

CSC 491 / 391
Mobile Application
Development for iOS II



Prof. Xiaoping Jia
 School of Computing, CDM
 DePaul University
xjia@cdm.depaul.edu
[@DePaulSWEEng](https://twitter.com/DePaulSWEEng)



Outline

- Location-Aware Apps
- Map Kit



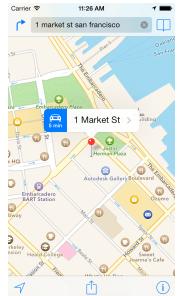
DEPAUL UNIVERSITY

Location-Aware Applications



Location Based Information

- Location services
 - User's location and heading
 - GPS and A-GPS
 - iBeacon
- Map Kit framework
 - Display maps with annotations
 - Support navigation and contextual data
 - Customizable

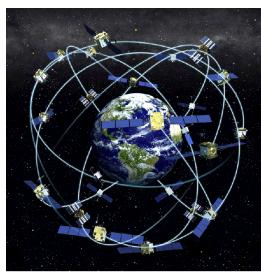


DEPAUL UNIVERSITY

GPS – Global Positioning System



- A *Satellite Navigation System* funded by U.S. DoD.
- Operated by the U.S. military, available for civilian uses global wide, free.
- Horizontal accuracy
 - Civilian uses: ≈ 3.5 m
 - Same signals for military uses.
 - No degradation.
 - Can be affected by surroundings
- Military uses:
 - Dual frequencies to correct signal distortion

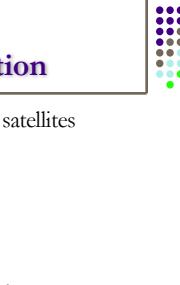


Credit: NOAA

DEPAUL UNIVERSITY

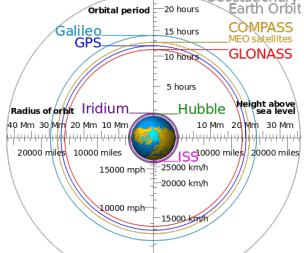
GPS Satellite Constellation

- A full constellation: minimum 24 satellites
 - Baseline expanded to 27 since 2011
 - As of 2017, 31 satellites in service
- Six equally spaced orbital planes
 - 60° apart, 4 satellite each
 - At 55° of inclination
 - the angle between the orbit and the Earth equator
 - Orbit height: $\approx 21,150$ km, in Medium Earth Orbit (MEO)
 - Orbital period: 11 hours 58 minutes. Repeat cycle: 1 day.
- Minimum 6 in line of sight anywhere on Earth anytime



DEPAUL UNIVERSITY

Medium Earth Orbit – For GPS Satellites

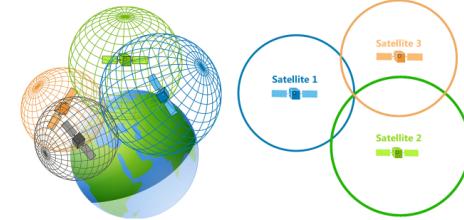


DEPAUL UNIVERSITY

7

GPS – Trilateration

- Requires signals from a minimum of 3 satellites
 - Usually need 4 or more for better accuracy



DEPAUL UNIVERSITY

8

Other GNSS

- Global Navigation Satellite System (GNSS)
- GNSS projects with global coverage
 - GLONASS – Soviet Union/Russian Federation
 - Galileo – European Union
 - BaiDou – China

DEPAUL UNIVERSITY

9

GLONASS



- Operated by Soviet Union/Russian Federation
- Constellation size: 27 satellites
 - As of 2017, 24 satellites in service
- Three equally spaced orbital planes
 - 120° apart, 8 satellite each, at 64.8° inclination
 - Orbit height: ≈ 19,130 km
 - Orbital period: 11 hour 15 minute. Repeat cycle: 8 days
- Horizontal accuracy: 4.5 – 7.4 m
- Free for civilian use
- Most iOS devices are equipped with receivers for both GPS and GLONASS since 2011

DEPAUL UNIVERSITY

10

Galileo



- Operated by European Space Agency (ESA)
- Constellation size: 30 satellites
 - 27 operational + 3 active spares
 - As of 2017, 11 satellites in service
- Three equally spaced orbital planes
 - 120° apart, 8 satellite each, at 56° inclination
 - Orbit height: ≈ 23,222 km
 - Orbital period: 14 hours 7 minutes. Repeat cycle: 10 days
- Horizontal accuracy: 1 m
- Primarily for civilian use

DEPAUL UNIVERSITY

11

BeiDou-2 & COMPASS



- Operated by Chinese National Space Agency (CNSA)
- Constellation size:
 - BeiDou-1: 4 geostationary satellites (GEO orbit)
 - BeiDou-2: 35 satellites, 30 MEO + 5 GEO orbit
 - As of 2017, 20 satellites in orbit
- MEO satellites
 - Orbit height: ≈ 21,150 km, at 55.5° inclination
- Regional service operational since 2012
- Global service: estimated 2020
- Horizontal accuracy: 10 m
- Free for civilian use

DEPAUL UNIVERSITY

12

A-GPS – Assisted GPS

- GPS augmented by signals from cell towers and WI-FI base stations
 - Faster than GPS, but less accurate than GPS and limited coverage
- Advantage:
 - Areas with strong cell and/or WI-FI signals
 - Even indoor or multi-floor buildings
- Disadvantage:
 - Open areas with weak cell/WI-FI signals
 - Moving vehicles, cars, airplanes, ships
- Require network connectivity, cellular or WI-FI, not free
- iOS devices use A-GPS for location

DEPAUL UNIVERSITY

13

Core Location

Location Services – Core Location Framework

- Current location, track changes
- Region monitoring
 - boundary crossing, including beacons
- Significant change notifications
 - Conserving power
- Warning**
 - Gathering location data is power-intensive
 - GPS receiver, cellular, WI-FI, CPU
 - Design appropriate pattern and interval for acquiring data and receiving notifications

DEPAUL UNIVERSITY

15

Required Device Capabilities

- Ensure devices have the required capabilities
- Add an entry in **Info.plist**
 - Required Device Capabilities**
 - Used by the App Store. Only the devices with the required capabilities can download and install.
- Keys related to location service
 - location-services**
 - If requires location service in general for current location
 - gps**
 - If needs location data with an accuracy offered only by GPS. More accurate than the data based on cellular or Wi-Fi

DEPAUL UNIVERSITY

16

Acquire Current Location

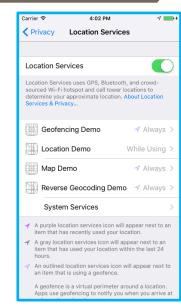
- Two location related services
- Standard location service**
 - Getting location data and tracking location changes
 - Configurable level of accuracy
- Significant-change location service**
 - Updates only on significant changes in location, e.g., $\geq 500m$
 - Every 15 minutes even if no change in location.
 - Less accurate. More energy efficient.
 - But, wake up every 15 minutes.

DEPAUL UNIVERSITY

17

Enable Location Service

- Enable Location Service in **Settings** on your device
 - Settings ► Privacy ► Location Services: **On**



DEPAUL UNIVERSITY

18

Using Location Service

- Create an instance of the `CLLocationManager`
 - Must maintain a strong reference
 - Location related tasks run asynchronously
- Request authorization for using Location Services
- Assign a delegate that conforms to the `CLLocationManagerDelegate` protocol
 - Callbacks for receiving location data and notifications
- Configure the service
- Start delivery of events

DEPAUL UNIVERSITY

19

Authorization for Location Services

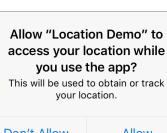
- Check status of authorization
 - `CLLocationManager.authorizationStatus()`
- Enum `CLAuthorizationStatus`
 - `notDetermined`
 - This app may request authorization
 - `restricted`
 - This app is not authorized to use location service
 - `denied`
 - Location service is disabled in Settings
 - `authorizedAlways`
 - This app is authorized to start location services at any time, even in background.
 - `authorizedWhenInUse`
 - This app is authorized to start most location services while running in the foreground.

DEPAUL UNIVERSITY

20

Request Authorization for Location Services

- Create an instance of *Location Manager*
- Request authorization when the status is `notDetermined`
 - `locationManager.requestWhenInUseAuthorization()`
 - `locationManager.requestAlwaysAuthorization()`
- Add a key in the `Info.plist`
 - *Privacy - Location When In Use Usage Description*
 - *Privacy - Location Always Usage Description*
 - A message explains to the user how the app uses this data



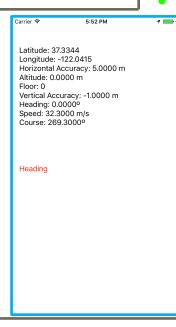
DEPAUL UNIVERSITY

21

Demo – Location Demo

Location Demo App

- Display location data
 - Current location
 - Accuracy
 - Heading



DEPAUL UNIVERSITY

22

Location Demo App – Initialization

```

import UIKit
import CoreLocation
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    let locationManager = CLLocationManager()
    override func viewDidLoad() {
        super.viewDidLoad()
        let status = CLLocationManager.authorizationStatus()
        if status == .denied || status == .restricted {
            message.text = "Location service not authorized"
        } else {
            locationManager.desiredAccuracy = kCLLocationAccuracyBest
            locationManager.distanceFilter = 1 // meter
            locationManager.delegate = self
            locationManager.requestWhenInUseAuthorization()
        }
    }
}

```

DEPAUL UNIVERSITY

24

Location Demo App – Start & Stop Updates

- Keep the location service active for the shortest duration necessary

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    let locationManager = CLLocationManager()

    override func viewDidAppear(_ animated: Bool) {
        if CLLocationManager.locationServicesEnabled() {
            locationManager.startUpdatingLocation()
        }
    }

    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        locationManager.stopUpdatingLocation()
    }
}
```

DEPAUL UNIVERSITY

25

Location Updates

- Each location update contains an array of `CLLocation` objects
 - At least one object, which represents the current location.
- May contain additional entries
 - in the order in which they occurred
 - the most recent location update at the end

DEPAUL UNIVERSITY

26

Location Data

- Class `CLLocation`

- Properties:

```
var coordinate: CLLocationCoordinate2D { get }

var altitude: CLLocationDistance { get }
    • CLLocationDistance alias of Double
var floor: CLFloor? { get }

var horizontalAccuracy: CLLocationAccuracy { get }
var verticalAccuracy: CLLocationAccuracy { get }
    • CLLocationAccuracy alias of Double

var timestamp: Date { get }
```

DEPAUL UNIVERSITY

27

Location Data - Coordinate

- Class `CLLocationCoordinate2D`

- Properties:

```
var latitude: CLLocationDegrees
    • CLLocationDegrees alias of Double
    • The latitude in degrees: [-90.0, +90.0]
    • Positive values indicate latitudes north of the equator.
    • Negative values indicate latitudes south of the equator.
```

```
var longitude: CLLocationDegrees
```

- The longitude in degrees: (-180.0, + 180.0)
- Measurements are relative to the zero meridian, aka the prime meridian
- Positive values extending east of the prime meridian
- Negative values extending west of the prime meridian

DEPAUL UNIVERSITY

28

Location Demo App – Current Location Updates

- Callback method of the location delegate

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    let locationManager = CLLocationManager()
    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        location.coordinate.latitude
        location.coordinate.longitude
        location.horizontalAccuracy
        location.altitude
        location.floor?.level
        location.verticalAccuracy
        locationManager.heading
    }
}
```

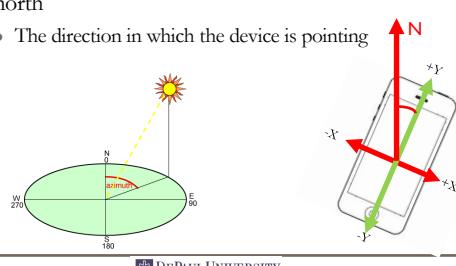
DEPAUL UNIVERSITY

29

Heading

- A device's azimuth relative to true or magnetic north

- The direction in which the device is pointing



DEPAUL UNIVERSITY

30

Heading Updates & Data

- Class `CLHeading`
- Properties


```
var magneticHeading: CLLocationDirection { get }
    • CLLocationDirection alias of Double
```

 - The heading (in degrees) relative to the magnetic north.
 - Clockwise. 0 – North, 90 – East, 180 – South, 270 – West

```
var trueHeading: CLLocationDirection { get }
```

 - The heading (in degrees) relative to the true north.

```
var headingAccuracy: CLLocationDirection { get }
```

DEPAUL UNIVERSITY

31

Location Demo App – Start & Stop Heading Updates

- Handled similarly to location updates

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    let locationManager = CLLocationManager()
    ...
    override func viewDidAppear(_ animated: Bool) {
        if CLLocationManager.headingAvailable() {
            locationManager.startUpdatingHeading()
        }
    }
    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
        locationManager.stopUpdatingHeading()
    }
    ...
}
```

DEPAUL UNIVERSITY

32

Location Demo App – Updates of Current Heading

- Callback of the location delegate
 - Notified of updates on heading

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    let locationManager = CLLocationManager()
    func locationManager(_ manager: CLLocationManager,
        didUpdateHeading newHeading: CLHeading) {
        ...
        newHeading.trueHeading
        newHeading.magneticHeading
        newHeading.headingAccuracy
    }
    ...
}
```

DEPAUL UNIVERSITY

33

Movement of Devices

- Properties of class `CLLocation`

```
var course: CLLocationDirection { get }
    • CLLocationDirection alias of Double
```

 - The direction in which the device is moving
 - In degrees starting at due north and continuing clockwise

```
var speed: CLLocationSpeed { get }
```

 - CLLocationSpeed alias of Double
 - The instantaneous speed of the device in meters per second
- Both are measured based on location data

DEPAUL UNIVERSITY

34

Monitoring Location: Pull or Push

- Push
 - Notification by the system, via callback to the delegate
 - Frequency dictated by change of location, heading, and accuracy
 - Configurable accuracy
 - System manages to minimize power usage
- Pull
 - Frequency dictated by application logic
 - Programmers are responsible to manage the frequency and power consumption

DEPAUL UNIVERSITY

35

Using Significant-Change Location Service

- Must request *Authorized Always* for location updates


```
locationManager.requestAlwaysAuthorization()
```
- Start and stop significant-change location service


```
locationManager.startMonitoringSignificantLocationChanges()
locationManager.stopMonitoringSignificantLocationChanges()
```
- Callback of the location delegate, same as standard location service.


```
func locationManager(_ manager: CLLocationManager,
        didUpdateLocations locations: [CLLocation]) {
        ...
    }
```

DEPAUL UNIVERSITY

36

Location Service in Background

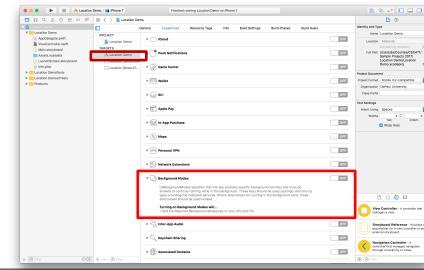
- Location events can be delivered even when the app is in background
 - May significantly increase battery consumption
 - Do this only when it offers genuine benefits, e.g., turn-by-turn navigation, fitness app
- Consider using significant-change location service
 - Lower accuracy. Lower power consumption
- Use standard location service when necessary
 - Enable the *Background Modes* capability in Xcode

DEPAUL UNIVERSITY

37

Enable Background Modes

- Xcode Project, *Capability* tab

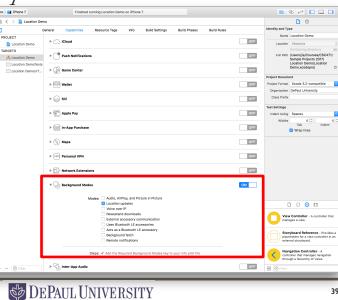


DEPAUL UNIVERSITY

38

Enable Background Modes

- Enable *Location updates*



DEPAUL UNIVERSITY

39

Map Kit



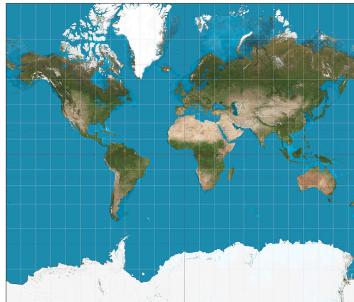
Map Coordinate Systems

- Map Kit uses a variant of *Mercator* map projection
 - Flemish cartographer *Gerardus Mercator* in 1569
- Cylindrical map projection
 - Standard for nautical navigation
 - Constant heading: straight line
- Parallel longitude lines
 - Distortion in polar regions
- Google Web Mercator projection
 - Non-conformant in polar regions (> 85° N/S)

DEPAUL UNIVERSITY

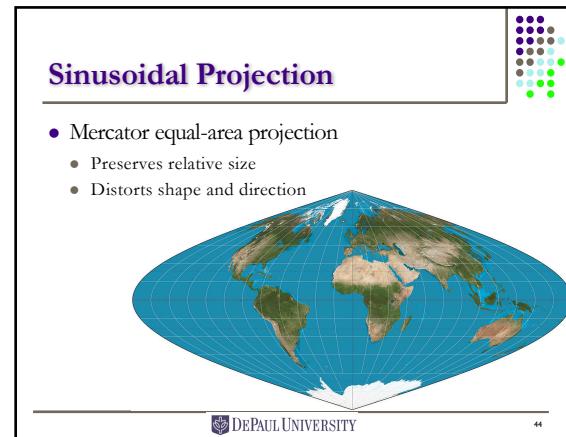
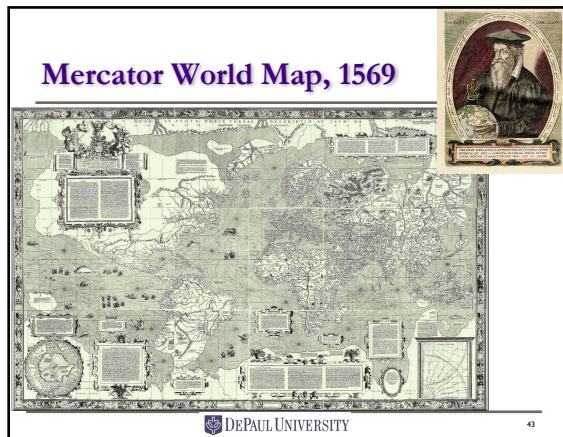
41

Mercator Projection of the World



DEPAUL UNIVERSITY

42



Map Coordinate vs Map Points

- Map coordinate:** latitude and longitude
 - A point on the spherical representation of the Earth.
 - `CLLocationCoordinate2D MKCoordinateSpan MKCoordinateRegion`
- Map point:** a point (x, y) on the Mercator map projection.
 - Specify shape and position of custom map overlays
 - `MKMapPoint MKMapSize MKMapRect`
- Point:** A point (x, y) in the coordinate system of a view object
 - For drawing custom view
 - `CGPoint CGSize CGRect`

DEPAUL UNIVERSITY 45

MapKit View

- Class `MKMapView`**
 - A subclass of `UIView`
 - Display map information
 - Manipulate the map contents
 - Annotations
 - Custom overlay
- Configurable**
 - Center at a coordinate. Specify the size of the area
 - Type of map: street map, satellite, hybrid, 3D
- Add `MapKit.framework` to project**

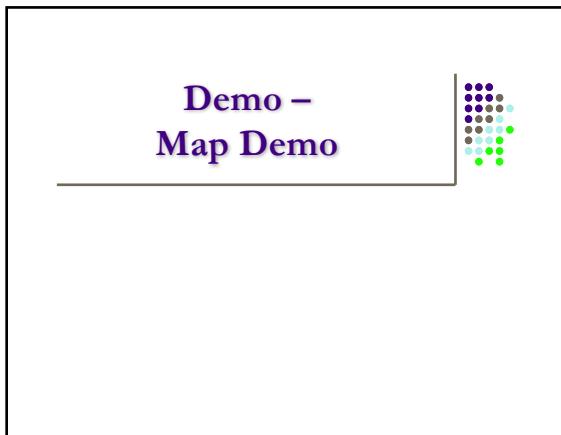
DEPAUL UNIVERSITY 46

Add a New Framework

The screenshot shows the Xcode interface with the 'Add New Framework' dialog open. The 'Linked Frameworks and Libraries' section is highlighted, and the 'MapKit.framework' checkbox is checked. The DePaul University logo is in the bottom left.

The Linked Frameworks

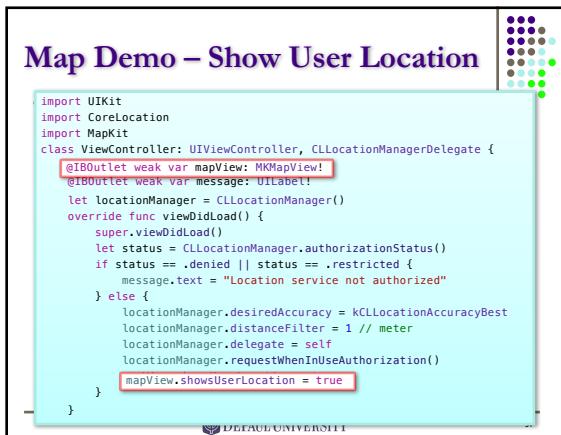
The screenshot shows the Xcode interface with the 'Linked Frameworks and Libraries' section of the build settings open. It lists several frameworks including MapKit, CoreLocation, and Foundation. The DePaul University logo is in the bottom right.



Map Demo – Screen Design

- Add a **MKMapView**
- Connect the outlets

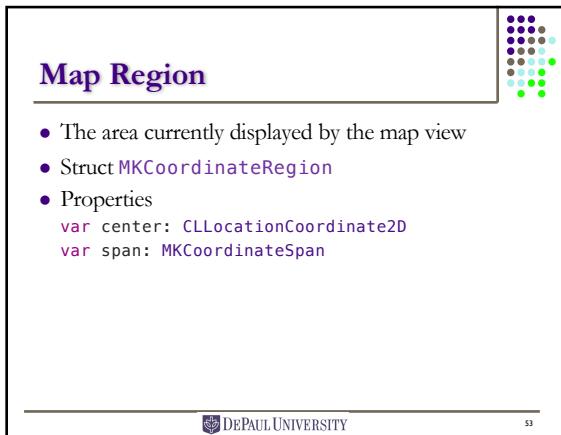
The screenshot shows the Xcode storyboard interface. On the left, the storyboard canvas displays a single view controller scene with a blue rectangular view labeled "MKMapView". On the right, a preview window shows a map of North America with a location marker. The bottom of the screen displays the DEPAUL UNIVERSITY logo and the number 50.



Map Demo – Show User Location

```
import UIKit
import CoreLocation
import MapKit
class ViewController: UIViewController, CLLocationManagerDelegate {
    @IBOutlet weak var mapView: MKMapView!
    @IBOutlet weak var message: UILabel!
    let locationManager = CLLocationManager()
    override func viewDidLoad() {
        super.viewDidLoad()
        let status = CLLocationManager.authorizationStatus()
        if status == .denied || status == .restricted {
            message.text = "Location service not authorized"
        } else {
            locationManager.desiredAccuracy = kCLLocationAccuracyBest
            locationManager.distanceFilter = 1 // meter
            locationManager.delegate = self
            locationManager.requestWhenInUseAuthorization()
            mapView.showsUserLocation = true
        }
    }
}
```

The screenshot shows a map of the United States with a blue circular marker indicating the user's location. Below the map, text displays the coordinates: Latitude: 41.8789, Longitude: -87.6261, and Horizontal Accuracy: 5.0000 M. The bottom of the screen displays the DEPAUL UNIVERSITY logo and the number 50.



Coordinate Span

- Struct **MKCoordinateSpan**
- Properties
 - var **latitudeDelta**: **CLLocationDegrees**
 - CLLocationDegrees** alias of **Double**
 - The amount of north-to-south distance (measured in degrees) to display on the map.
 - One degree of latitude is always approximately 111 kilometers (69 miles).
- var **longitudeDelta**: **CLLocationDegrees**
 - The amount of east-to-west distance (measured in degrees) to display for the map region.
 - One degree of longitude spans a distance of approximately 111 kilometers (69 miles) at the equator (vary based on the latitude)

DEPAUL UNIVERSITY

Map Demo - Region

- A span of 0.1° in latitude and 0.1° in longitude

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        message.text = ...

        mapView.setRegion(MKCoordinateRegion(center: location.coordinate,
                                             span: MKCoordinateSpan(latitudeDelta: 0.1, longitudeDelta: 0.1)),
                          animated: true)
    }
}

```

DEPAUL UNIVERSITY

55

Map Demo - Region

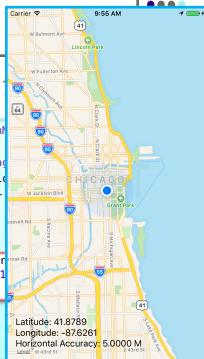
- A span of 0.1° in latitude and 0.1° in longitude

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        message.text = ...

        mapView.setRegion(MKCoordinateRegion(center: location.coordinate,
                                             span: MKCoordinateSpan(latitudeDelta: 0.1, longitudeDelta: 0.1)),
                          animated: true)
    }
}

```



DEPAUL UNIVERSITY

56

Map Demo – Region with Distance

- Specify the dimension of a region in meters
MKCoordinateRegionMakeWithDistance(
centerCoordinate, latitudinalMeters, longitudinalMeters)

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        message.text = ...

        mapView.setRegion(MKCoordinateRegionMakeWithDistance(
            location.coordinate, 2000, 2000),
            animated: true)
    }
}

```

DEPAUL UNIVERSITY

57

Map Demo – Region with Distance

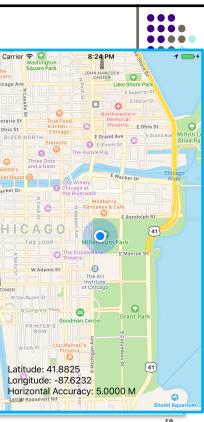
- Specify the dimension of the region in meters
MKCoordinateRegionMakeWithDistance(
centerCoordinate, latitudinalMeters, longitudinalMeters)

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        message.text = ...

        mapView.setRegion(MKCoordinateRegionMakeWithDistance(
            location.coordinate, 2000, 2000),
            animated: true)
    }
}

```



DEPAUL UNIVERSITY

58

Map Demo – Annotation

- Adopt protocol MKAnnotation

```
class Place : NSObject, MKAnnotation {
    var coordinate: CLLocationCoordinate2D
    var title: String?
    var subtitle: String?

    init(_ coordinate: CLLocationCoordinate2D,
         _ title: String? = nil,
         _ subtitle: String? = nil) {
        self.coordinate = coordinate
        self.title = title
        self.subtitle = subtitle
    }
}

```

DEPAUL UNIVERSITY

59

Map Demo - Annotation

```
class ViewController: UIViewController, CLLocationManagerDelegate {
    ...
    var annotation: MKAnnotation?

    func locationManager(_ manager: CLLocationManager,
                         didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]

        message.text = ...
        mapView.setRegion(...)

        if annotation != nil {
            mapView.removeAnnotation(annotation!)
        }
        let place = Place(location.coordinate, "You are here!")
        mapView.addAnnotation(place)
        annotation = place
    }
}

```

DEPAUL UNIVERSITY

60

Map Demo - Annotation

```

class ViewController: UIViewController, CLLocationManagerDelegate {
    var annotation: MKAnnotation?
    func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
        let location = locations[locations.count - 1]
        message.text = ""
        mapView.setRegion(...)

        if annotation != nil {
            mapView.removeAnnotation(annotation!)
        }
        let place = Place(location.coordinate, "You are here!")
        mapView.addAnnotation(place)
        annotation = place
    }
}

```

DEPAUL UNIVERSITY 61

References

- Apple Developer Resources
 - *Location and Maps Programming Guide*
 - https://developer.apple.com/library/prerelease/content/documentation/UserExperience/Conceptual/LocationAwarenessPG/Introduction/Introduction.html#/apple_ref/doc/uid/TP40009497-CH1-SW1

DEPAUL UNIVERSITY

62

Sample Code

- Location Demo
- Map Demo

DEPAUL UNIVERSITY 63

Next ...

- Geofencing
- 3D maps
- Geocoding

◆ Xcode, iOS, WatchOS are trademarks of Apple Inc.

DEPAUL UNIVERSITY

64