CS5530
Mobile/Wireless Systems
Core Location Framework

Yanyan Zhuang

Department of Computer Science

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

Overview

- Updates:
  - IT will upgrade Mac OS in 138
  - 3 iPhones, 3 iPads, 1 Apple watch
  - 2 Android phones, 2 Android tabs
- Core Location Framework

Core Location Framework

- Where does location come from
  - iOS devices employ different techniques
    - GPS, cell tower triangulation, IP address of available WiFi connections
  - Mechanism used by iOS is transparent to developer: auto uses the most accurate solution at a given time
- Just use the core location framework
  - Key classes: CLLocationManager and CLLocation
    - Access and store location

Location Manager Class

- Core Location Framework
  - var locationManager: CLLocationManager = CLLocationManager()
  - Location manager instance must seek permission from user
- Location access permission
  - locationManager.requestWhenInUseAuthorization()
  - locationManager.requestAlwaysAuthorization()

More on Permissions

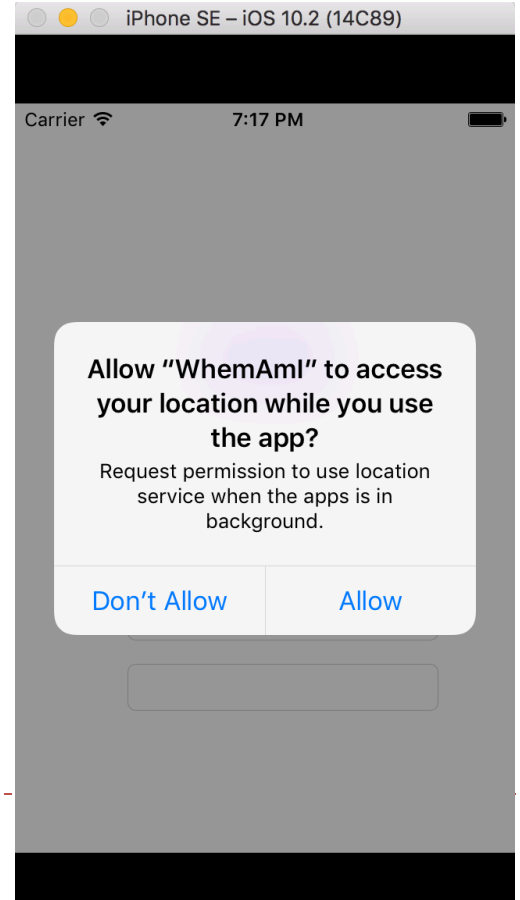
- locationManager.requestWhenInUseAuthorization()
- locationManager.requestAlwaysAuthorization()
  - Each method call requires a specific key-value pair added to **Information Property List** dictionary in app's Info.plist file
  - Value must describe the reason why the app needs access
    - NSLocationWhenInUseUsageDescription
    - NSLocationAlwaysUsageDescription

More on Permissions

- Info.plist

Application requires iPhone enviro...	Boolean	YES
Privacy - Location When In U...	String	Request permission to use location service when the apps is in background.
Main storyboard file base name	String	Main

- GUI



Location Accuracy

- Level of accuracy is specified via the desiredAccuracy property of the CLLocationManager object
  - locationManager.desiredAccuracy = kCLLocationAccuracyBest
- The greater the accuracy the greater the drain on device battery

Location Accuracy

- The greater the accuracy the greater the drain on device battery
  - kCLLocationAccuracyBestForNavigation – highest accuracy: intended solely for use when device is connected to power supply
  - kCLLocationAccuracyBest – The highest recommended level of accuracy for devices running on battery power
  - kCLLocationAccuracyNearestTenMeters - Accurate to within 10m
  - kCLLocationAccuracyHundredMeters, kCLLocationAccuracyKilometer, kCLLocationAccuracyThreeKilometers

Configuring the Distance Filter

- Location manager: report updates whenever any changes are detected in the location
  - locationManager.startUpdatingLocation() // details later
- **distanceFilter** property allows apps to specify amount of distance the location must change before an update is triggered
  - locationManager.distanceFilter = 1500.0

Location Manager Delegate

- Location manager updates/errors result in calls to two delegate methods
  - func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) { … }
    - Each time location changes, didUpdateLocations delegate method is called and passed as an argument an array of CLLocation objects: last object in the array containing the most recent location data
  - func locationManager(_ manager: CLLocationManager, didFailWithError error: Error) { … }

Starting Location Updates

- After suitably configured and authorized
  - locationManager.startUpdatingLocation()
- With each location update, didUpdateLocations is called by the location manager and passed information about the current location

Obtaining Location Information from CLLocation Objects

- Location information is passed through to the didUpdateLocation delegate method in the form of CLLocation objects
  - Latitude
  - Longitude
  - Horizontal Accuracy
  - Altitude
  - Altitude Accuracy

Obtaining Location Information from CLLocation Objects

- Longitude and Latitude
  - let latitude: CLLocationDistance = location.coordinate.latitude
  - let longitude: CLLocationDistance = location.coordinate.longitude
- Accuracy
  - let verticalAccuracy: CLLocationAccuracy = location.verticalAccuracy
  - let horizontalAccuracy: CLLocationAccuracy = location.horizontalAccuracy
- Altitude
  - let altitude: CLLocationDistance = location.altitude

Getting the Current Location

- Want user's current location without the need for continuous location updates
  - locationManager.requestLocation()
  - Identify the current location and call didUpdateLocations one time passing through the current location
  - Location updates are automatically turned off

Calculating Distances

- Distance between two CLLocation points can be calculated by calling distance(from:) of the end location and passing through the start location as an argument
  - var distance: CLLocationDistance = endLocation.distance(from: startLocation)

Reverse Geocode

```
func getPlacemarkFromLocation(location: CLLocation){
    CLGeocoder().reverseGeocodeLocation(location,
        completionHandler: {(placemarks, error) in
            if error {println("reverse geodcode fail: \(error)")}
            let pm = placemarks as [CLPlacemark]
            if pm.count > 0 {
              self.showAddPinViewController(placemarks[0] as CLPlacemark)
            }
    })
}
```

Simulating a Location in Simulator