Intro to iOS Development

Lecture 3

Navigation, MVC, and Delegation

Mathew Scullin
Lucy Xu

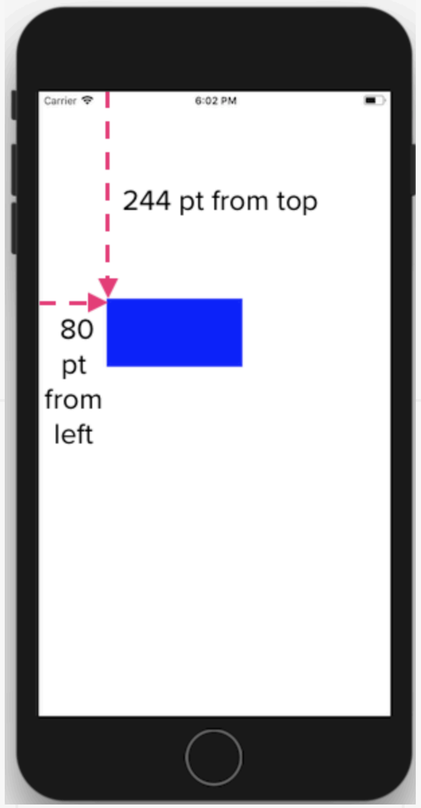
Resource for AutoLayout

- WTF AutoLayout www.wtfautolayout.com

```
    Probably at least one of the constraints in the following list is one you don't want.
    Try this:
        (1) look at each constraint and try to figure out which you don't expect;
        (2) find the code that added the unwanted constraint or constraints and fix it.
(
    "<NSLayoutConstraint:0x600003b770c0 UITextField:0x7fd131857c00.centerX == UIView:0x7fd130f0ae30.centerX + 130
        (active)>",
    "<NSLayoutConstraint:0x600003b77110 H:|-(0)-[UITextField:0x7fd131857c00]   (active, names: '|':UIView:
        0x7fd130f0ae30 )>",
    "<NSLayoutConstraint:0x600003b77160 UITextField:0x7fd131857c00.trailing == UIView:0x7fd130f0ae30.trailing   (active)>"
)

Will attempt to recover by breaking constraint
<NSLayoutConstraint:0x600003b770c0 UITextField:0x7fd131857c00.centerX == UIView:0x7fd130f0ae30.centerX + 130   (active)>
```

Some other ways of doing UI in iOS

Frame-based

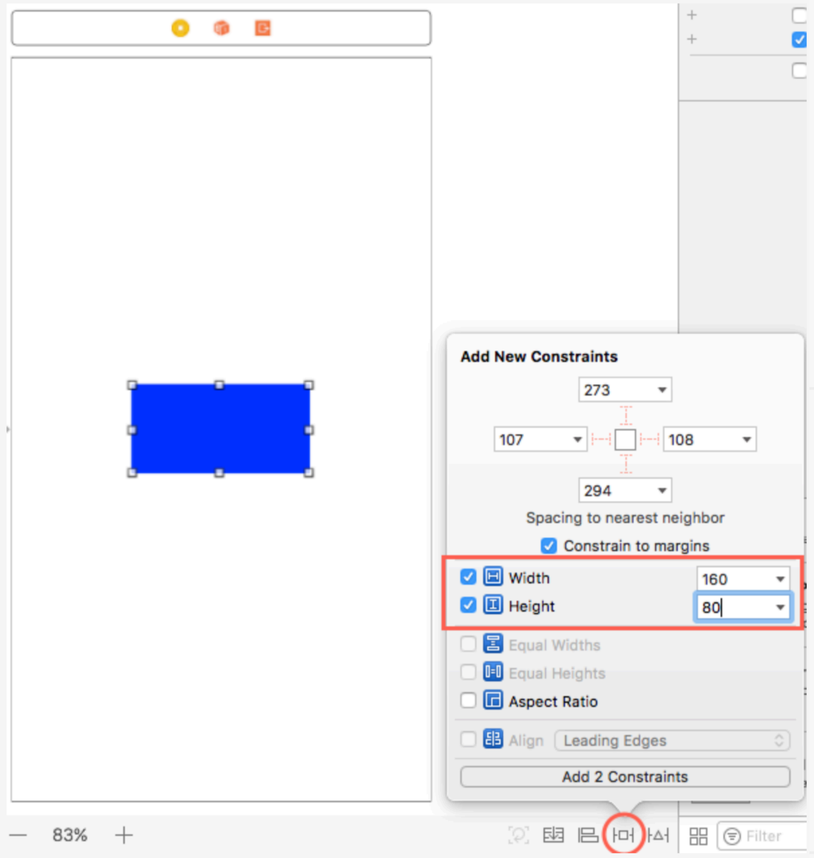



```
let blueView = UIView(frame:
    CGRect(x: 80.0, y: 244.0, width: 160.0, height: 80.0))
blueView.backgroundColor = .blue
view.addSubview(blueView)
```

SwiftUI

```
//
//  Content.swift
//

import SwiftUI

struct Content : View {

    @State var model = Themes.listModel

    var body: some View {
        List(model.items, action: model.selectItem) { item in
            Image(item.image)
            VStack(alignment: .leading) {
                Text(item.title)
                Text(item.subtitle)
                    .color(.gray)
            }
        }
    }
}
```

Storyboards



Announcements

- OH Calendar new system.
- Do not email us! Please post a private post on Piazza visible to all instructors
- Grades for P1 will be made available later this week.

Review

How would you center a UIView in the center of the screen?

A. `myView.centerXAnchor.equalTo(view.centerXAnchor)`

B. `myView.leadingAnchor.equalTo(view.leadingAnchor)`

C. `myView.centerYAnchor.equalTo(view.centerYAnchor)`

D. `A + B`

E. `A + C`

How would you center a UIView in the center of the screen?

A. `myView.centerXAnchor.equalTo(view.centerXAnchor)`

B. `myView.leadingAnchor.equalTo(view.leadingAnchor)`

C. `myView.centerYAnchor.equalTo(view.centerYAnchor)`

D. `A + B`

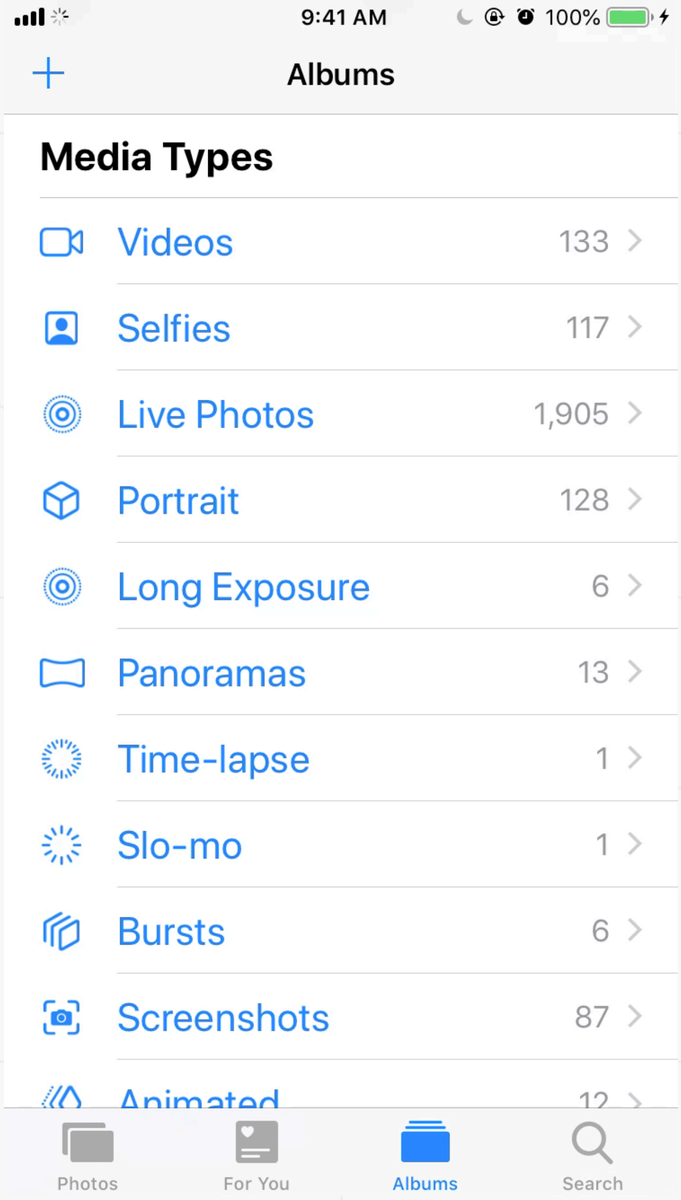
E. `A + C`

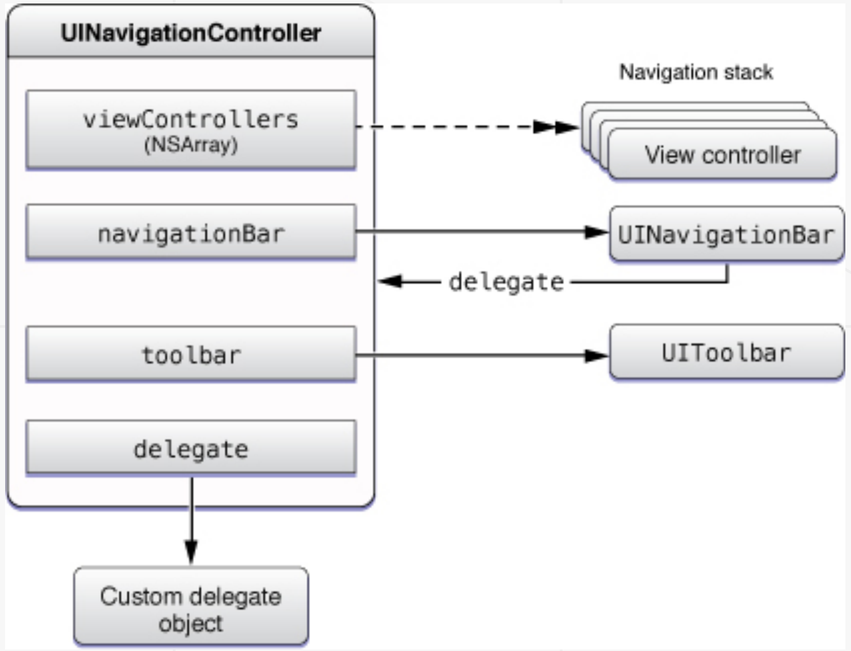
How do you create multi-view apps?

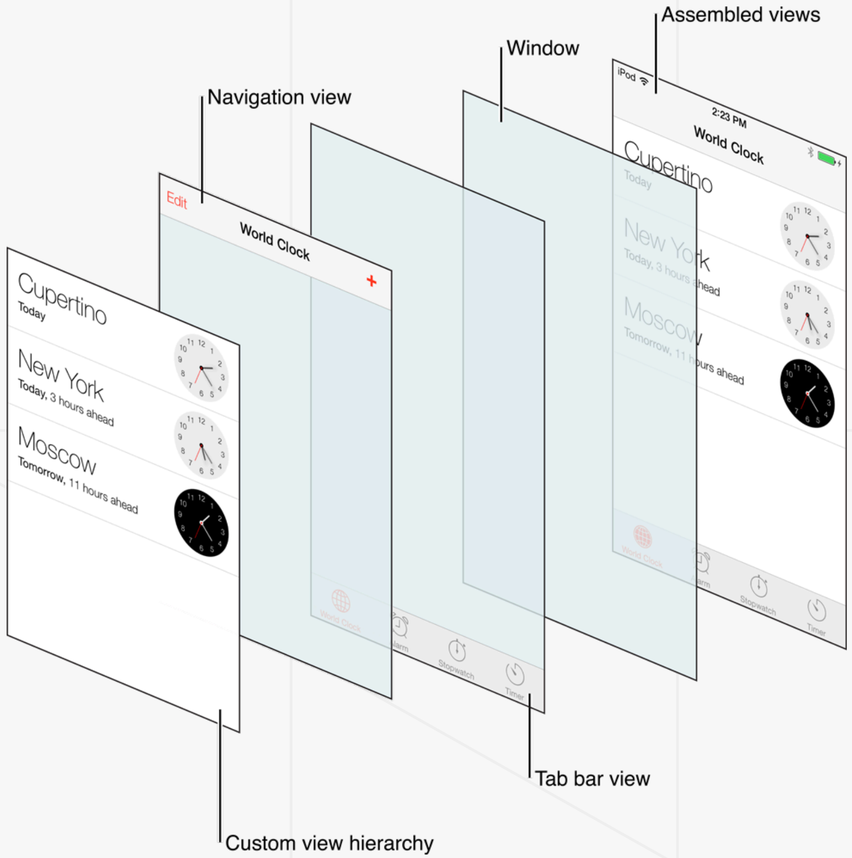
UINavigationController

1. Push new UIViewController onto navigation stack
2. Pop UIViewController off navigation stack

```
let myViewController = ViewController()
navigationController?.pushViewController(myViewController, animated: true)
navigationController?.popViewController(animated: true)
```


9:41 AM
100%
Albums
Media Types
Videos 133
Selfies 117
Live Photos 1,905
Portrait 128
Long Exposure 6
Panoramas 13
Time-lapse 1
Slo-mo 1
Bursts 6
Screenshots 87
Animated 12
Photos
For You
Albums
Search



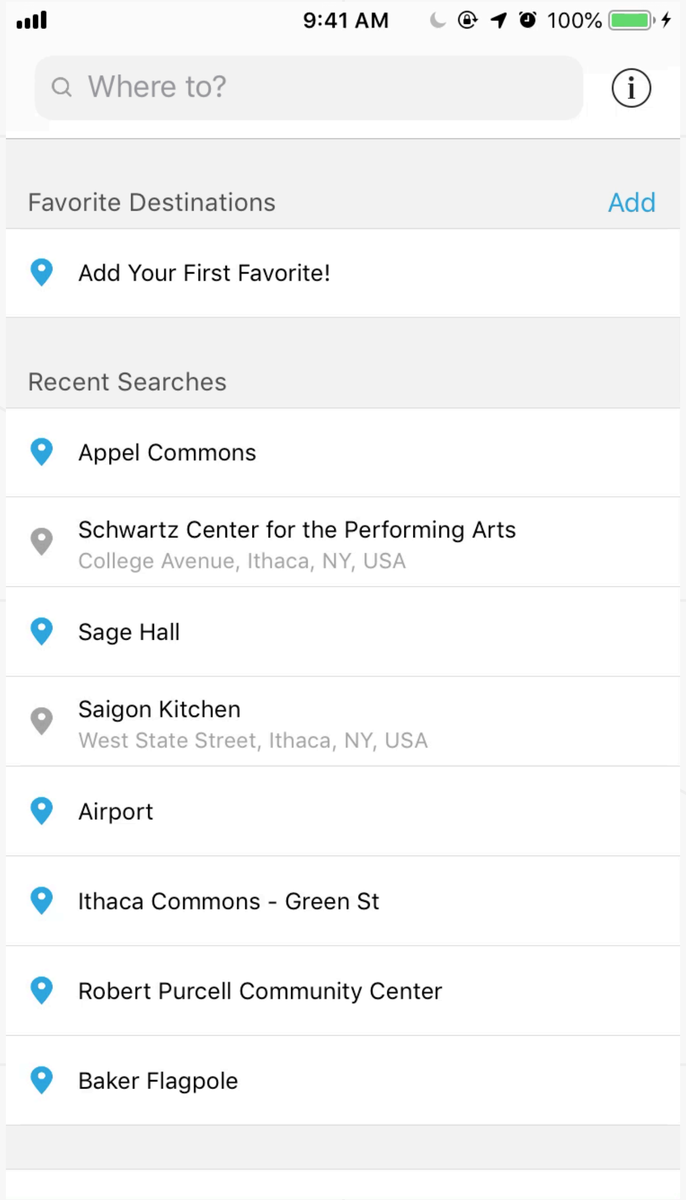


 Source: developer.apple.com

Modal Presentation

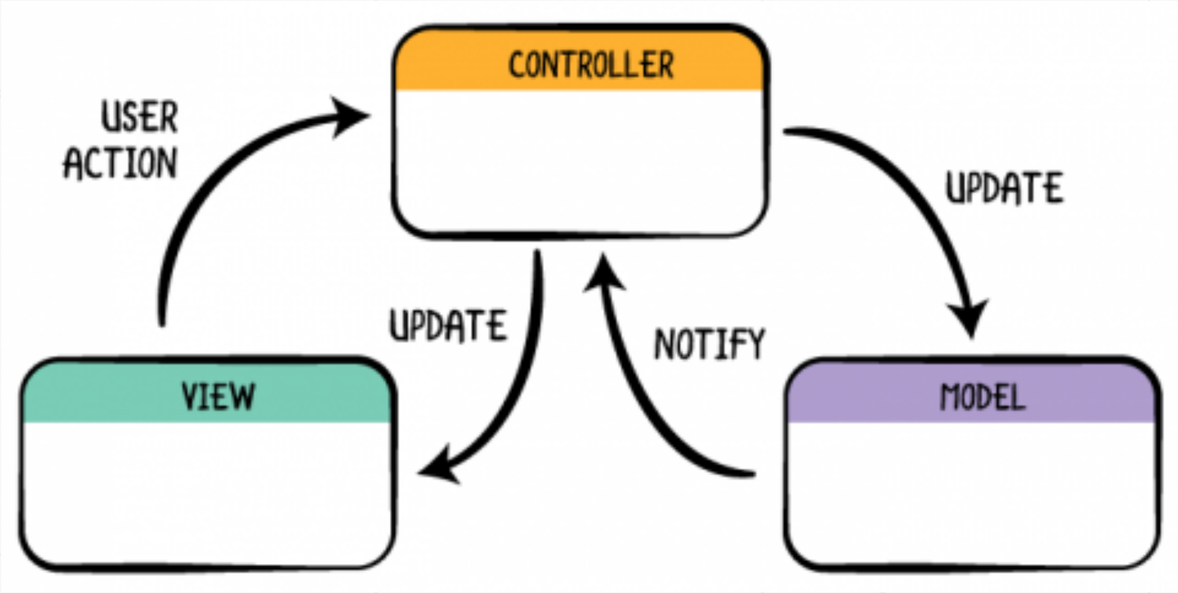
- Directly present new UIViewController on top of previous one
- No navigation stack

```
let myViewController = ViewController()

present(myViewController, animated: true, completion: nil)

dismiss(animated: true, completion: nil)
```


9:41 AM
100%
Where to?
Favorite Destinations
Add
Add Your First Favorite!
Recent Searches
Appel Commons
Schwartz Center for the Performing Arts
College Avenue, Ithaca, NY, USA
Sage Hall
Saigon Kitchen
West State Street, Ithaca, NY, USA
Airport
Ithaca Commons - Green St
Robert Purcell Community Center
Baker Flagpole

Model View Controller

- Common design pattern
- Controller receives information from view and model
- Hence, UIViewController



Example: Ithaca Transit

- **Model:** BusStop objects
- **View:** Displays list of all bus stops
- **Controller:** Listens to user selection to update the view, updates the list of BusStops to display

Delegation

- Pass information between view controllers
- Communication method is specified through **protocols**
- One object can **delegate** another object to perform an action through the protocol
- Protocols are like interfaces in Java

Restaurant Analogy

- Chef, customer, menu
- Chef: knows how to cook items on the menu
- Customer: orders items off the menu
- Menu: is a **protocol**
- Customer **delegates** chef to make their food because they know how to cook the menu items

```
protocol MenuDelegate: class {
    func makeBurrito()
}
```

```
class Chef: MenuDelegate {
    init() { }
    func makeBurrito() {
      print(“Making the Burrito!”)
    }
}
```

```
class Customer {
    var delegate: MenuDelegate
    init(delegate: MenuDelegate) { … }
    func orderBurrito() {
      delegate.makeBurrito()
    }
}
```

```
protocol MenuDelegate: class {
    func makeBurrito()
}
```

```
class Chef: MenuDelegate {
    init() { }
    func makeBurrito() {
      print(“Making the Burrito!”)
    }
}
```

```
class Customer {
    var delegate: MenuDelegate
    init(delegate: MenuDelegate) { … }
    func orderBurrito() {
      delegate.makeBurrito()
    }
}
```

What happens?

```
let chef = Chef()
let customer = Customer(delegate: chef)
customer.orderBurrito()
```

Protocols

```
protocol ChangeColorDelegate: class {
    func didChangeColor(to newColor: UIColor)
}

extension ViewController: ChangeColorDelegate {
    func didChangeColor(to newColor: UIColor) {
        label.backgroundColor = newColor
    }
}
```

Protocols

```
class ViewController: UIViewController {
. . .
    @objc func presentViewController() {
     let otherViewController = OtherViewController()
     otherViewController.delegate = self
     present(otherViewController, animated: true)
    }
}
```

Delegates

Before we present `OtherViewController`, we set the delegate to `ViewController`

This method calls the `didChangeColor` method in `ViewController`

```
class OtherViewController: UIViewController {
 weak var delegate: ChangeColorDelegate?
 …
 func tellViewControllerToChangeColor() {
  delegate?.didChangeColor(to: .red)
 }
}
```

Action Items

- Project 3 due 10/11 11:59pm
- New OH system.