

Mobile Apps Development for Journalists

Week 13, Tuesday - Fall 2016
University of Texas at Austin
School of Journalism

Jeff Linwood
jlinwood@gmail.com
@jefflinwood

Agenda

- Working with Locations - Hand Back Assignments
- Custom Classes
- Implementing MKAnnotation
- MKMapViewDelegate
- Showing a Detail Screen when you click on a pin

Working with Locations

- Great job, everyone!

Custom Classes

- We've built custom classes for our view controllers
- Can also build custom classes for our data
- Very useful for map applications
- Can have almost anything

Sample Custom Class

```
import UIKit

import CoreLocation
import MapKit

class StateCapital: NSObject, MKAnnotation {
    var title:String? = ""
    var coordinate:CLLocationCoordinate2D = CLLocationCoordinate2DMake(0,0)
    var websiteURL:String = ""
    var stateName:String = ""
    var population:Integer = 0
}
```

Implementing MKAnnotation

- Very important to import CoreLocation and MapKit
- Extend NSObject
- Implement MKAnnotation
- Create properties for title and coordinate (possibly subtitle)
- <https://developer.apple.com/reference/mapkit/mkannotation>

MKMapViewDelegate

- Many different things we can do to customize map behavior
- <https://developer.apple.com/reference/mapkit/mkmapviewdelegate>
- Annotation Views
- Overlays
- Tracking User Location

Using the MapView Delegate

- Make sure your view controller (for the screen that holds the map) imports MapKit
- Implement the MKMapViewDelegate protocol
- Use the storyboard to select the map, then pick the last tab on the right (Outlet Inspector)
- Drag a connection between the delegate and the view controller in the outline

Showing Detail Screen

```
// MARK: - Map View Delegate methods
```

```
func mapView(mapView: MKMapView, didSelectAnnotationView view: MKAnnotationView) {  
    let capital = view.annotation as! StateCapital  
    let vc = storyboard.instantiateInitialViewController() as! StateCapitalViewController  
    vc.capital = capital  
    self.navigationController?.pushViewController(vc, animated: true)  
}
```

Inside Detail Screen

- Similar to Segue
- We do all the work in `viewDidLoad`
- All of the data that we need is on the custom class