iBeacon

Danny Shen 申潤五

關於 iBeacon

權限設定

為了

- 偵測 iPhone 進入某個 iBeacon 範圍(Raging monitoring) &
- 量測 iPhone & iBeacon 間的距離 (iBeacon ranging),

我們需在 Info.plist 頁面加入以下兩種權限的存取文字說明,

- Privacy — Location Always and When In Use Usage Description
- Privacy — Location When In Use Usage Description

使用 CoreLocation 來登記 iBeacon Region

```
var locationManager:CLLocationManager!

override func viewDidLoad() {
    super.viewDidLoad()
    locationManager = CLLocationManager()
    locationManager.delegate = self
    locationManager.requestAlwaysAuthorization()
    monitorBeacons()
}
func monitorBeacons() {
    if CLLocationManager.isMonitoringAvailable(for: CLBeaconRegion.self) {
        if let proximityUUID = UUID(uuidString: "B9407F30-F5F8-466E-AFF9-25556B57FE6D"){
            let region = CLBeaconRegion(uuid: proximityUUID, major: 3736, identifier: "beaconID")
            locationManager.startMonitoring(for: region)
        }
    }
}
```

實作 CordLocationManagementDelegate

```
func locationManager(_ manager: CLLocationManager, didEnterRegion region: CLRegion) {
    print("enter region")
}
func locationManager(_ manager: CLLocationManager, didExitRegion region: CLRegion) {
    print("exit region")
}
```

```
    func locationManager(_ manager: CLLocationManager, didStartMonitoringFor region:
CLRegion) {
        print("start Region")
    }
```

進入/離開區域時通知

```
let content = UNMutableNotificationContent()
content.title = "歡迎光臨"
content.subtitle = "\((region as! CLBeaconRegion).minor)"
content.badge = 1
content.sound = UNNotificationSound.default
let request = UNNotificationRequest(identifier: "notification", content: content, trigger: nil)
UNUserNotificationCenter.current().add(request, withCompletionHandler: nil)
if CLLocationManager.isRangingAvailable() {
    locationManager.startRangingBeacons(in: region as! CLBeaconRegion)
}
```

側測 Beacons

```
locationManager.startRangingBeacons(in: region)


  func locationManager(_ manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon], in region: CLBeaconRegion) {

for beacon in beacons{

        print(beacon.proximityUUID, beacon.major, beacon.minor)

}

}
```

Beacon 距離

```
switch beacon.proximity {
            case .immediate:
                print("immediate")
            case .near:
                print("near")
            case .far:
                print("far")
            case .unknown:
                print("unknown")
            @unknown default:
                print("@unknown default")
            }
```

```
beacon.accuracy
```

海龍公式計算座標

```
// c 為 Beacon1 與 Beacon 2 的距離, a 為  手機與 Beacon1 的 距離, b 為 Beacon2 的

var s = ( a + b + c ) / 2  //由三角形三邊長取得海龍公司參數, 用來計算面積

var area = sqrt( s * ( s - a ) * ( s - b ) * ( s - c ) )  //取得面積

let y = area * 2 / c  //由三角形面積估算Y 座標

let x = sqrt( ( a * a ) - ( y * y ) ) // 使用畢氏定理(勾股定理)計算x 座標

print("x:\(x) y:\(y)")
```