

ATLS 4120/5120: Mobile Application Development

Week 7: Core location and Map Kit

Location

Create a new Single view app called Location for the iPhone.

Click on the Target and go into the Build Phases tab.

Open Link Binary with Libraries and click the + to add the CoreLocation and MapKit frameworks under iOS 8.4.

Basic

To use these frameworks you must import them into ViewController.swift

```
import MapKit
import CoreLocation
```

Now go into MainStoryboard and add a Map View to take up the whole view. Pin the leading, trailing, top and bottom so it fills up the view.

Create an outlet connection for it called mapView.

ViewController.swift

```
override func viewDidLoad() {
    let location = CLLocationCoordinate2D(
        latitude: 40.74836,
        longitude: -73.984607
    )
    let span = MKCoordinateSpanMake(0.05, 0.05) //defines the area
    spanned by a map region
    let region = MKCoordinateRegion(location, span) //region of the map
    to be displayed
    mapView.setRegion(region, animated: true) //animates changing the
    currently visible region
    let annotation = MKPointAnnotation() //create an annotation
    annotation.coordinate=location //sets the coordinates of the
    annotation
    annotation.title="Empire State Building" //sets the title of the
    annotation
    annotation.subtitle="New York" //sets the subtitle of the annotation
    mapView.addAnnotation(annotation) //adds the annotation to the map
    view
    super.viewDidLoad()
}
```

You will see the map (default mapType) and the annotation when you run it but it's set to only that location.

Let's update it so it uses your location.

Snapshot: basic map and annotation

Track location

In Supporting Files you need to add a row to Info.plist. Right click on "Information Property List", click Add Row, and manually enter this key.

NSLocationWhenInUseUsageDescription with the string "This application requires location services"

ViewController.swift

Adopt CLLocationManagerDelegate

```
class ViewController: UIViewController, CLLocationManagerDelegate,
```

Define a CLLocationManager instance.

```
var locationManager = CLLocationManager()
```

Add CLLocationManager Delegate method

```
func locationManager(manager: CLLocationManager!, didUpdateLocations
locations: [AnyObject]!)
```

Move from viewDidLoad the lines for span, region, and setRegion (change to

MKCoordinateRegionMake and update parameters)

```
//called when a new location value is available
func locationManager(manager: CLLocationManager!, didUpdateLocations
locations: [AnyObject]!) {
    var span = MKCoordinateSpanMake(0.05, 0.05) //defines the area
spanned by a map region
    var region = MKCoordinateRegionMake(manager.location.coordinate,
span) //region of the map to be displayed
    mapView.setRegion(region, animated: true) //animates changing the
currently visible region
}
```

Remove from viewDidLoad the definition for location and comment out the annotation.

Add to viewDidLoad

```
mapView.mapType=MKMapType.Hybrid //hybrid with map and satellite
var status:CLAuthorizationStatus =
CLLocationManager.authorizationStatus()
if status==CLAuthorizationStatus.NotDetermined{
    locationManager.requestWhenInUseAuthorization() //ios8 and
later only
}
locationManager.delegate=self
locationManager.desiredAccuracy=kCLLocationAccuracyBest //specify
the desired accuracy
locationManager.distanceFilter=kCLDistanceFilterNone //specify the
distance a device must move laterally(in meters) to generate an update. We
specify to be notified of all movements
mapView.showsUserLocation=true
```

Add these delegate methods as well:

```
//called when the authorization status for the application changed.
func locationManager(manager: CLLocationManager!,
didChangeAuthorizationStatus status: CLAuthorizationStatus) {
    println("didchangeauth")
    if status==CLAuthorizationStatus.AuthorizedWhenInUse {
        locationManager.startUpdatingLocation() //starts the location
manager
    }
}
```

```

    }

    //called when a location cannot be determined
    func locationManager(manager: CLLocationManager!, didFailWithError
error: NSError!) {
        var errorType=String()
        if let clError=NSError(rawValue: error.code) {
            if clError == .Denied {
                errorType="access denied"
                let alert=UIAlertController(title: "Error", message:
errorType, preferredStyle: UIAlertControllerStyle.Alert)
                let okAction:UIAlertAction=UIAlertAction(title: "OK",
style:UIAlertActionStyle.Default, handler: nil)
                alert.addAction(okAction)
                presentViewController(alert, animated: true, completion:
nil)
            }
        }
    }
}

```

Save and Run. The simulator will not know your actual location. Either run it on a device or use Debug | Location | Custom to enter in your own longitude and latitude. Try None to see the error alert. Address lookup web site <http://stevemorse.org/jcal/latlon.php> Once you've allow or not allowed location services in the simulator you can't change it. To test you must "Reset Content and Settings" for you to be prompted again.

Snapshot: map location 2

Annotation

Now let's use annotations to add markers to the map to represent our location.

```
var annotation = MKPointAnnotation()
```

update locationManager(, didUpdateLocations) to add the annotation (or uncomment what you had previously)

```

        annotation.coordinate=manager.location.coordinate
        annotation.title="You are here"
        annotation.subtitle="Latitude:
\"(manager.location.coordinate.latitude), Longitude:
\"(manager.location.coordinate.longitude)"
        mapView.addAnnotation(annotation)

```

A tap on the pushpin will display the title and subtitle.

On a device as you move around the pushpin will move with you.

Don't forget to add app icons and a launch screen

Snapshot: final

Changing Location Service Permissions

Users can grant or revoke location permission via the Settings app. You can test this on the simulator. Launch the app and grant yourself permission to use Core Location (if you've previously denied permission, you'll need to remove and reinstall the app first). You should see your location on the map. Now go to the Settings app and choose Privacy | Location. Turn the location services switch to OFF and go back to your application. You'll see that the map no longer shows your position. That's because the location manager called the `locationManager(_, didChangeAuthorizationStatus)` method with authorization code `CLAuthorizationStatus.Denied` so the application stops receiving position updates and tells Map Kit to stop tracking the user's position.

Now go back to the Settings app, turn location services back on and run your app again and you'll find that it's tracking your position again.

You can also switch Location Services off per app. Back in Settings Privacy | Location below the location services switch you'll see a list of all the apps that are using it, including your app. Clicking the application name takes you to another page where you can allow or deny access to your application. At the moment, the application can use location services while the user is using the app. If you click Never, that permission is revoked.

Note: You might need to go into your app's target and in the Capabilities tab turn Maps on.