

# «MyLocation»

Fabrice.Kordon@lip6.fr





# Goals of the example

## Fetch data from CLLocationManager

-  Current position
-  Magnetic data

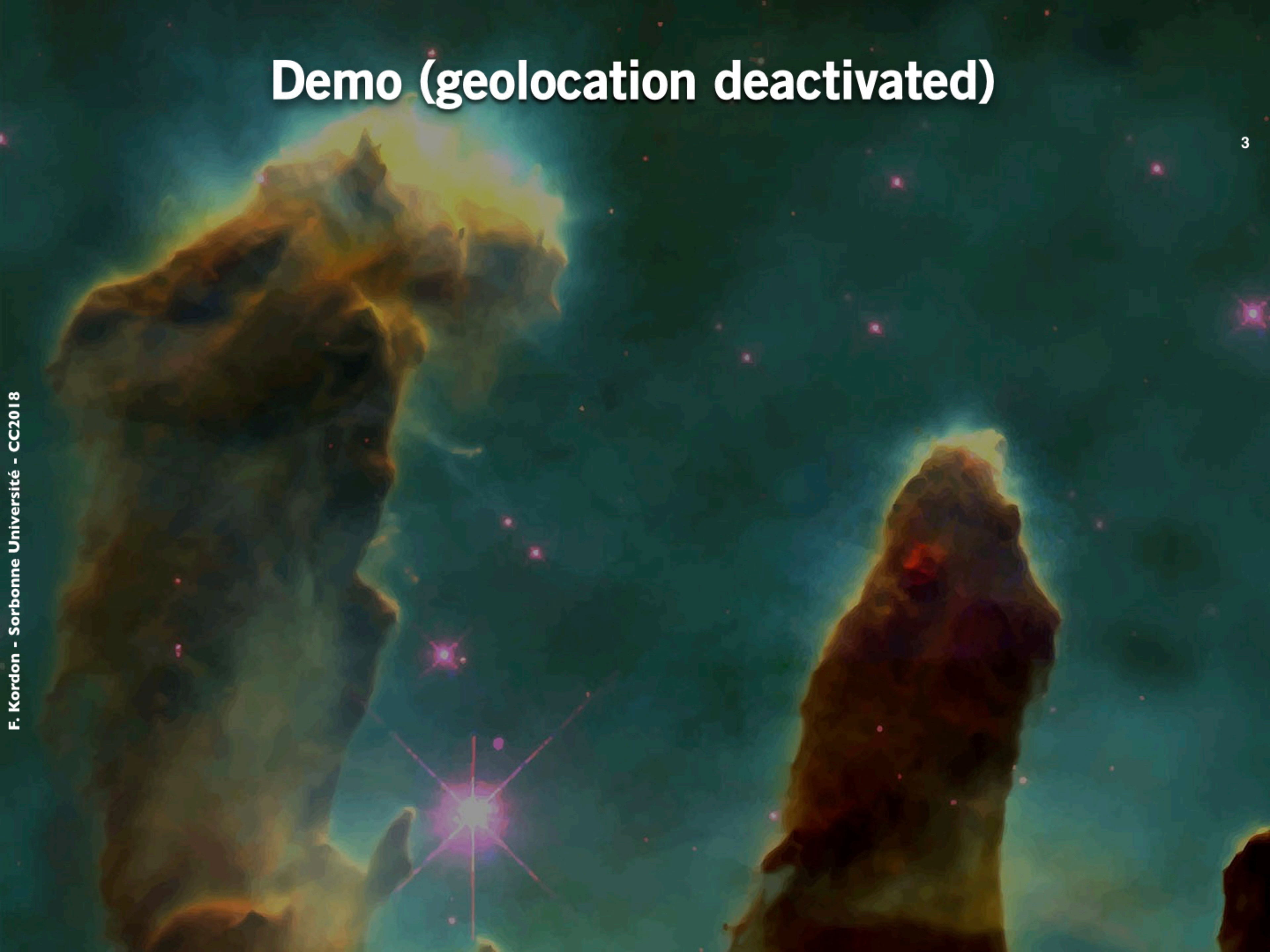


## Easy!!!

-  But you must configure a bit too...



# Demo (geolocation deactivated)





# Demo (no geolocation for MyLocation)

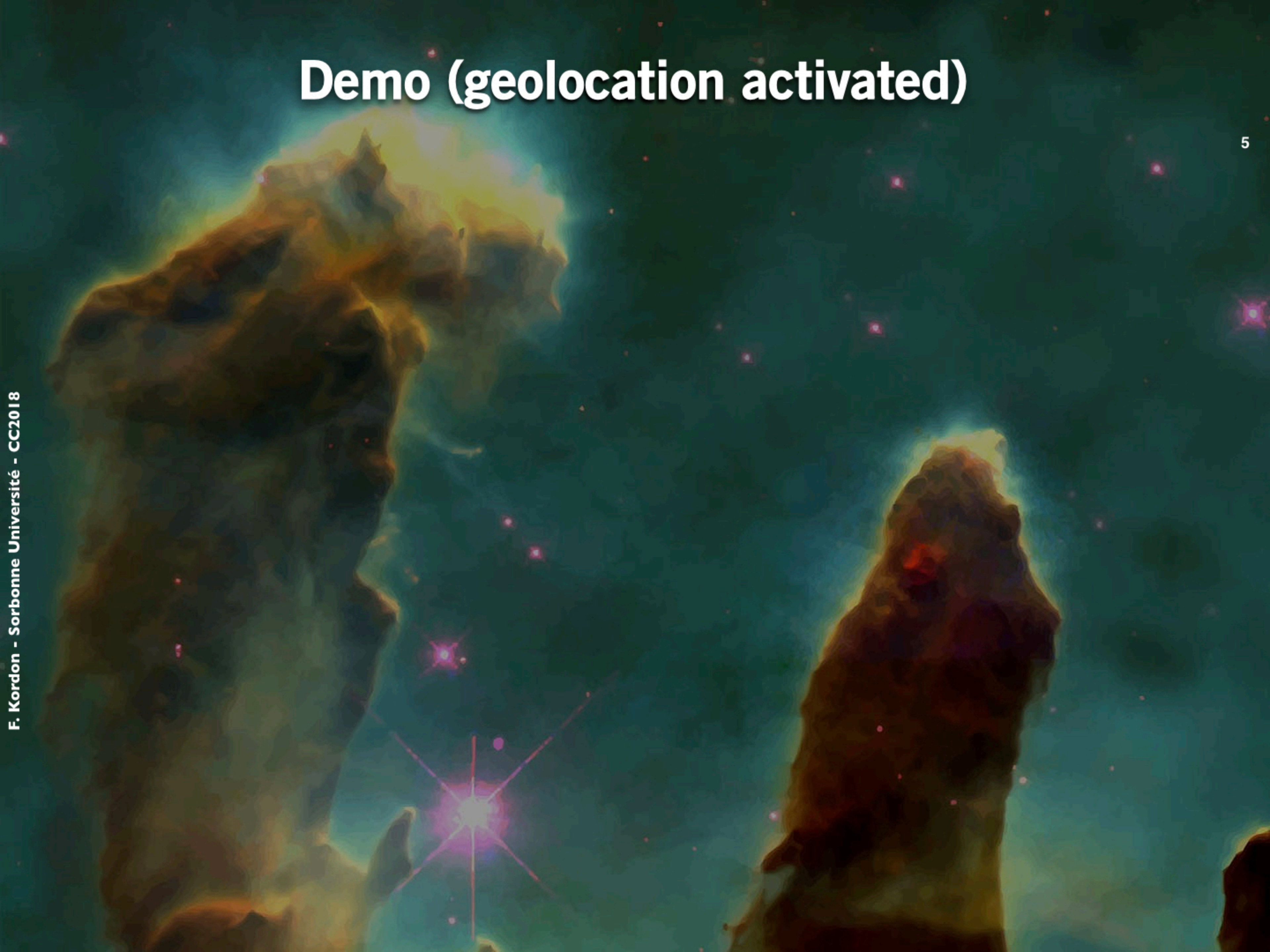
4





# Demo (geolocation activated)

5





# info.plist

```
20 <string>1</string>
21 <key>LSRequiresIPhoneOS</key>
22 <true/>
23 <key>NSLocationAlwaysAndWhenInUseUsageDescription</key>
24 <string>Explains why the App needs geolocation (used in the pref panel)</string>
25 <key>NSLocationWhenInUseUsageDescription</key>
26 <string>This App needs to access to geolocation</string>
27 <key>UILaunchStoryboardName</key>
28 <string>LaunchScreen</string>
29 <key>UIMainStoryboardFile</key>
30 <string>Main</string>
31
```

`<key>NSLocationAlwaysAndWhenInUseUsageDescription</key>`  
`<string>Explains why the App needs geolocation (used in the pref panel)</string>`  
`<key>NSLocationWhenInUseUsageDescription</key>`  
`<string>This App needs to access to geolocation</string>`



# ViewController

```
import UIKit
import CoreLocation // do not forget

class ViewController: UIViewController {

    private let v = MyView(frame: UIScreen.main.bounds)

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        self.view = v
    }
}
```



# ViewController

7

```
import UIKit
import CoreLocation // do not forget

class ViewController: UIViewController {

    private let v = MyView(frame: UIScreen.main.bounds)

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        self.view = v
    }

    override func viewWillAppear(_ animated: Bool) {
        // Cannot be done in viewDidLoad (too early for self to
        // be able to perform present
        if !CLLocationManager.locationServicesEnabled() {
            self.view = UIView()
            self.view.backgroundColor = .red
            let alert = UIAlertController(title: "Error",
                                         message: "Location deactivated",
                                         preferredStyle: .alert)
            alert.addAction(UIAlertAction(title: "OK",
                                         style: .default,
                                         handler: nil))
            self.present(alert, animated: true, completion:nil)
        }
    }
}
```



# ViewController

```
override func viewWillTransition(to size: CGSize,  
    with coordinator: UINavigationControllerTransitionCoordinator) {  
    v.drawInSize(size)  
}  
}
```



# MyView

```
import UIKit
import CoreLocation // do not forget

class MyView: UIView, CLLocationManagerDelegate {

    private let button = UIButton(type: .system)
    private let crLocation = UITextView()
    private let lattitude = UILabel()
    private let longitude = UILabel()
    private let altitude = UILabel()
    private let speed = UILabel()
    private let crtTime = UILabel()
    private let trueH = UILabel()
    private let magnH = UILabel()
    private let headings = UILabel()
    private let locMgr = CLLocationManager()

    private var activated = false // true when CLLocationManager is active
```



# MyView

```
override init(frame: CGRect) {
    super.init(frame: frame)
    self.backgroundColor = UIColor.white
    button.setTitle("Get my location", for: .normal)
    button.addTarget(self,
                    action: #selector(getCurrentPosition),
                    for: .touchDown)

    locationManager.requestWhenInUseAuthorization() // when in use
    // locationManager.requestAlwaysAuthorization() // always

    latitude.text = "Lat : --"
    latitude.textAlignment = .center
    longitude.text = "Long : --"
    longitude.textAlignment = .center
    altitude.text = "Alt : --"
    altitude.textAlignment = .center
    speed.text = "Speed : --"
    speed.textAlignment = .center
    crtTime.text = "--"
    crtTime.textAlignment = .center
    crLocation.text = "where am I?"
    crLocation.isSelectable = false
    crLocation.textColor = UIColor.red
    crLocation.font = UIFont.systemFont(ofSize: 16.0)
    crLocation.textAlignment = .center
    trueH.text = "True N : --"
    trueH.textAlignment = .center
    magnH.text = "Magn N : --"
    magnH.textAlignment = .center
    headings.text = "--"
    headings.textAlignment = .center
    self.addSubview(button)
    self.addSubview(latitude)
    self.addSubview(longitude)
    self.addSubview(altitude)
    self.addSubview(speed)
    self.addSubview(crtTime)
    self.addSubview(crLocation)
    self.addSubview(trueH)
    self.addSubview(magnH)
    self.addSubview(headings)
    self.drawInSize(frame.size)
}
```



# MyView

```
required init(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}

func drawInSize(_ size: CGSize) {
    var delta = CGFloat(0.0)
    button.frame = CGRect(x: size.width / 2.0 - 60.0,
                          y: 40.0, width: 120.0, height: 20.0)
    latitude.frame = CGRect(x: 10.0, y: 70.0,
                            width: 120.0, height: 30.0)
    longitude.frame = CGRect(x: size.width - 130.0,
                             y: 70.0, width: 120.0, height: 30.0)
    altitude.frame = CGRect(x: 10.0, y: 100.0,
                            width: 100.0, height: 30.0)
    speed.frame = CGRect(x: size.width - 130.0,
                         y: 100.0, width: 120.0, height: 30.0)
    crtTime.frame = CGRect(x: 10.0, y: 130,
                           width: size.width - 20.0, height: 30.0)
    if size.width > size.height {
        delta = 0.0
    } else {
        delta = 50.0
    }
    crLocation.frame = CGRect(x: 10.0, y: 180.0,
                              width: size.width - 20.0,
                              height: 40.0 + delta)
    trueH.frame = CGRect(x: 10.0, y: 230.0 + delta,
                         width: 120.0, height: 30.0)
    magnH.frame = CGRect(x: size.width - 130.0, y: 230.0 + delta,
                         width: 120.0, height: 30.0)
    headings.frame = CGRect(x: 10.0, y: 270.0 + delta,
                            width: size.width - 20.0, height: 30.0)
}
```



# MyView

```
@objc func getCurrentPosition() {
    if activated {
        button.setTitle("Get my location", for: .normal)
        activated = false
        locMngr.stopUpdatingLocation()
        locMngr.stopUpdatingHeading()
        crLocation.text = "(Last known location)"
    } else {
        crLocation.text = "Searching..."
        locMngr.delegate = self
        locMngr.distanceFilter = kCLLocationAccuracyBest // Greedy in energy!
        locMngr.startUpdatingLocation()
        if CLLocationManager.headingAvailable() {
            locMngr.startUpdatingHeading()
        } else {
            trueH.text = "True N : N.A."
            magnH.text = "Magn N : N.A."
            headings.text = "N.A."
        }
        button.setTitle("Stop", for: .normal)
        activated = true
    }
}
```



# MyView

```
// CLLocationManagerDelegate protocol

func locationManager(_ manager: CLLocationManager,
                    didFinishWithError error: Error) {
    crLocation.text = "Error : " + error.localizedDescription
    lattitude.text = "Lat : --"
    longitude.text = "Long : --"
    altitude.text = "Alt : --"
    speed.text = "Speed : --"
    crtTime.text = "--"
    trueH.text = "True N : --"
    magnH.text = "Magn N : --"
    headings.text = "--"
    locMngr.stopUpdatingLocation()
    locMngr.stopUpdatingHeading()
    button.setTitle("Cherche moi!", for: .normal)
    activated = false
    print(error)
}
```



# MyView

```
// CLLocationManagerDelegate protocol (continued)

func locationManager(_ manager: CLLocationManager,
                    didUpdateLocations locations: [CLLocation]) {
    latitude.text = String(format: "Lat : %2.3f",
                          (manager.location?.coordinate.latitude)!)
    longitude.text = String(format: "Long : %2.3f",
                            (manager.location?.coordinate.longitude)!)
    altitude.text = String(format: "Alt : %2.3f",
                           (manager.location?.altitude)!)
    speed.text = String(format: "speed : %2.3f",
                        (manager.location?.speed)!)
    let myDate = DateFormatter()
    myDate.dateFormat = "dd-MM-yyyy, hh:mm:ss"
    crtTime.text = myDate.string(from: (manager.location?.timestamp)!)
    crLocation.text = locations[0].description
}

func locationManager(_ manager: CLLocationManager,
                    didUpdateHeading newHeading: CLHeading) {
    trueH.text = String(format: "True N : %.1f",
                        (manager.heading?.trueHeading)!)
    magnH.text = String(format: "Magn N : %.1f",
                        (manager.heading?.magneticHeading)!)
    headings.text = String(format: "x=%.3f, y=%.3f, z=%.3f",
                            (manager.heading?.x)!, (manager.heading?.y)!,
                            (manager.heading?.z)!)
}
}
```

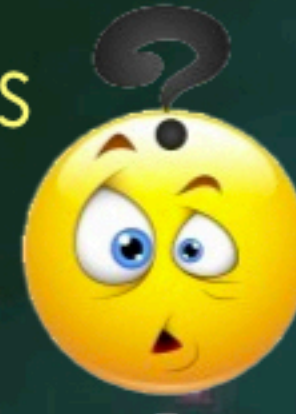


# As a conclusion...

## Retrieving GPS coordinates is easy

 As well as for orientation informations


▶ If you know how to use these



## Privacy handled by iOS

 Declaration (info.plist) + request

▶ always or whenInUse

 Apple's suggestions

▶ Think about when using geolocation

▶ choose the appropriate mode

▶ Then program it!

## Now you are ready to play with maps!

