Johns Hopkins Engineering

Location Services and Frameworks

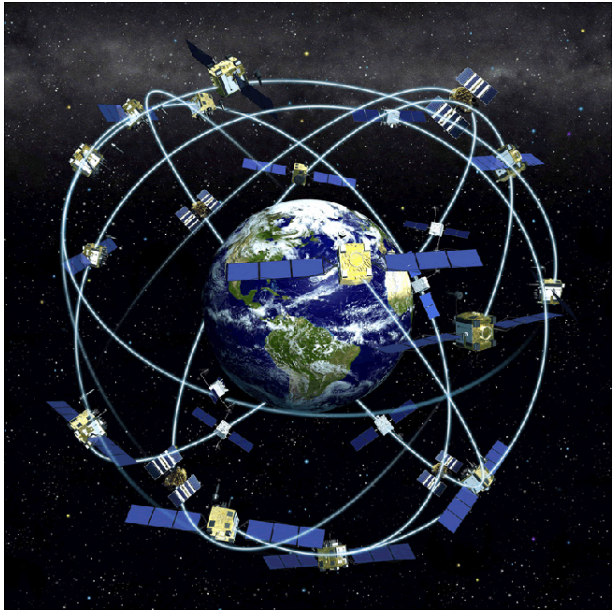
Module 4

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

CoreLocation Framework

Location Scale



Images courtesy Creative Commons Image Search

CLLocationManager

- CLLocationManager is the main class for getting location information from your device:
  - Tracking location changes with a configurable level of accuracy
  - Reporting a heading from the compass
  - Monitoring if the device enters or exits a region
  - Calculating range to nearby beacons
- CLLocationManager needs a delegate object (of type CLLocationManagerDelegate)
- Remember to ask for user permission and check for availability before asking for location data!

CLLocation

- Represents the location data generated by a CLLocationManager
- Geographic Coordinates and Altitude
- Also provides accuracy of measurements and time stamps
- In iOS, provides speed and heading
- Not usually created by user

https://developer.apple.com/documentation/corelocation/cllocation

Common Type Aliases

- CLLocationDegrees – a double, used to represent a longitudinal or latitudinal value, in degrees
- CLLocationDistance – a double, used to represent a distance (in meters) from a given location
- CLFloor – the logical floor of the building in which the user is located
- CLLocationAccuracy – a double, representing the accuracy of a coordinate value, in meters.

CLVisit

- Used to store information about places the user has been
- Created by the CLLocationManager, delivered to the delegate
- Includes location, and information about arrival and departure times
- Not created directly

https://developer.apple.com/documentation/corelocation/clvisit

Geofencing: Using CLCircularRegion

- CLCircularRegion is defined by a center coordinate and a radius

- CLCircularRegion’s contains method can be directly queried to see if a particular coordinate is within the region

- CLLocationManager listens for region entrances and exits – it gets notified of such events via delegate methods, such as locationManager(_, didEnterRegion)

CLBeacon and CLBeaconRegion

- Core Location can only provide accuracy to a certain level before various types of interference (e.g. building structures and the materials they are made of) result in bad results

- iOS supports the iBeacon protocol, which allows the device to sense bluetooth signals from small BTLE (Bluetooth Low Energy) devices. This signal can be used to calculate a range to the beacon, based on the power of the signal

- Beacons can be used as the center of a region (CLBeaconRegion), allowing for geofencing around the beacon.

- If an app knows where the various beacons are located in a known space, it can use the signals from the beacons to determine where the device is within that space, allowing for “indoor mapping”

- Estimote (www.estimote.com) is one of the leading manufacturers of beacons and has a very mature set of hardware and software

CLHeading

- Contains the computed values for the device’s azimuth (orientation) relative to true or magnetic north

- Also includes the raw, 3 dimensional vector used to calculate the heading

- Instances of this are received from the delegate through the `CLLocationManager` once `startUpdatingHeading()` is called

https://developer.apple.com/documentation/corelocation/clheading

CLGeoCoder

- Provides services to convert between a coordinate (longitude/latitude) and a user-friendly representation (street, city, state, country, landmark, or point of interest)
- Reverse-geocoding converts coordinate to user-friendly
- Forward-geocoding converts user-friendly to coordinate
- The results for both are returned via CLPlacemark

https://developer.apple.com/documentation/corelocation/clgeocoder

CLPlacemark

- Contains a user-friendly description of a geographic coordinate containing
  - Name of the place
  - Address
  - Other relevant information (Points of interest, geographically related data)

- Reverse-geocoding gives you a CLPlacemark for that location

- You can also create an instance yourself and populate information yourself

https://developer.apple.com/documentation/corelocation/clplacemark

Asking for Permission

- Just like with CoreMotion, don’t forget to ask for permission to access user location data!
  - In Info.plist, set the Privacy - Location Usage Description key, and provide a value stating why you need to use the user’s location
  - In the CLLocationManager, invoke the `requestWhenInUseAuthorization()`

JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING