

CS5530

Mobile/Wireless Systems

Map and Flyover

Yanyan Zhuang

Department of Computer Science

<http://www.cs.uccs.edu/~yzhuang>

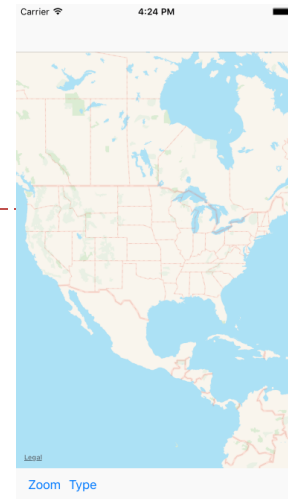
Overview

- MapKit and MKMapView
- MapKit Flyover
- Announcement
 - Mac OS and Xcode upgraded
 - I will be away March 13-15



MapKit Framework

- MapKit Framework
 - Based on Apple Maps and APIs
 - A simple mechanism for integrating mapping capabilities into any app
 - Core element: the MKMapView class
 - MKMapViewDelegate protocol
 - ▶ Allows an app to receive notifications of events relating to map view
 - ▶ Change in either the location of the user or region of the map displayed



Creating a Map

- Presenting latitude/longitude in forms of map/satellite images
- MKMapView Instance and Toolbar
 - Map Kit View object
 - Toolbar + Bar Button
 - ▶ Zoom in and change map type
 - `import MapKit`



Creating a Map

- In ViewController.swift
 - Control-drag: Map Kit View and two Bar Buttons

```
import UIKit
import MapKit

class ViewController: UIViewController, MKMapViewDelegate {
    @IBOutlet weak var mapView: MKMapView!

    override func viewDidLoad() {
        super.viewDidLoad()
    }
    @IBAction func zoomIn(_ sender: Any) {
    }
    @IBAction func changeMapType(_ sender: Any) {
    }
    ...
}
```



Obtaining Location Permission

- In AppDelegate.swift (Don't forget Info.plist)

```
import UIKit
```

```
import CoreLocation
```

```
@UIApplicationMain class AppDelegate: UIResponder, UIApplicationDelegate {
```

```
    var window: UIWindow?
```

```
    var locationManager: CLLocationManager?
```

```
    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
```

```
        locationManager = CLLocationManager()
```

```
        locationManager?.requestWhenInUseAuthorization()
```

```
        return true
```

```
    }
```

```
.. }
```

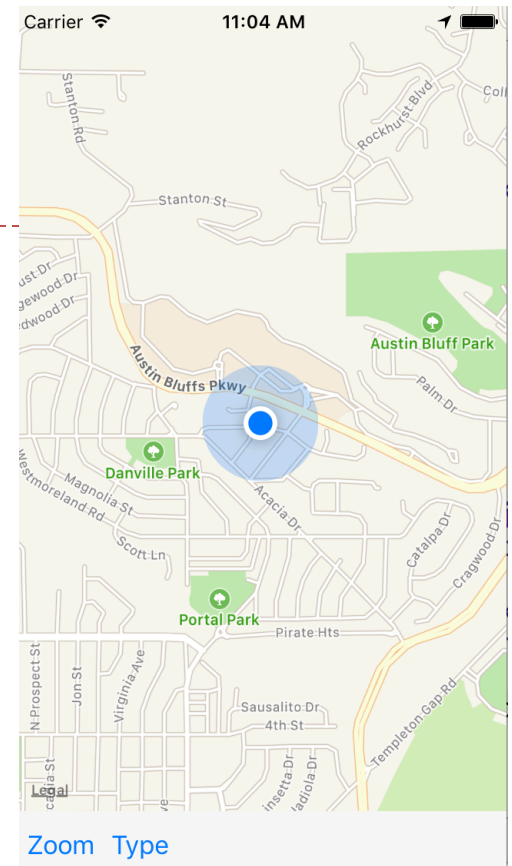


Configuring the Map View

- In ViewController.swift again

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    mapView.showsUserLocation = true  
}
```

```
@IBAction func zoomIn(_ sender: Any) {  
    let userLocation = mapView.userLocation  
    let region = MKCoordinateRegionMakeWithDistance(  
        userLocation.location!.coordinate, 2000, 2000)  
    mapView.setRegion(region, animated: true)  
}
```



Changing the Map Type

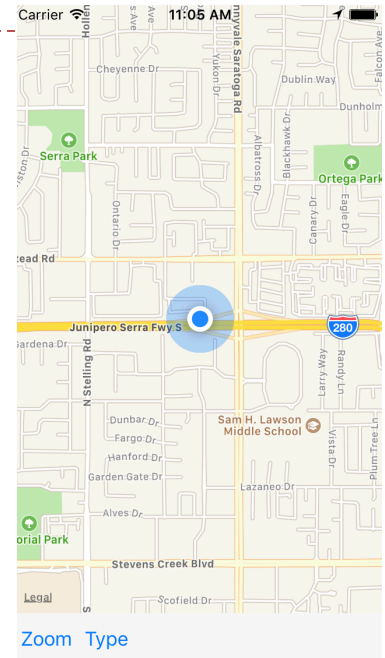
- The map type of a map view is controlled by its `mapType` property
- Supported values for property
 - `MKMapType.standard`, `MKMapType.satellite`, `MKMapType.hybrid`
 - `MKMapType.satelliteFlyover`, `MKMapType.hybridFlyover`
- In `ViewController.swift`

```
@IBAction func changeMapType(_ sender: Any) {  
    if mapView.mapType == MKMapType.standard {  
        mapView.mapType = MKMapType.satellite  
    } else {  
        mapView.mapType = MKMapType.standard  
    }  
}
```



Updating Map based on User Movement

- In `ViewController.swift`
override func `viewDidLoad()` {
 `super.viewDidLoad()`
 `mapView.showsUserLocation = true`
 `mapView.delegate = self`
}

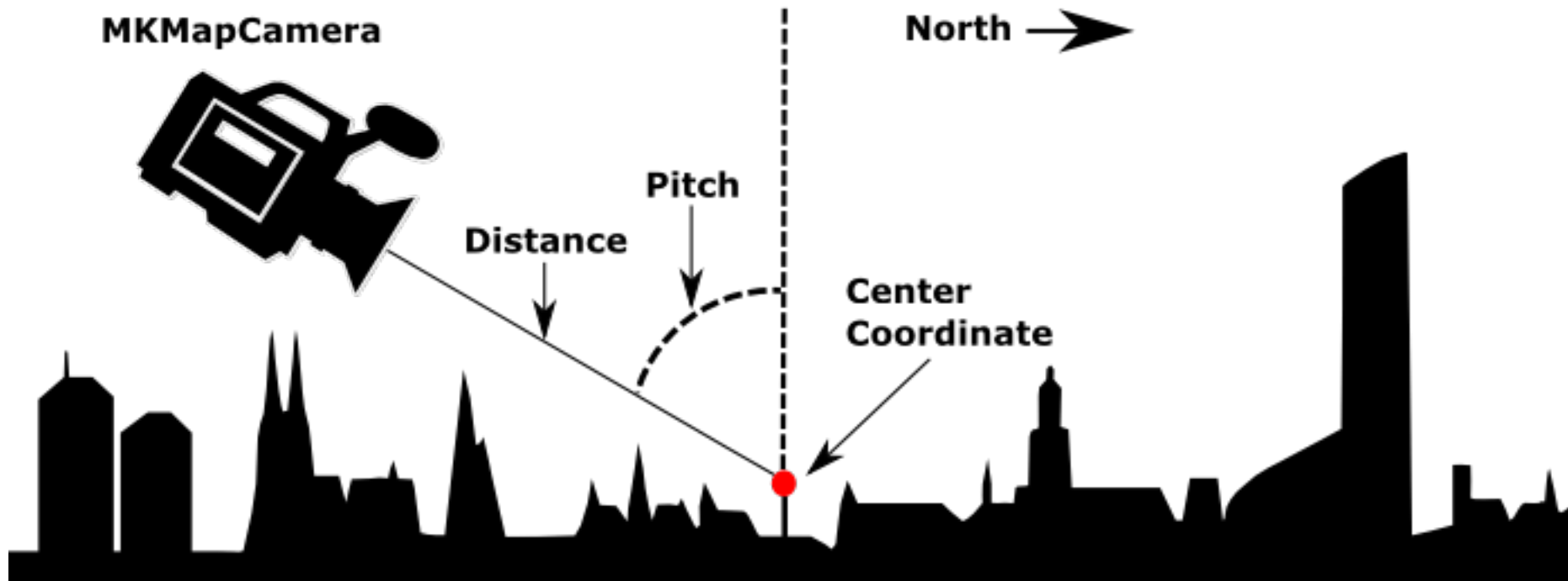


```
func mapView(_ mapView: MKMapView, didUpdate userLocation:
MKUserLocation) {
    mapView.centerCoordinate = userLocation.location!.coordinate
}
```



MapKit Flyover

- MKMapCamera Class



MapKit Flyover

- Configures an MKMapCamera instance
 - Located 650 meters from the Empire State Building
 - Facing north at a pitch of 30 degrees

let distance: CLLocationDistance = 650

let pitch: CGFloat = 30

let heading = 90.0

mapView.mapType = .satelliteFlyover

let coordinate = CLLocationCoordinate2D(latitude: 40.7484405,
longitude: -73.9856644)

camera = MKMapCamera(lookingAtCenter: coordinate,
fromDistance: distance, pitch: pitch, heading: heading)

mapView.camera = camera



MapKit Flyover

- Use a button to trigger animation

```
@IBAction func animateCamera(_ sender: AnyObject) {  
    UIView.animate(withDuration: 20.0, animations: {  
        self.camera!.heading += 180  
        self.camera!.pitch = 25  
        self.mapView.camera = self.camera!  
    })  
}
```



MapKit Flyover

- Center point: Empire State Building
 - let coordinate = CLLocationCoordinate2D(latitude: 40.7484405, longitude: -73.9856644)

