

# «WhereAml»



Fabrice.Kordon@lip6.fr



# Goals of the example

 **Display the current location**

 **Insert it in a map**

-  Let the map move
-  Let the user add pins

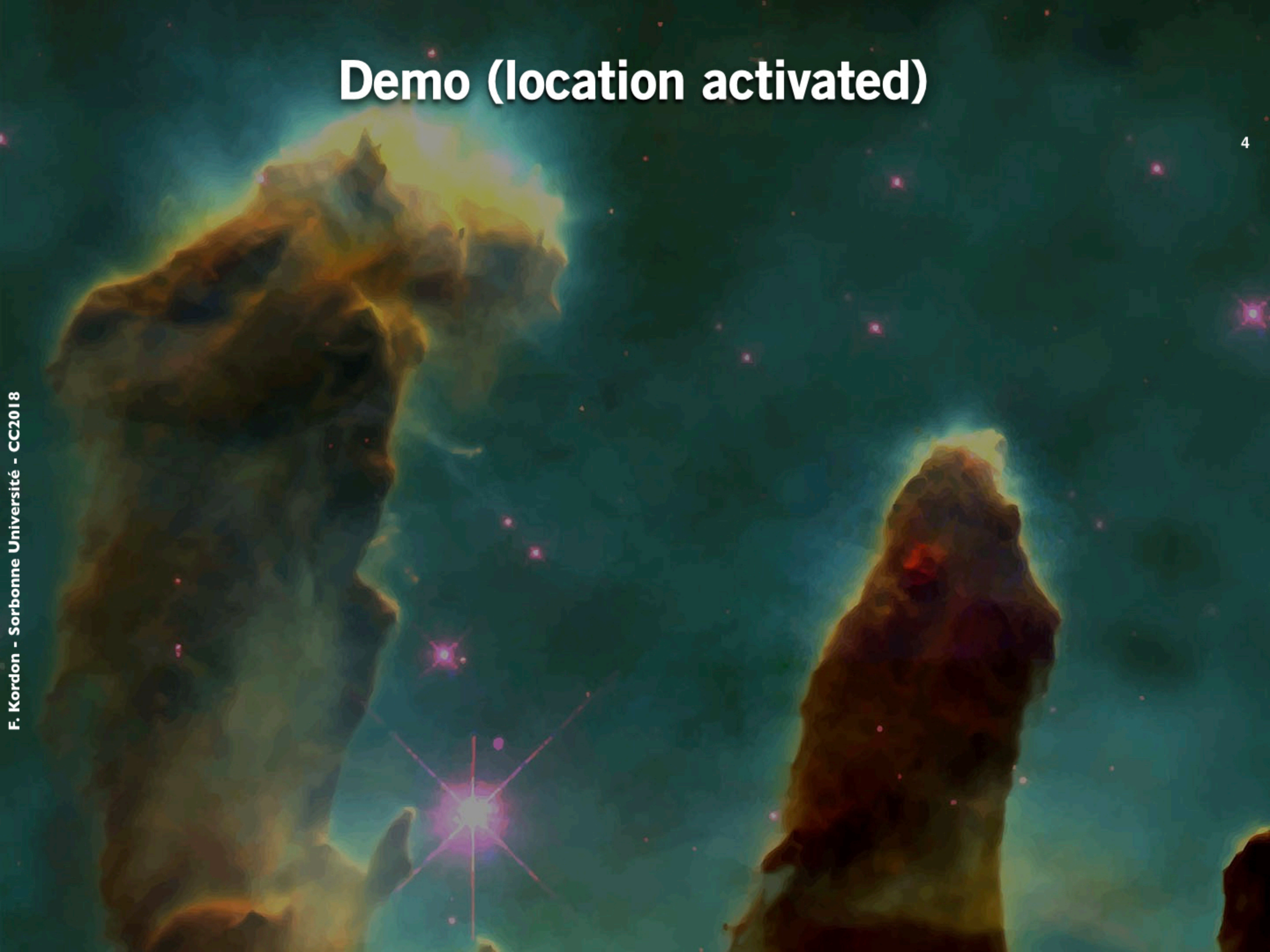


# Demo (no geolocation for WhereAml)



# Demo (location activated)

4



# AnAnnotation

```
import UIKit
import MapKit // Important (for MKAnnotation)

class AnAnnotation: NSObject, MKAnnotation {
    // Properties requested by the protocol
    var coordinate: CLLocationCoordinate2D
    var title: String?
    var subtitle: String?

    // some init mechanism
    init(c : CLLocationCoordinate2D) {
        coordinate = c
        super.init()
    }

    convenience init(c : CLLocationCoordinate2D, t : String) {
        self.init(c: c)
        title = t
    }

    convenience init(c : CLLocationCoordinate2D, t : String, st : String) {
        self.init(c: c, t: t)
        subtitle = st
    }
}
```

# ViewController



**Already seen**

The exact same of «MyLocation»



## Basically

- Creation a MyView
- Checking for Location availability
- Handling orientation

# MyView

7

```
import UIKit
import CoreLocation
import MapKit

class MyView: UIView, CLLocationManagerDelegate, MKMapViewDelegate {

    private let find = UIButton(type: .system)
    private let add = UIButton(type: .contactAdd)
    private let location = UITextView()
    private let lattitude = UILabel()
    private let longitude = UILabel()
    private let crtTime = UILabel()
    private let CLmngr = CLLocationManager()
    private let map = MKMapView()

    private var count = 1
    private var color = UIColor.orange
}
```

# MyView

```
override init(frame : CGRect) {
    find.setTitle("Where am I?", for: .normal)
    location.isSelectable = false
    location.isEditable = false
    location.text = "Where am I?"
    location.textAlignment = .center
    lattitude.text = "Lat : --"
    lattitude.textAlignment = .left
    longitude.text = "Long : --"
    longitude.textAlignment = .right
    crtTime.text = "--"
    crtTime.textAlignment = .center
    map.isScrollEnabled = true
    map.isZoomEnabled = true
    CLmngr.distanceFilter = 1.0 // precision = 1m
    CLmngr.requestWhenInUseAuthorization()
    super.init(frame: frame)
    self.backgroundColor = UIColor.white
    find.addTarget(self,
        action: #selector(computePosition(sender:)),
        for: .touchDown)
    add.addTarget(self,
        action: #selector(addPin(sender:)),
        for: .touchDown)
    map.delegate = self
    CLmngr.delegate = self
    self.addSubview(find)
    self.addSubview(add)
    self.addSubview(location)
    self.addSubview(lattitude)
    self.addSubview(longitude)
    self.addSubview(crtTime)
    self.addSubview(map)
    self.drawInSize(UIScreen.main.bounds.size)
}
```



# MyView

```
required init?(coder aDecoder: NSCoder) { // Requested by Xcode
    fatalError("init(coder:) has not been implemented")
}

func drawInSize(_ size: CGSize) {
    var top = 20;
    if UIDevice.current.userInterfaceIdiom == .phone &&
        size.height >= 812 {
        top = 30
    } else if UIDevice.current.userInterfaceIdiom == .phone &&
        size.width > size.height {
        top = 0
    }
    find.frame = CGRect(x: Int(size.width / 2 - 50),
                        y: top + 10, width: 100, height: 30)
    add.frame = CGRect(x: Int(size.width - 40),
                      y: top + 10, width: 30, height: 30)
    crtTime.frame = CGRect(x: 10, y: top + 40,
                          width: Int(size.width - 20), height: 20)
    lattitude.frame = CGRect(x: 10, y: top + 70,
                            width: 120, height: 30)
    longitude.frame = CGRect(x: Int(size.width - 130),
                            y: top + 70, width: 120, height: 30)
    location.frame = CGRect(x: 10, y: top + 100,
                          width: Int(size.width - 20), height: 60)
    map.frame = CGRect(x: 0, y: top + 160,
                      width: Int(size.width),
                      height: Int(size.height - 160))
}
```

# MyView

```
@objc func computePosition(sender : UIButton) {
    location.text = "I am searching..."
    CLmngr.startUpdatingLocation()
}

@objc func addPin(sender : UIButton) {
    let a = AnAnnotation(c: map.centerCoordinate,
                        t: String(format:"location %d", count),
                        st: "subtitle")

    count += 1
    map.addAnnotation(a)
}

func nextColor (c : UIColor) -> UIColor {
    switch c {
        case .orange : return .red
        case .red : return .blue
        case .blue : return .green
        case .green : return .black
        case .black : return .purple
        default : return .orange
    }
}
```

# MyView

```
// CLLocationManagerDelegate protocol

func locationManager(_ manager: CLLocationManager,
                    didUpdateLocations locations: [CLLocation]) {
    lattitude.text = String(format: "Lat : %2.4f",
                            (manager.location?.coordinate.latitude)!)
    longitude.text = String(format: "Long : %2.4f",
                             (manager.location?.coordinate.longitude)!)

    let myDate = DateFormatter()
    myDate.dateFormat = "dd-MM-yyyy, HH'h'mm"
    crtTime.text = myDate.string(from: (manager.location?.timestamp)!)
    location.text = manager.location?.description
    CLmngnr.stopUpdatingLocation() // Only one measure
    // Map update
    let span = MKCoordinateSpan(latitudeDelta: 0.035, longitudeDelta: 0.035)
    let region = MKCoordinateRegion(center: (manager.location?.coordinate)!,
                                    span: span)

    map.setRegion(region, animated: true)
    map.showsUserLocation = true // Add a pin on the current location
}

func locationManager(_ manager: CLLocationManager,
                    didFailWithError error: Error) {
    lattitude.text = "Lat : --"
    longitude.text = "Long : --"
    crtTime.text = "--- error ---"
    location.text = error.localizedDescription
}
```

# MyView

```
// MKMapViewDelegate protocol

func mapView(_ mapView: MKMapView,
             viewFor annotation: MKAnnotation) -> MKAnnotationView? {
    let epingle = MKPinAnnotationView(annotation: annotation,
                                     reuseIdentifier: "ppm")

    epingle.pinTintColor = color
    color = nextColor(c:color)
    epingle.canShowCallout = true
    epingle.rightCalloutAccessoryView = UIButton(type : .detailDisclosure)
    return epingle
}

func mapView(_ mapView: MKMapView,
             annotationView view: MKAnnotationView,
             calloutAccessoryControlTapped control: UIControl) {
    location.text = "coordinate of «" + (view.annotation?.title!)! + "»"
    let c = view.annotation?.coordinate
    lattitude.text = String (format:"Lat : %2.4f", (c?.latitude)!)
    longitude.text = String (format:"Long : %2.4f", (c?.longitude)!)
}

func mapView(_ mapView: MKMapView, didSelect view: MKAnnotationView) {
    location.text = "You selected «" + (view.annotation?.title!)! + "»"
}
}
```

# info.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>CFBundleDevelopmentRegion</key>
  <string>$(DEVELOPMENT_LANGUAGE)</string>
  <key>CFBundleExecutable</key>
  <string>$(EXECUTABLE_NAME)</string>
  <key>CFBundleIdentifier</key>
  <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
  <key>CFBundleInfoDictionaryVersion</key>
  <string>6.0</string>
  <key>CFBundleName</key>
  <string>$(PRODUCT_NAME)</string>
  <key>CFBundlePackageType</key>
  <string>APPL</string>
  <key>CFBundleShortVersionString</key>
  <string>1.0</string>
  <key>CFBundleVersion</key>
  <string>1</string>
  <key>LSRequiresIPhoneOS</key>
  <true/>
  <key>UILaunchStoryboardName</key>
  <string>LaunchScreen</string>
  <key>UIMainStoryboardFile</key>
  <string>Main</string>
  <key>UIRequiredDeviceCapabilities</key>
  <array>
    <string>armv7</string>
  </array>
  <key>UISupportedInterfaceOrientations</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>UISupportedInterfaceOrientations~ipad</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationPortraitUpsideDown</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>NSLocationUsageDescription</key>
  <string>Explain why your App needs Core Location</string>
  <key>NSLocationWhenInUseUsageDescription</key>
  <string>Explain why your App needs Core Location (in preferences)</string>
</dict>
</plist>
```

# As a conclusion...



**That's all!!!**



**But...**

**... you can play more**

🎧 Let's be back again a next video