be customized

The new segue
The new Third scene
The new Third tab

14

Customize the Title and Icon of the New Tab

- Select the *tab item* at the bottom of the new scene
 - Use the *Attribute Inspector*
- Option 1:
 - Use *System Items*, e.g., *featured*
- Option 2:
 - Use a *Custom* item
 - Choose a *title* for the tab item

15

Run ... the Tabbed App

- The *First* tab
- The *Featured* tab

The First View
The First tab
The Featured tab

16

“Order My Sub” App Using Pickers

17

“Order My Sub” App

- A three-tab app using pickers
- A date picker to pick date and time
- A simple one-component picker to choose the type of sub to order
- A two-component picker to choose the type and the size of the sub to order
- Using popup alerts and action sheets
 - Using closures for actions
- Sharing data between tabs

Select Type & Size
Big "A" Italian
Tuna

18

Using a Date Picker

- Start with the *App* template
- Add *Tab Bar Controller*
- Add a third tab for a *Date Picker*
 - Add new *View Controller* class: *DateViewController*
- Customize the third tab
 - Set the tab icon and title
 - Add a *Date Picker* and other controls in the view
 - Preferred Style: Wheels

Pick a Time & Date
Select

19

The “Pick a Date & Time” Scene

- Connect outlets to
 - The *Segmented Control*
 - The *Date Picker*
- Connect actions to
 - The *Segmented Control*
 - The “Select” Button

DEPAUL UNIVERSITY 16

20

The Date View Controller – the Outlets and Actions

```
class DateViewController: UIViewController {
    @IBOutlet weak var options: UISegmentedControl!
    @IBOutlet weak var datePicker: UIDatePicker!
    @IBAction func optionSelected(_ sender: UISegmentedControl) {
        ...
    }
    @IBAction func datePicked(_ sender: UIButton) {
        ...
    }
}

override func viewDidLoad() {
    super.viewDidLoad()
    options.selectedSegmentIndex = 0
}
...
}
```

Set the initial state of the *Segmented Control*

DEPAUL UNIVERSITY

17

21

Set the Date Picker Mode

```
@IBAction func optionSelected(_ sender: UISegmentedControl) {
    if sender.selectedSegmentIndex == 0 {
        datePicker.datePickerMode = .dateAndTime
    } else if sender.selectedSegmentIndex == 1 {
        datePicker.datePickerMode = .time
    }
}
```

DEPAUL UNIVERSITY 18

22

Action for Picking Date

```
@IBAction func datePicked(_ sender: UIButton) {
    let title = "Selected Time/Date"
    let message = "You have selected \(datePicker.date)"
    let alertController = UIAlertController(title: title,
                                         message: message, preferredStyle: .alert)
    let cancelAction = UIAlertAction(title: "Cancel",
                                    style: .cancel, handler: nil)
    let okayAction = UIAlertAction(title: "Confirm",
                                 style: .default, handler: nil)
    alertController.addAction(cancelAction)
    alertController.addAction(okayAction)
    present(alertController, animated: true,
            completion: nil)
}
```

DEPAUL UNIVERSITY

19

23

Select a Time and a Date

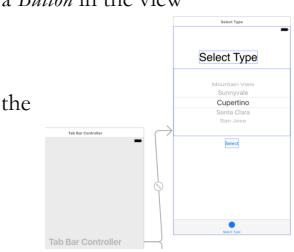
An alert popup with a *Cancel* and a *Confirm* button.

DEPAUL UNIVERSITY 20

24

The “Select Type” Scene – Single Component Picker

- Use *View Controller* class: `FirstViewController`
- Add a *Picker View* and a *Button* in the view
- Connect an outlet to the *Picker*
- Connect an actions to the “Select” Button



DEPAUL UNIVERSITY

21

25

The First View Controller

- The outlet and the action

```
class FirstViewController: UIViewController,
    UIPickerViewDataSource, UIPickerViewDelegate {
    @IBOutlet weak var picker: UIPickerView!
    @IBAction func selected(_ sender: UIButton) { ... }
}
```

DEPAUL UNIVERSITY

22

26

Populate Data in the Picker

- Use the *Delegate* pattern
 - The *Picker View* will delegate the task of populating the data to this *View Controller*.
- This *View Controller* must conform to two protocols
 - `UIPickerViewDelegate`
 - `UIPickerViewDataSource`

```
class FirstViewController: UIViewController,
    UIPickerViewDataSource, UIPickerViewDelegate {
    ...
}
```

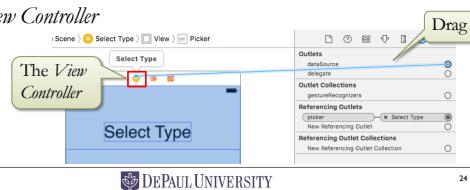
DEPAUL UNIVERSITY

23

27

Connection to the Delegate

- The connection between the *Picker View* and its *data source* and *delegate* must be explicitly set
- In storyboard, select the *Picker View*
- Select the *Connection Inspector*
- Connect the outlets named *dataSource* and *delegate* to the *View Controller*



DEPAUL UNIVERSITY

24

28

The Single Component Picker

- Populate the *Picker View* data in the *View Controller*

```
let subs = [
    "Blockbuster",
    "Roast Beef",
    ...
    "American"
]

class FirstViewController: UIViewController,
    UIPickerViewDataSource, UIPickerViewDelegate {
    ...
}
```

DEPAUL UNIVERSITY

25

29

Implement the Data Source

- The methods of the `UIPickerViewDataSource` protocol

```
func numberOfComponents(in pickerView: UIPickerView) -> Int {
    return 1
}

func pickerView(_ pickerView: UIPickerView,
               numberOfRowsInComponent component: Int)
               -> Int {
    return subs.count
}
```

DEPAUL UNIVERSITY

26

30

Implement the Delegate

- The methods for the `UIPickerViewDelegate` protocol

```
func pickerView(_ pickerView: UIPickerView,
               titleForRow row: Int,
               forComponent component: Int)
               -> String? {
    return subs[row]
}
```

DEPAUL UNIVERSITY

27

31

The Select Action

```
@IBAction func selected(_ sender: UIButton) {
    let title = "Order My Sub"
    let message = "You have selected
    \n(subs[picker.selectedRow(inComponent: 0)])"
    let alertController = UIAlertController(title: title,
                                           message: message, preferredStyle: .alert)
    let cancelAction = UIAlertAction(title: "Cancel",
                                    style: .destructive, handler: nil)
    let okayAction = UIAlertAction(title: "Confirm",
                                    style: .default, handler: nil)
    alertController.addAction(cancelAction)
    alertController.addAction(okayAction)
    present(alertController, animated: true, completion: nil)
}
```

DEPAUL UNIVERSITY 28

32

The Test Run ...

DEPAUL UNIVERSITY 29

34

The “Select Type & Size” Scene – Double Component Picker

- Use View Controller class: `SecondViewController`
- Add a `Picker View` and a `Button` in the view
- Connect an outlet to the `Picker`
- Connect an actions to the “Select” Button
- Connect the `dataSource` and the `delegate` of the `Picker View`

DEPAUL UNIVERSITY 30

35

The Second View Controller

- The outlet and the action
- This View Controller also conforms to two protocols
 - `UIPickerViewDelegate`
 - `UIPickerViewDataSource`

```
class SecondViewController: UIViewController, UIPickerViewDataSource, UIPickerViewDelegate {

    @IBOutlet weak var picker: UIPickerView!
    @IBAction func selected(_ sender: UIButton) { ... }

    ...
}
```

DEPAUL UNIVERSITY 31

36

The Double Component Picker

- Populate the `Picker View` data in the `View Controller`

```
let sizes = [
    "6 inch",
    "8 inch",
    "10 inch",
    "12 inch",
    "16 inch",
    "3 foot"
]
class SecondViewController: UIViewController, UIPickerViewDataSource, UIPickerViewDelegate {
    ...
}
```

An array of strings used to populate the second component of the Picker View.

DEPAUL UNIVERSITY 32

37

Implement the Data Source

- The `UIPickerViewDataSource` protocol

```
func numberOfComponents(in pickerView: UIPickerView) -> Int {
    return 2
}
func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
    if component == 0 {
        return subs.count
    } else {
        return sizes.count
    }
}
```

Number of rows in component 0
Number of rows in component 1

DEPAUL UNIVERSITY 33

38

Implement the Delegate

- The `UIPickerViewDelegate` protocol

```
func pickerView(_ pickerView: UIPickerView,
               titleForRow row: Int,
               forComponent component: Int)
    -> String? {
    if component == 0 {
        return subs[row] // For component 0
    } else {
        return sizes[row] // For component 1
    }
}
```

Select Type & Size

Italian Special Corned Beef Big "AI" Italian Tuna White Guy	6 inch 8 inch 10 inch 12 inch 16 inch
C. 0	C. 1

DEPAUL UNIVERSITY

34

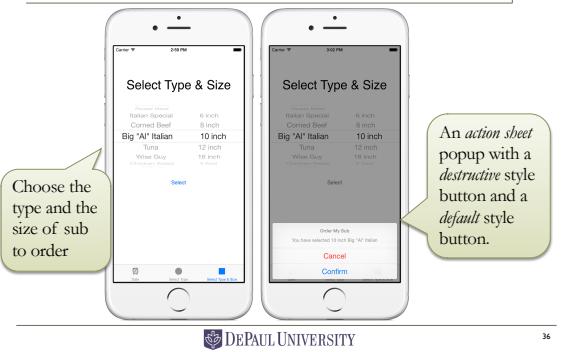
The Select Action

```
@IBAction func selected(_ sender: UIButton) {
    let title = "Order My Sub"
    let message = "You have selected"
    let sizes[picker.selectedRow(inComponent: 1)]
    let subs[picker.selectedRow(inComponent: 0)]
    let alertController = UIAlertController(title: title,
                                           message: message, preferredStyle: .actionSheet)
    let cancelAction = UIAlertAction(title: "Cancel",
                                    style: .destructive) { action in ... } // Action for Cancel
    let confirmAction = UIAlertAction(title: "Confirm",
                                    style: .default) { action in ... } // Action for Confirm
    alertController.addAction(cancelAction)
    alertController.addAction(confirmAction)
    present(alertController, animated: true, completion: nil)
}
```

DEPAUL UNIVERSITY

35

The Action Sheet Popup



DEPAUL UNIVERSITY

36

The Select – Cancel Action

```
@IBAction func selected(_ sender: UIButton) {
    ...
    let cancelAction = UIAlertAction(title: "Cancel",
                                    style: .destructive) { action in
        let alertController = UIAlertController(
            title: "No Problem", message: "Select again.",
            preferredStyle: .alert)
        let okayAction = UIAlertAction(title: "Okay",
                                      style: .default, handler: nil)
        alertController.addAction(okayAction)
        self.present(alertController,
                     animated: true, completion: nil)
    }
    ...
}
```

DEPAUL UNIVERSITY

37

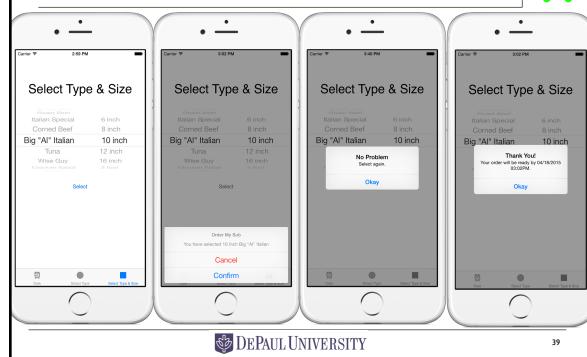
The Select – Confirm Action

```
let confirmAction = UIAlertAction(title: "Confirm",
                                style: .default) { action in
    let okayController = UIAlertController(
        title: "Thank You!", message: "Your order ...",
        preferredStyle: .alert)
    let okayAction = UIAlertAction(title: "Okay",
                                  style: .default, handler: nil)
    okayController.addAction(okayAction)
    self.present(okayController,
                 animated: true, completion: nil)
}
alertController.addAction(cancelAction)
alertController.addAction(confirmAction)
...
}
```

DEPAUL UNIVERSITY

38

The Cancel and Conform Popup



DEPAUL UNIVERSITY

39

One More Refinement

- Use the date/time selected at “Pick a Time & Data” tab as the pick-up time.
- Format the confirm message of the “Pick Type & Size” tab using the pick-up time.
- To pass data between tabs, we use a global variable

```
var date = Date()  
A global variable  
class DateViewController: UIViewController {  
    ...  
}
```

DEPAUL UNIVERSITY

49

Pick a Time and Date – Handle Confirm Action

```
@IBAction func datePicked(_ sender: UIButton) {  
    let title = "Selected Time/Date"  
    let message = "You have selected \(datePicker.date)"  
    let alertController = UIAlertController(title: title,  
                                         message: message, preferredStyle: .alert)  
    let cancelAction = UIAlertAction(title: "Cancel",  
                                   style: .cancel, handler: nil)  
    let okayAction = UIAlertAction(title: "Confirm",  
                                 style: .default) {  
        _ in date = self.datePicker.date  
    }  
    alertController.addAction(cancelAction)  
    alertController.addAction(okayAction)  
    present(alertController, animated: true,  
           completion: nil)  
}
```

50

Pick Type & Size – The Select – Confirm Action

```
let confirmAction = UIAlertAction(title: "Confirm",  
                                style: .default) { action in  
    let dateFormatter = DateFormatter()  
    dateFormatter.dateFormat = "MM/dd/yyyy hh:mm"  
    let dateString = dateFormatter.string(from: date)  
    let okayController = UIAlertController(  
        title: "Thank You!",  
        message: "Your order will be ready by \(dateString).",  
        preferredStyle: .Alert)  
    ...  
}
```



DEPAUL UNIVERSITY

51

Sample Code

- Tabbed App – SB.zip
- Order My Sub – SB.zip

DEPAUL UNIVERSITY

43

Next ...

- Static table views
- Dynamic table views
- Navigation views

◊ iOS is a trademark of Apple Inc.

DEPAUL UNIVERSITY

44

53