



iOS Device

Xcode 12 / iOS 14

Ryan Chung



課程大綱

- 取得Beta版iOS
 - 手機登入 BetaSoftware
- 取得裝置相關資訊
 - 手機型號 UIDevice
 - 作業系統版本
- 相簿與相機
 - 開啟相機
 - 取得相簿圖片
- 地圖
 - 顯示指定位置地圖
 - 顯示圖標與客製化
 - 顯示目前位置與地址
- 內建瀏覽器
 - 顯示特定網頁
 - 分頁瀏覽
- 感應器
 - 磁力儀 Magnetometer
 - 加速度計 Accelerometer
 - 三軸陀螺儀 Gyroscope
 - 綜合 Device-motion
- 裝置震動
 - 發出震動 AudioService
- 音樂播放 AVKit
 - MP3播放 AVAudioPlayer
 - 兩顆按鈕/一顆按鈕 @State
- 影片播放 AVKit
 - 專案內影片 VideoPlayer
 - 網路影片 URL



取得iOS Beta版



Apple Beta Software Program

Help make the next releases of iOS, iPadOS, macOS, tvOS and watchOS our best yet. As a member of the Apple Beta Software Program, you can take part in shaping Apple software by test-driving pre-release versions and letting us know what you think.

<https://beta.apple.com/sp/betaprogram/>



安裝步驟

- 重要資料備份
- 用手機的瀏覽器，進入下方網址
- 完成後，進入設定->一般->軟體更新，與一般軟體更新步驟相同
- 應該會看見Beta版本出現!

<https://beta.apple.com/profile>



在實機上測試

- 設定Development Team
 - 點擊Project圖樣 -> Targets -> Team
 - 設定為個人Team或開發者帳號
- 確認手機允許第三方程式執行
 - 在手機上 -> 設定 -> 一般
 - 描述檔與裝置管理 -> 信任
- 免費有安裝App數量限制



取得裝置相關資訊

- 輸出裝置名稱
- 輸出作業系統版本





取得裝置相關資訊

```
import SwiftUI
struct ContentView: View {

    var body: some View {
        Button(action: {
            let myDevice: UIDevice = UIDevice.current
            print(myDevice.name)
            print(myDevice.systemName)
            print(myDevice.systemVersion)
            print(myDevice.model)
            print(myDevice.localizedModel)
            print(myDevice.identifierForVendor!)
        }) {
            HStack{
                Image(systemName: "gear")
                    .resizable()
                    .aspectRatio(contentMode: .fit)
                    .frame(width: 50, height: 50)
                    .foregroundColor(.gray)
                Text("Get Device Info")
                    .font(.largeTitle)
                    .bold()
                    .foregroundColor(.gray)
            }
        }
    }
}
```

iPhone SE (2nd generation)
iOS
14.0
iPhone
iPhone
XXXXXXXX-XXXX-XXXX-XXXX-XX



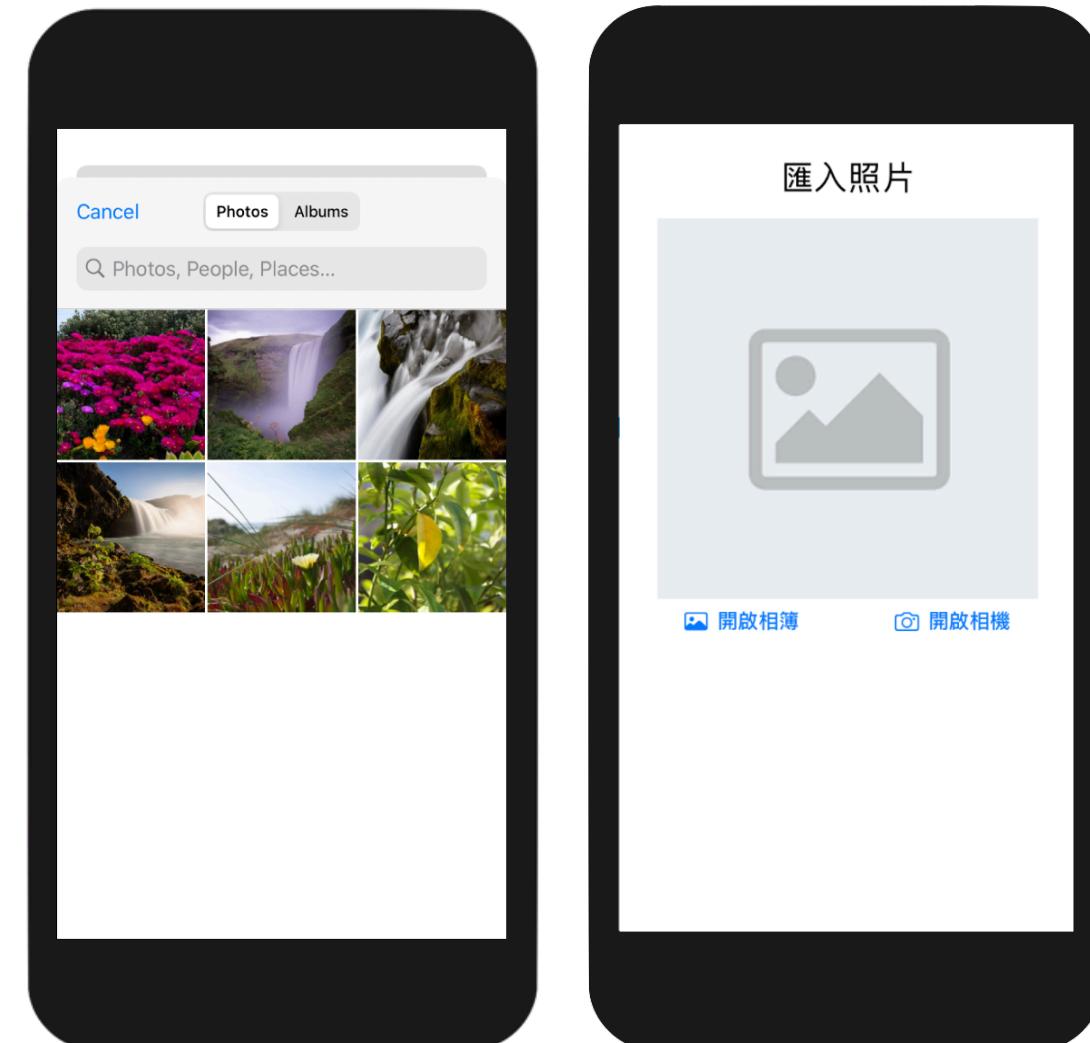
電池狀態

- 電池電量 batteryLevel
 - 0~1
 - 若為-1代表電池狀態為未知
- 電池狀態 batteryState
 - 0 未知 unknown
 - 1 未充電 unplugged
 - 2 充電中 charging
 - 3 滿電且插電中 full

```
myDevice.isBatteryMonitoringEnabled=true  
print(myDevice.batteryLevel)  
print(myDevice.batteryState.rawValue)  
myDevice.isBatteryMonitoringEnabled=false
```

開啟相機與相簿

- 點擊按鈕開啟相機
 - 切換前後鏡頭
 - 拍攝照片
 - 重拍或使用該張照片
- 點擊按鈕開啟相簿
 - 選擇相簿中的照片
 - 顯示選擇的照片





HelloImagePicker

- 建立專案
- 新增預準備檔案ImagePicker.swift
- 開始編輯ContentView.swift
- 新增三個變數

```
import SwiftUI
```

```
struct ContentView: View {
```

```
    @State var isShowPicker: Bool = false
    @State var image: Image? = Image("placeholder")
    @State var isSourceTypeAlbum: Bool = true
```

```
    var body: some View {
```

<https://tinyurl.com/swift-image-picker>



Info.plist

- 左邊導覽區 -> Info.plist
- 中間Information Property List 旁邊 + 號

Key	Type	Value
▼ Information Property List	 Dictionary	(15 items)

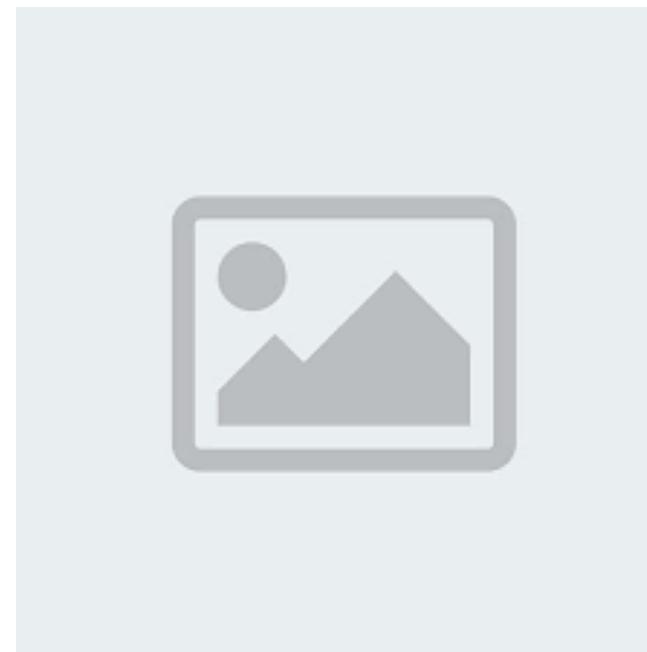
- 產生一列後，找到Privacy – Camera Usage ...
- 在Value輸入使用相機的原因

Key	Type	Value
▼ Information Property List	Dictionary	(16 items)
Privacy - Camera Usage Description	String	For User Profile Photo



放入預設圖片一張

- Assets.xcassets



<https://i.ibb.co/pv0fvZ4/placeholder.png>



ContentView主要結構

- 標題、圖片顯示區、兩顆按鈕

```
var body: some View {
    VStack {
        Text("匯入照片")
        image?
            .resizable()
            .scaledToFit()
            .frame(height: 320)
        HStack{
            Button(action: {
            }) {
            }
            Spacer()
            Button(action: {
            }) {
            }
        }
        .padding(.horizontal, 50)
        Spacer()
    }
    .sheet(isPresented: $isShowPicker) {
    }
}
```





詳細設定

- 標題與圖片

```
var body: some View {  
    VStack {  
        Text("匯入照片")  
            .font(.title)  
        image?  
            .resizable()  
            .scaledToFit()  
            .frame(height: 320)
```



詳細設定

- 兩顆按鈕

```
HStack{  
    Button(action: {  
        withAnimation {  
            self.isShowPicker.toggle()  
            self.isSourceTypeAlbum = true  
        }  
    }) {  
        Image(systemName: "photo")  
        Text("開啟相簿").font(.headline)  
    }  
    Spacer()  
    Button(action: {  
        withAnimation {  
            self.isShowPicker.toggle()  
            self.isSourceTypeAlbum = false  
        }  
    }) {  
        Image(systemName: "camera")  
        Text("開啟相機").font(.headline)  
    }  
}.padding(.horizontal, 50)
```



詳細設定

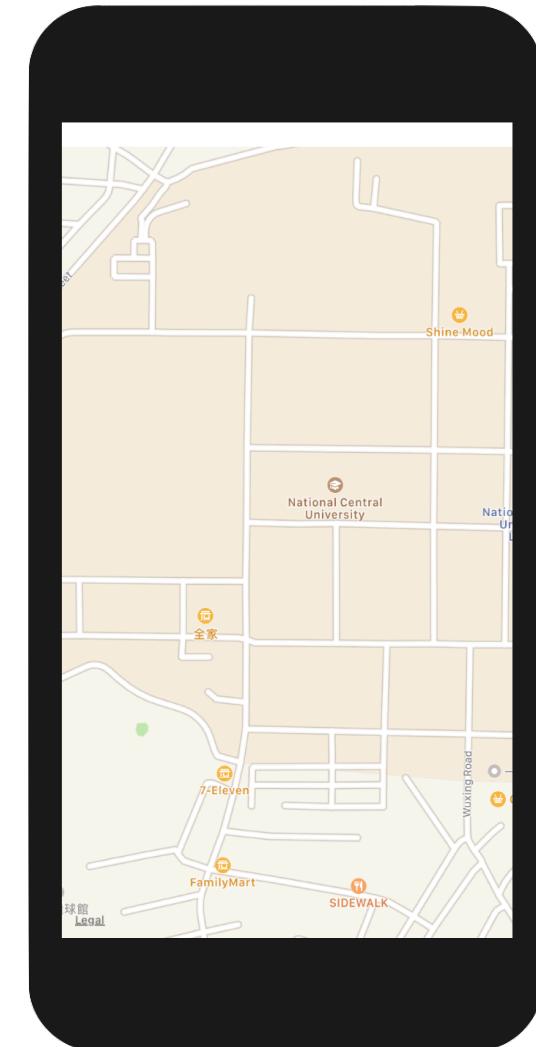
- Sheet內容

```
.sheet(isPresented: $isShowPicker) {  
    ImagePicker(  
        image: self.$image,  
        isSourceTypeAlbum: self.$isSourceTypeAlbum)  
}
```



地圖

- 打開地圖
 - 中心點
 - 縮放層級





建立MapView

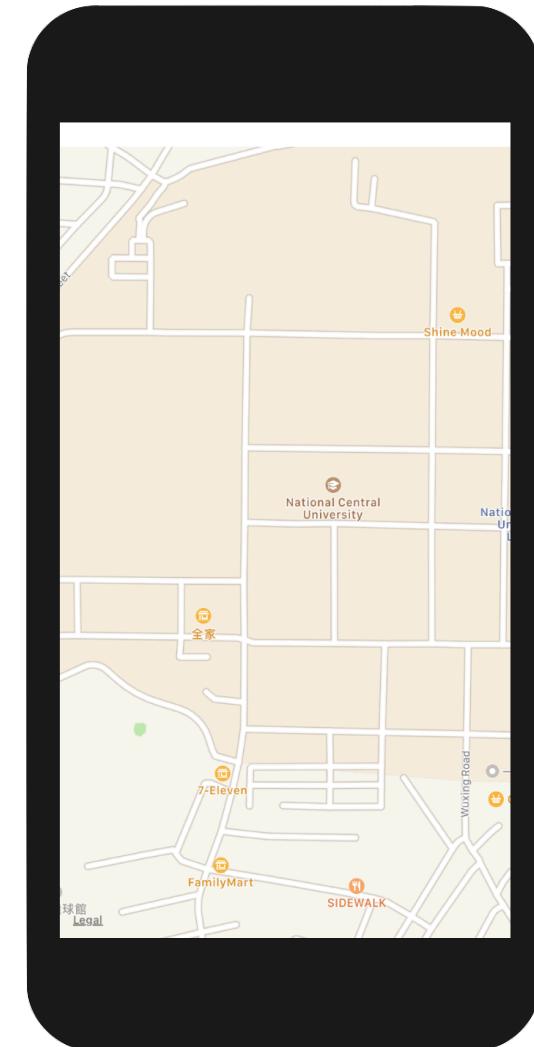
```
import SwiftUI
import MapKit

struct MapView: UIViewRepresentable {

    func makeUIView(context: Context) -> MKMapView{
        MKMapView(frame: .zero)
    }

    func updateUIView(_ view: MKMapView, context: Context){
        let coordinate = CLLocationCoordinate2D(
            latitude: 24.967773, longitude: 121.191585)
        let span = MKCoordinateSpan(
            latitudeDelta: 0.005, longitudeDelta: 0.005)
        let region = MKCoordinateRegion(
            center: coordinate, span: span)
        view.setRegion(region, animated: true)
    }
}

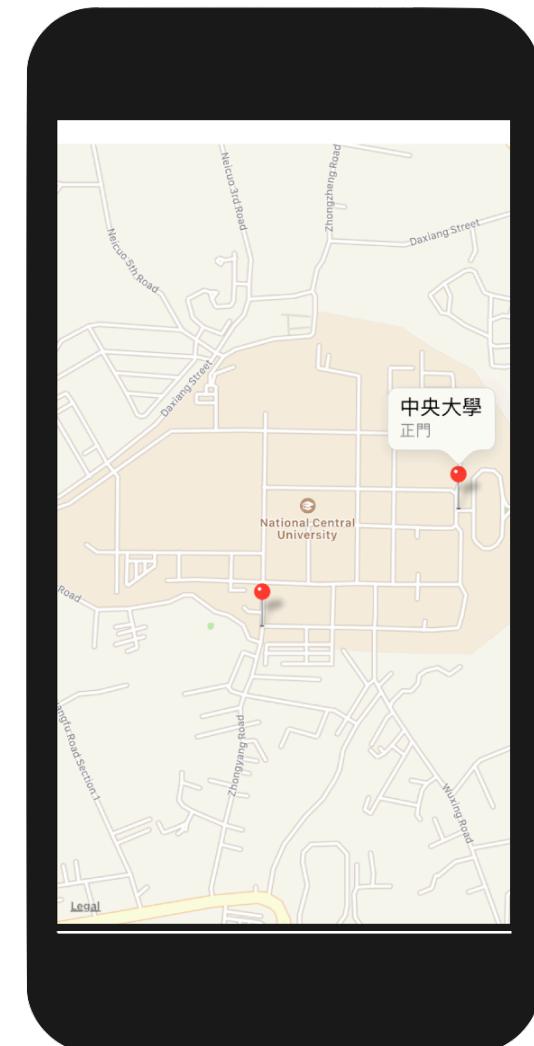
struct MapView_Previews: PreviewProvider {
    static var previews: some View {
        MapView()
    }
}
```





新增地標

- 指定特定經緯度進行標示
 - 主標題
 - 副標題
 - 多個





LandmarkAnnotation.swift

- 新增類別檔案，繼承NSObject, MKAnnotation

```
import UIKit
import MapKit

class LandmarkAnnotation: NSObject, MKAnnotation {
    let title: String?
    let subtitle: String?
    let coordinate: CLLocationCoordinate2D

    init(title: String?,
         subtitle: String?,
         coordinate: CLLocationCoordinate2D) {
        self.title = title
        self.subtitle = subtitle
        self.coordinate = coordinate
    }

    static func requestMockData() -> [LandmarkAnnotation]{
        return [
            LandmarkAnnotation(title: "中央大學",
                               subtitle:"正門",
                               coordinate: .init(latitude: 24.968299, longitude: 121.195464)),
            LandmarkAnnotation(title: "中央大學",
                               subtitle:"後門",
                               coordinate: .init(latitude: 24.965946, longitude: 121.191130))
        ]
    }
}
```



MapView

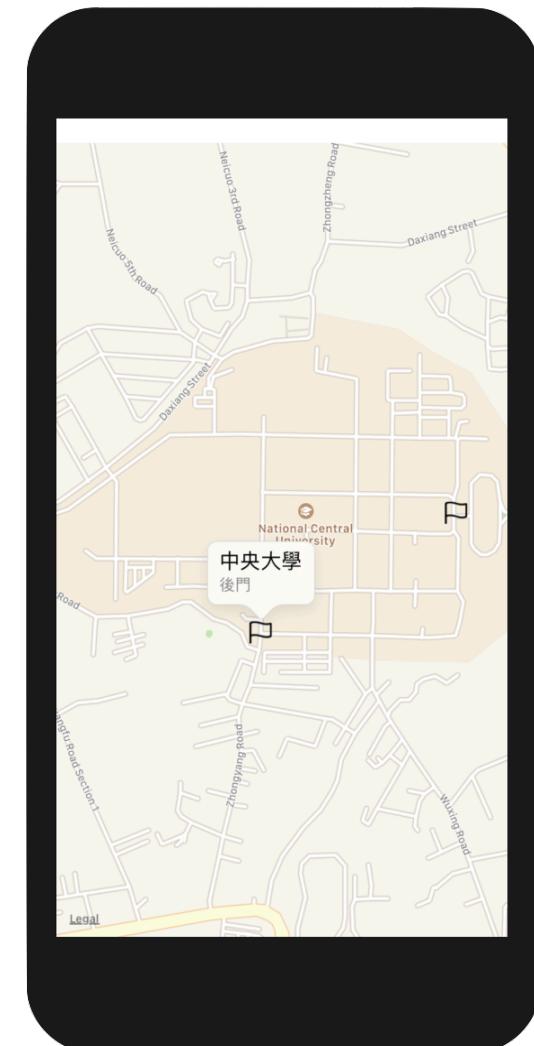
- 增加圖標於MapView中

```
struct MapView: UIViewRepresentable {  
  
    let landmarks = LandmarkAnnotation.requestMockData()  
  
    func makeUIView(context: Context) -> MKMapView{  
        MKMapView(frame: .zero)  
    }  
  
    func updateUIView(_ view: MKMapView, context: Context){  
        let coordinate = CLLocationCoordinate2D(  
            latitude: 24.967773, longitude: 121.191585)  
        let span = MKCoordinateSpan(latitudeDelta: 0.01, longitudeDelta: 0.01)  
        let region = MKCoordinateRegion(center: coordinate, span: span)  
        view.setRegion(region, animated: true)  
        view.addAnnotations(landmarks)  
    }  
}
```



客製化圖標

- 置換原始圖標





MapViewCoordinator.swift

```
import UIKit
import MapKit

class MapViewCoordinator: NSObject, MKMapViewDelegate {

    var mapViewController: MapView

    init(_ control: MapView) {
        self.mapViewController = control
    }

    func mapView(_ mapView: MKMapView, viewFor
                annotation: MKAnnotation) -> MKAnnotationView?{
        //Custom View for Annotation
        let annotationView = MKAnnotationView(annotation: annotation,
                                              reuseIdentifier: "customView")
        annotationView.canShowCallout = true
        //Your custom image icon
        annotationView.image = UIImage(systemName: "flag")
        return annotationView
    }
}
```



修改MapView

```
struct MapView: UIViewRepresentable {

    let landmarks = LandmarkAnnotation.requestMockData()

    func makeUIView(context: Context) -> MKMapView{
        MKMapView(frame: .zero)
    }

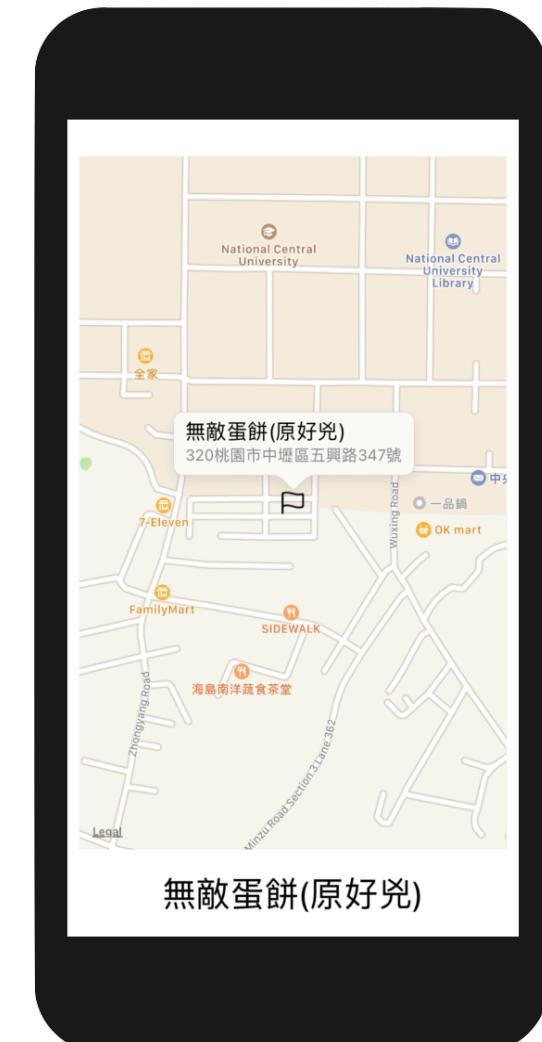
    func updateUIView(_ view: MKMapView, context: Context){
        let coordinate = CLLocationCoordinate2D(
            latitude: 24.967773, longitude: 121.191585)
        let span = MKCoordinateSpan(latitudeDelta: 0.01, longitudeDelta: 0.01)
        let region = MKCoordinateRegion(center: coordinate, span: span)
        view.setRegion(region, animated: true)
        view.delegate = context.coordinator
        view.addAnnotations(landmarks)
    }

    func makeCoordinator() -> MapViewCoordinator{
        MapViewCoordinator(self)
    }
}
```



地圖

- 顯示指定餐廳地圖
 - 下方寫餐廳名稱
 - 餐廳位置在地圖中央
 - 圖標
 - 直接顯示





ContentView

- 指定餐廳結構、內容，傳入MapView

```
struct Restaurant{  
    var name:String  
    var description:String  
    var location:CLLocationCoordinate2D  
}  
  
var thisRestaurant = Restaurant(  
    name: "無敵蛋餅(原好兇)",  
    description: "320桃園市中壢區五興路347號\n+886928210319",  
    location:CLLocationCoordinate2D(latitude: 24.9655141,  
                                    longitude: 121.1924017))  
  
struct ContentView: View {  
  
    @State var currentRestaurant:Restaurant = thisRestaurant  
  
    var body: some View {  
        VStack{  
            MapView(currentRestaurant: $currentRestaurant)  
            Text("\(thisRestaurant.name)")  
                .frame(height: 50, alignment: .center)  
                .font(.title)  
        }  
        .padding(.all, 10)  
    }  
}
```



MapView

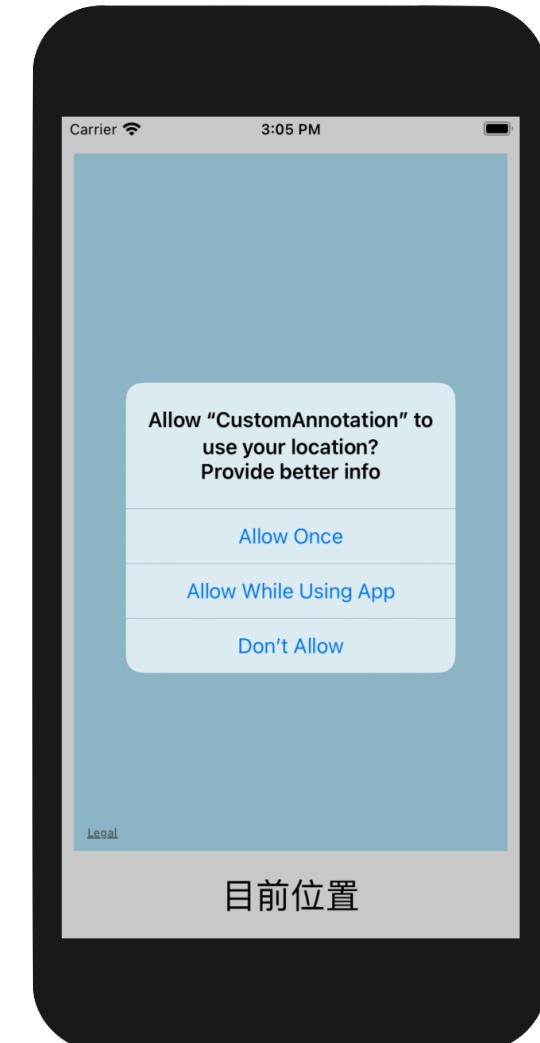
- 傳入餐廳資訊、調整地圖縮放、設定圖標顯示

```
struct MapView: UIViewRepresentable {  
  
    @Binding var currentRestaurant: Restaurant  
    //let landmarks = LandmarkAnnotation.requestMockData()  
  
    func makeUIView(context: Context) -> MKMapView{  
        MKMapView(frame: .zero)  
    }  
  
    func updateUIView(_ view: MKMapView, context: Context){  
        //let coordinate = CLLocationCoordinate2D(  
        //    latitude: 24.967773, longitude: 121.191585)  
        let coordinate = currentRestaurant.location  
        let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)  
        let region = MKCoordinateRegion(center: coordinate, span: span)  
        view.setRegion(region, animated: true)  
        view.delegate = context.coordinator  
        //view.addAnnotations(landmark)  
  
        let landmark = LandmarkAnnotation(  
            title: currentRestaurant.name,  
            subtitle: currentRestaurant.description,  
            coordinate: currentRestaurant.location)  
  
        view.addAnnotation(landmark)  
        view.selectAnnotation(landmark, animated: true)  
    }  
  
    func makeCoordinator() -> MapViewCoordinator{  
        MapViewCoordinator(self)  
    }  
}
```



取得使用者現在位置

- 設定取用說明
- 使用者允許開放權限
- 顯示使用者位置於地圖上





設定使用說明

- Info.plist
 - Location Always and When In Use
 - Location When In Use

Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
Privacy - Location Always and When In Use Usage Description	String	For better info provided
Privacy - Location When In Use Usage Description	String	For better info provided



MapView

```
struct MapView: UIViewRepresentable {

    @Binding var currentRestaurant: Restaurant
    let mapView = MKMapView(frame: .zero)
    var locationManager = CLLocationManager()

    func setupManager() {
        locationManager.desiredAccuracy = kCLLocationAccuracyBest
        locationManager.requestWhenInUseAuthorization()
        locationManager.requestAlwaysAuthorization()
    }

    func makeUIView(context: Context) -> MKMapView{
        setupManager()
        mapView.showsUserLocation = true
        mapView.userTrackingMode = .follow
        return mapView
    }

    func updateUIView(_ view: MKMapView, context: Context){
        let coordinate = mapView.userLocation.coordinate
        let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)
        let region = MKCoordinateRegion(center: coordinate, span: span)
        view.setRegion(region, animated: true)
    }

    func makeCoordinator() -> MapViewCoordinator{
        MapViewCoordinator(self)
    }
}
```



ContentView

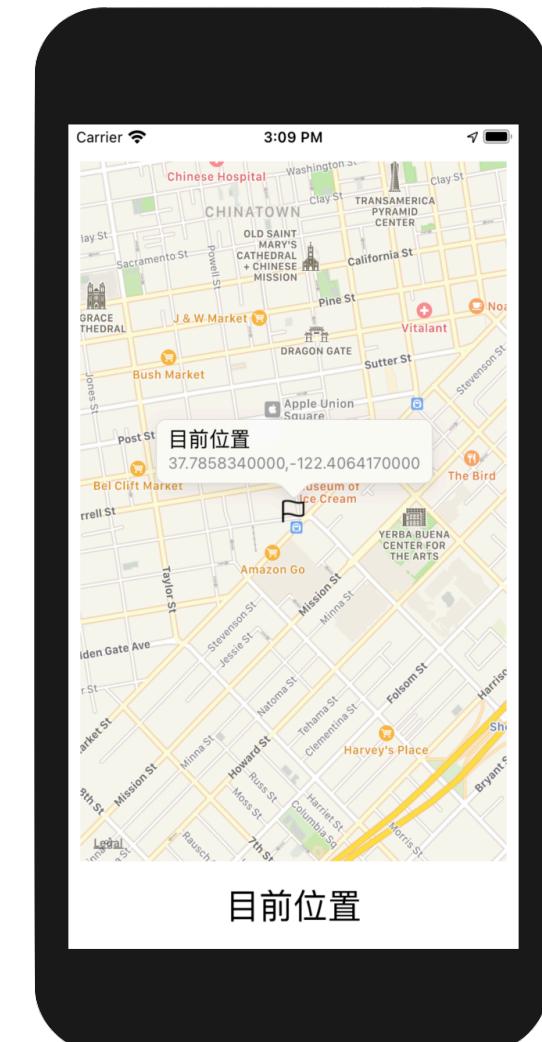
- 把餐廳名稱文字改為目前位置

```
var body: some View {  
    VStack{  
        MapView(currentRestaurant: $currentRestaurant)  
        Text("目前位置")  
            .frame(height: 50, alignment: .center)  
            .font(.title)  
    }  
    .padding(.all, 10)  
}
```



顯示使用者所在位置經緯度

- 取得經緯度資訊



目前位置



MapView中的updateUIView

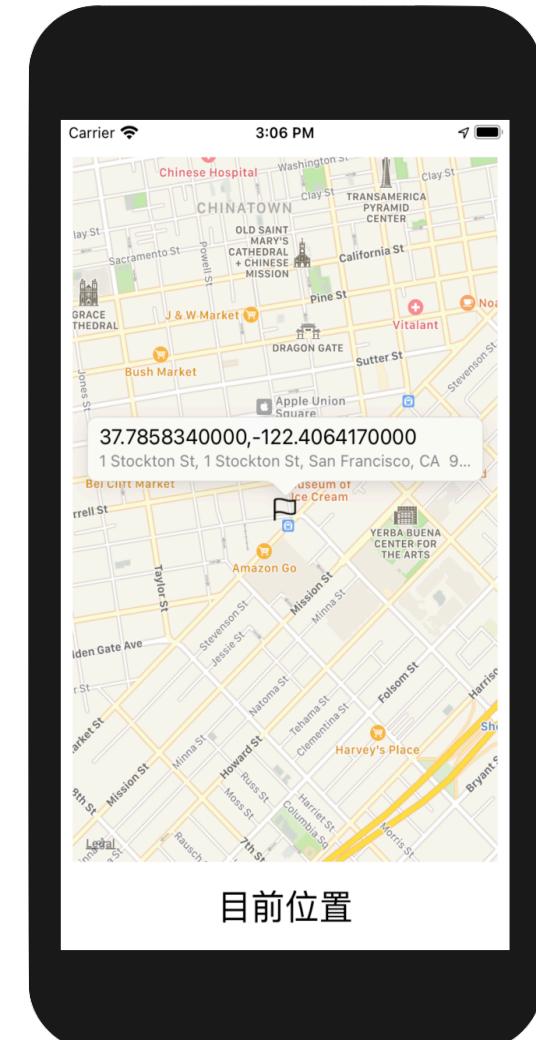
- 修改目前位置的標題與副標題

```
func updateUIView(_ view: MKMapView, context: Context){  
  
    let coordinate = mapView.userLocation.coordinate  
    let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)  
    let region = MKCoordinateRegion(center: coordinate, span: span)  
    view.setRegion(region, animated: true)  
    view.delegate = context.coordinator  
    mapView.userLocation.title = "目前位置"  
    mapView.userLocation.subtitle =  
        "\\\\(locationManager.location?.coordinate.latitude ?? 0.00),  
        \\\\(locationManager.location?.coordinate.longitude ?? 0.00)"  
}
```



經緯度轉換為地址

- 主標題
 - 經緯度
- 副標題
 - 轉換後的地址





MapView中的updateUIView

```
func updateUIView(_ view: MKMapView, context: Context){  
    let coordinate = mapView.userLocation.coordinate  
    let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)  
    let region = MKCoordinateRegion(center: coordinate, span: span)  
    view.setRegion(region, animated: true)  
    view.delegate = context.coordinator  
  
    let currentLocation = CLLocationCoordinate2D(latitude:  
        locationManager.location?.coordinate.latitude ?? 0.0, longitude:  
        locationManager.location?.coordinate.longitude ?? 0.0)  
  
    CLGeocoder().reverseGeocodeLocation(CLLocation(  
        latitude: currentLocation.latitude, longitude: currentLocation.longitude),  
        completionHandler: {(places, error) in  
            if error==nil{  
                mapView.userLocation.subtitle = places?[0].description ?? "Empty Add"  
            }else{  
                mapView.userLocation.subtitle = "No Address founded"  
            }  
        })  
  
    mapView.userLocation.title =  
        "\\"String(format:\"%.10f\",currentLocation.latitude)),  
        \\"String(format:\"%.10f\",currentLocation.longitude))"  
}
```



內建瀏覽器

- 開啟指定網頁
- 兩個分頁可切換





新增WebView.swift

- request變數：請求網址

```
import SwiftUI
import WebKit

struct WebView: UIViewRepresentable {

    let request: URLRequest

    func makeUIView(context: Context) -> WKWebView {
        return WKWebView()
    }

    func updateUIView(_ uiView: WKWebView, context: Context) {
        uiView.load(request)
    }
}
```



ContentView

- Picker與WebView連動

```
import SwiftUI
```

```
struct ContentView: View {  
  
    @State private var selectedSegment = 0  
    @State private var websites = ["morededev.tw", "developer.apple.com"]  
  
    var body: some View {  
  
        VStack{  
            Picker(selection: $selectedSegment, label:Text("網頁選擇")){  
                ForEach(0..<websites.count,id:\.self){  
                    Text(self.websites[$0]).tag($0)  
                }  
            }.pickerStyle(SegmentedPickerStyle())  
            WebView(request: URLRequest(url:  
                URL(string: "https://\\"+websites[selectedSegment] )!) ))  
        }  
    }  
}
```



裝置內建感應器

- CMMotionManager 類別
 - Magnetometer 磁力儀
 - Accelerometer 加速度計
 - Gyroscope 三軸陀螺儀
 - Device-motion 綜合



感應器支援

- 不怕，大家都有

iPhone 11 Pro Max

iPhone SE (第 2 代)

iPhone 6

感測器

三軸陀螺儀

三軸陀螺儀

三軸陀螺儀

加速度計

加速度計

加速度計

接近感測器

接近感測器

接近感測器

環境光度感測器

環境光度感測器

環境光度感測器

氣壓感測器

氣壓感測器

氣壓感測器



MotionManager.swift

```
import Foundation
import Combine
import CoreMotion
import UIKit

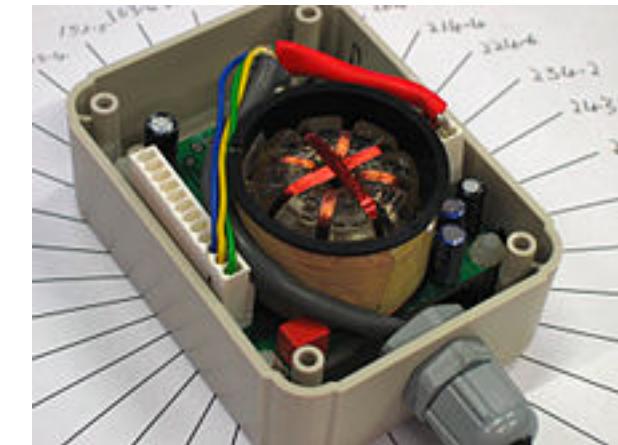
class MotionManager: ObservableObject {

    private var motionManager: CMMotionManager

    @Published var x: Double = 0.0
    @Published var y: Double = 0.0
    @Published var z: Double = 0.0

    init() {
        self.motionManager = CMMotionManager()
        self.motionManager.magnetometerUpdateInterval = 1/60
        self.motionManager.startMagnetometerUpdates(to: .main) {
            (magnetometerData, error) in
            guard error == nil else {
                print(error!)
                return
            }

            if let magnetData = magnetometerData {
                self.x = magnetData.magneticField.x
                self.y = magnetData.magneticField.y
                self.z = magnetData.magneticField.z
            }
        }
    }
}
```





ContentView

- 讀取X, Y, Z 資料 (弄塊磁鐵試試)

```
import SwiftUI
struct ContentView: View {
    @ObservedObject var motion: MotionManager
    var body: some View {
        VStack {
            Text("Magnetometer Data")
            Text("X: \(motion.x)")
            Text("Y: \(motion.y)")
            Text("Z: \(motion.z)")
        }
    }
}

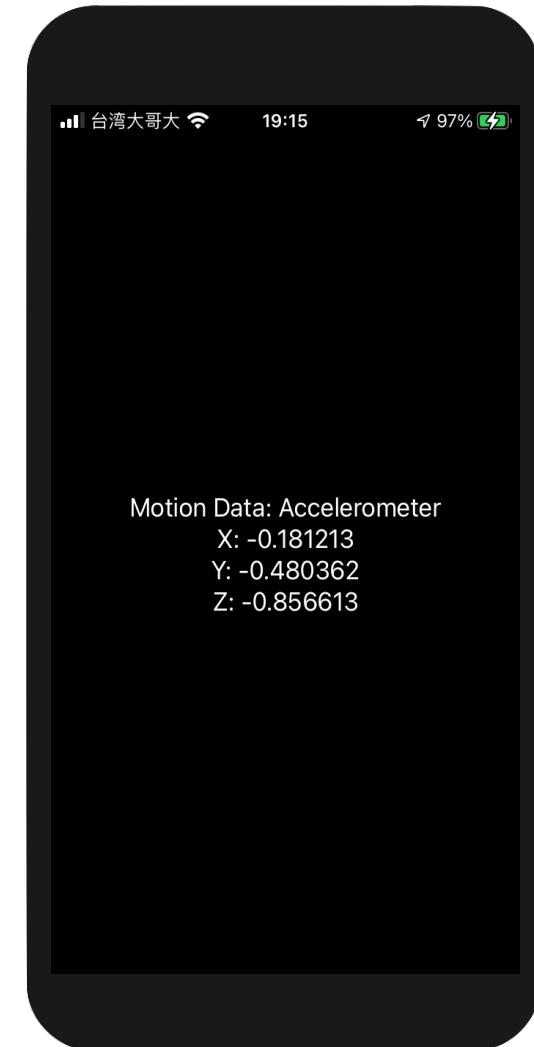
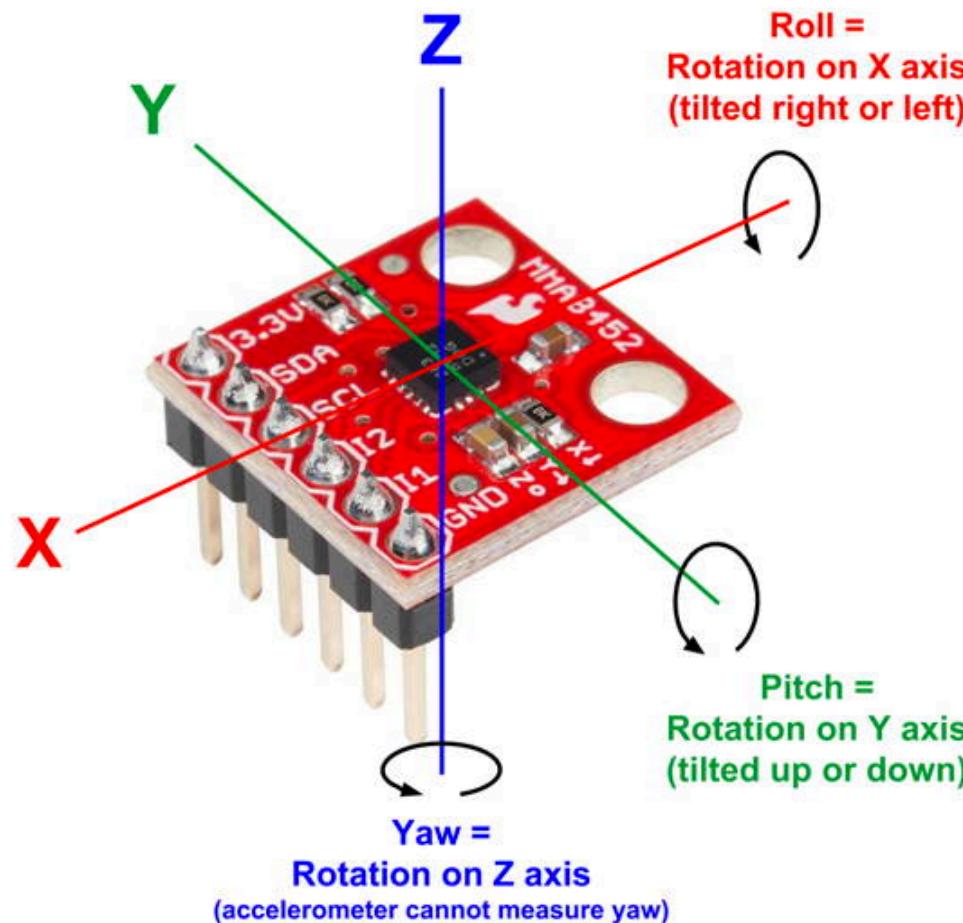
struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView(motion: MotionManager())
    }
}
```





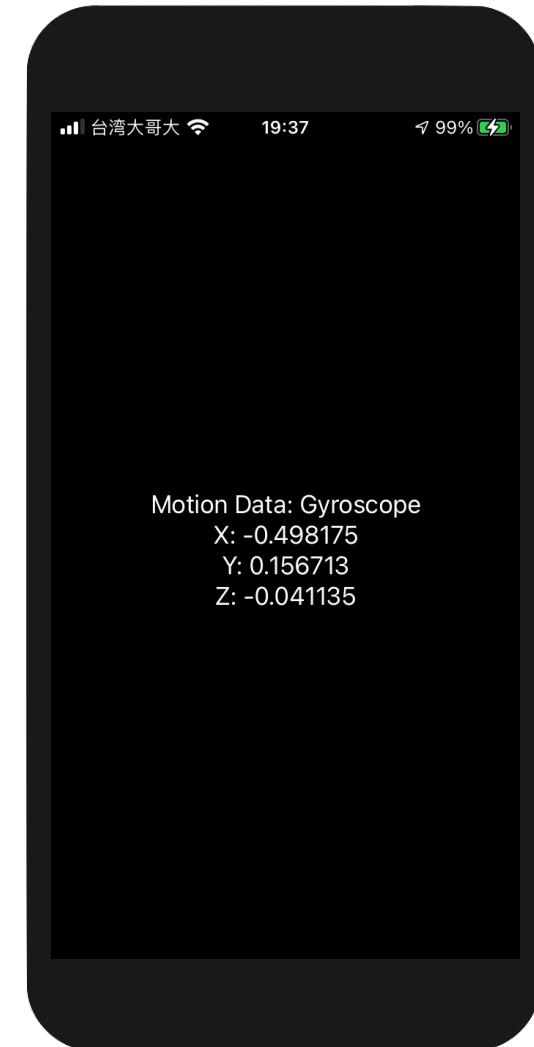
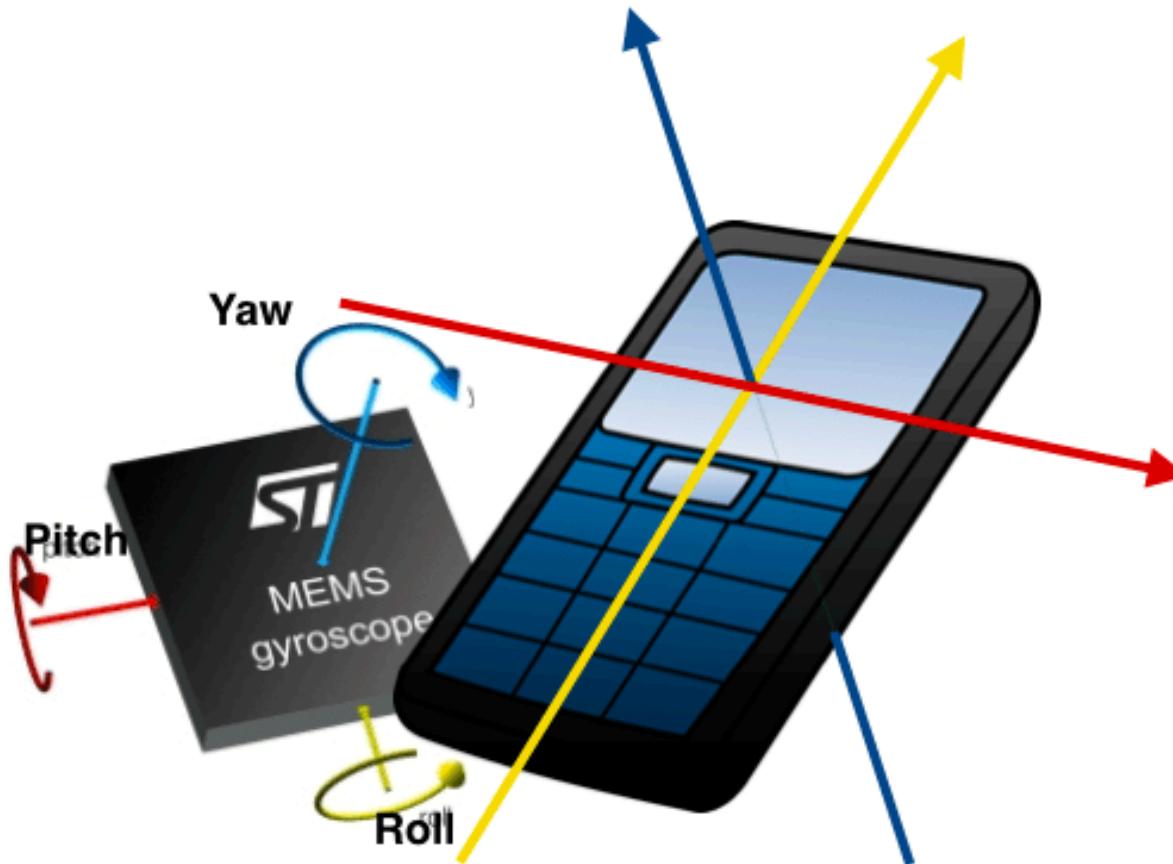
Accelerometer - acceleration

- 對應修改成其他感應器版本



Gyroscope - rotationRate

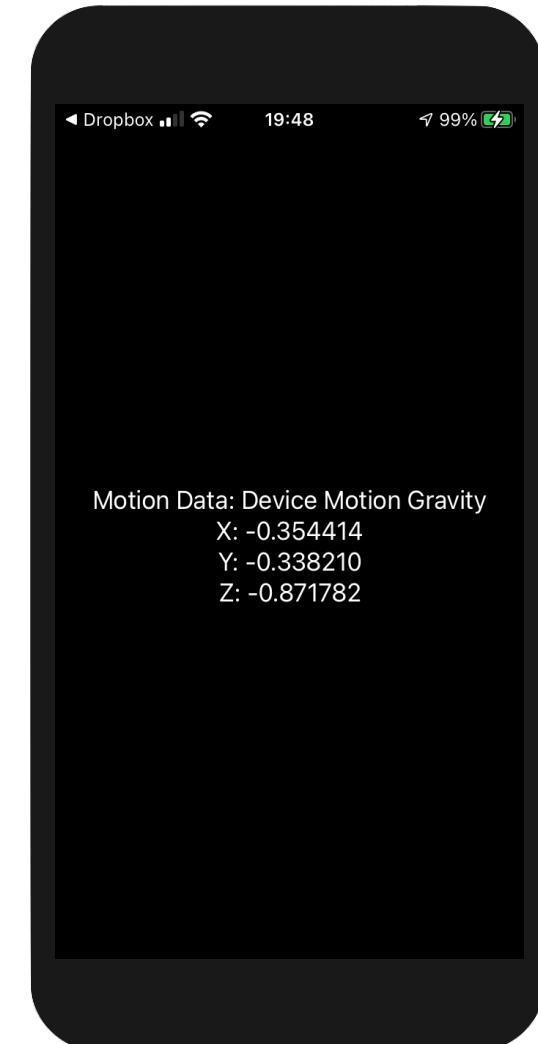
- 對應修改成其他感應器版本





DeviceMotion

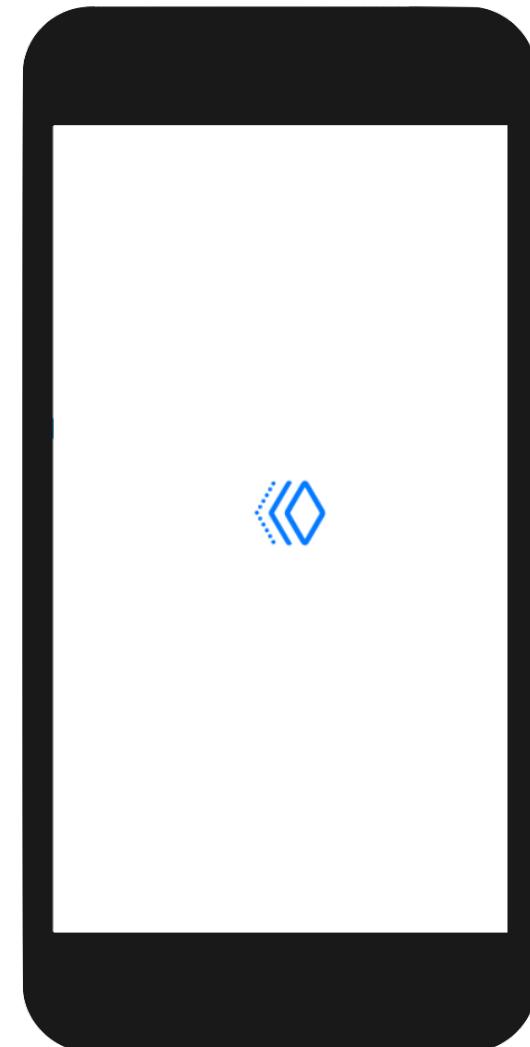
- X, Y, Z
 - gravity
 - userAcceleration
- attitude
- heading
- sensorLocation





震動

- 按下按鈕後，發出系統內建震動





ContentView.swift

- 在最下方加上Extension

```
struct ContentView_Previews: PreviewProvider {  
    static var previews: some View {  
        ContentView()  
    }  
}
```

```
extension UIDevice{  
    static func vibrate(){  
        AudioServicesPlaySystemSound(kSystemSoundID_Vibrate)  
    }  
}
```



ContentView

- AVFoundation

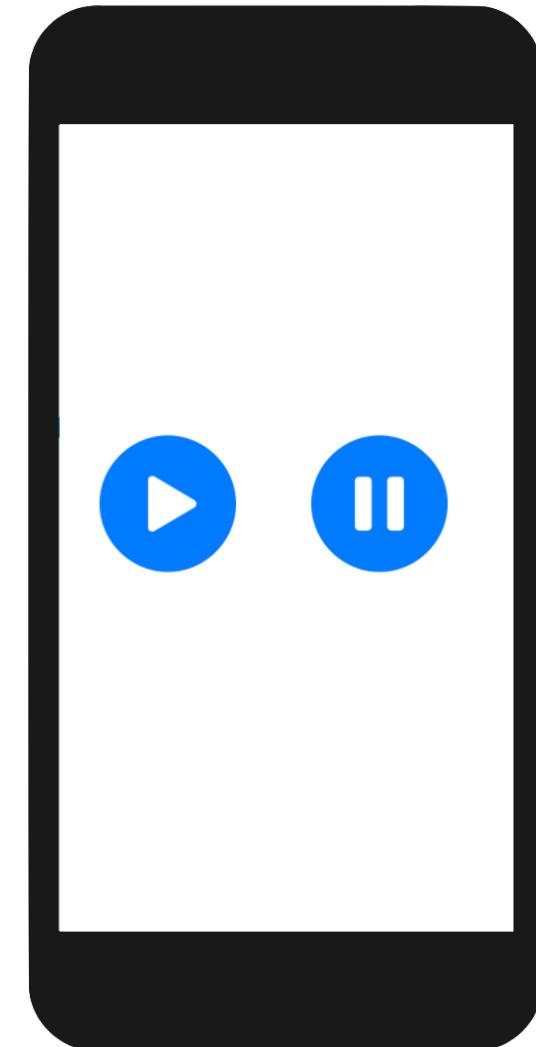
```
import SwiftUI
import AVFoundation

struct ContentView: View {
    var body: some View {
        Button(action: {
            UIDevice.vibrate()
        }) {
            Image(systemName: "square.stack.3d.down.dottedline")
                .resizable()
                .aspectRatio(contentMode: .fit)
                .frame(width: 50, height: 50)
                .foregroundColor(.blue)
        }
    }
}
```



音樂播放

- 兩顆按鈕
 - 播放音樂
 - 暫停播放音樂
- 實機限定





加入音樂檔案

- 專案圖案 -> Targets
- Build Phases -> Copy Bundle Resources
- + 輸入檔案

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "PlayMusic". It includes a file "sampleMusic.mp3" and a folder "PlayMusic" containing "PlayMusicApp.swift", "ContentView.swift", "Assets.xcassets", "Info.plist", "Preview Content", and "Products".
- Build Phases Tab:** The "Build Phases" tab is selected, showing the build configuration for the "PlayMusic" target.
- PROJECT Section:** Shows the project name "PlayMusic".
- TARGETS Section:** Shows the target name "PlayMusic".
- Build Phases:**
 - Dependencies (0 items)**
 - Compile Sources (2 items)**
 - Link Binary With Libraries (0 items)**
 - Copy Bundle Resources (3 items)**
 - "sampleMusic.mp3"
 - "Preview Assets.xcassets" ...in PlayMusic/Preview Content
 - "Assets.xcassets" ...in PlayMusic



ContentView

- AVKit, AVAudioPlayer, play(), pause()

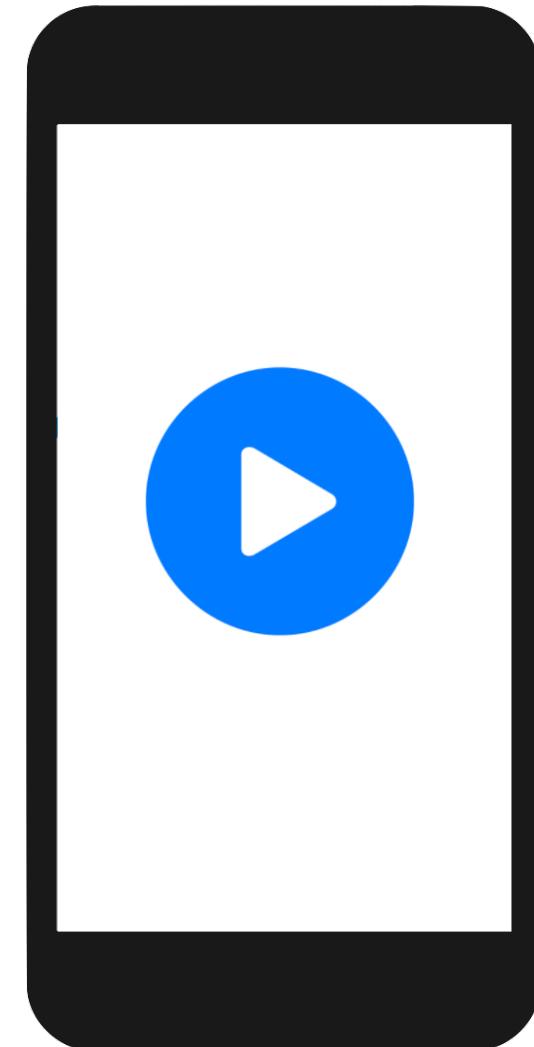
```
import SwiftUI
import AVKit

struct ContentView: View {
    @State var audioPlayer: AVAudioPlayer!
    var body: some View {
        HStack{
            Button(action: {
                self.audioPlayer.play()
            }) {
                Image(systemName: "play.circle.fill")
                    .resizable()
                    .frame(width: 100, height: 100, alignment: .center)
            }
            Spacer()
            Button(action: {
                self.audioPlayer.pause()
            }) {
                Image(systemName: "pause.circle.fill")
                    .resizable()
                    .frame(width: 100, height: 100, alignment: .center)
            }
        }.padding(.all, 60)
        .onAppear {
            let sound = Bundle.main.path(forResource: "sampleMusic", ofType: "mp3")
            self.audioPlayer = try! AVAudioPlayer(contentsOf: URL(fileURLWithPath: sound!))
        }
    }
}
```



重構：改成一顆按鈕

- 播放跟暫停同一顆按鈕
 - 按下開始播放，圖案變成暫停
 - 按下暫停播放，圖案變成開始
- 注意事項
 - 圖案與狀態相反
 - 一開始AVAudioPlay尚未初始化





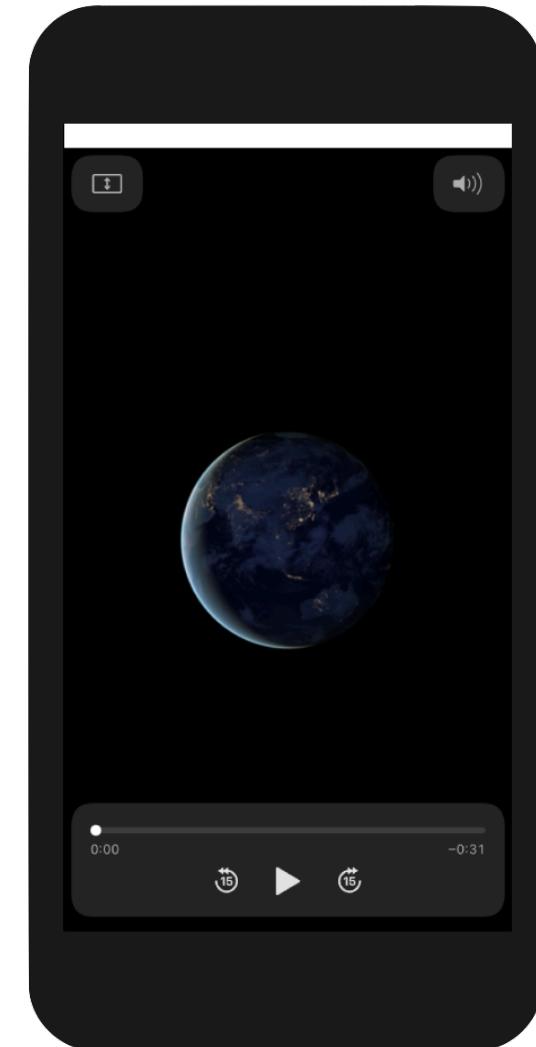
重構：改成一顆按鈕

```
struct ContentView: View {
    @State var audioPlayer: AVAudioPlayer!
    @State var isPlaying=false
    var body: some View {
        HStack{
            Button(action: {
                if audioPlayer.isPlaying{
                    audioPlayer.pause()
                    isPlaying=false
                }else{
                    audioPlayer.play()
                    isPlaying=true
                }
            }) {
                Image(systemName: isPlaying ? "pause.circle.fill" : "play.circle.fill")
                    .resizable()
                    .foregroundColor(isPlaying ? .green : .blue)
                    .frame(width: 200, height: 200, alignment: .center)
            }
        }.padding(.all, 60)
        .onAppear {
            let sound = Bundle.main.path(forResource: "sampleMusic", ofType: "mp3")
            self.audioPlayer = try! AVAudioPlayer(contentsOf: URL(fileURLWithPath: sound!))
        }
    }
}
```



影片播放

- 開啟影片播放器
- 播放指定影片





加入影片檔案

- 左邊導覽區->專案名稱->Build Phases

The screenshot shows the Xcode interface with the 'HelloVideo' project selected in the left sidebar. The 'Build Phases' tab is active in the top navigation bar. In the main area, the 'Copy Bundle Resources' section is expanded, listing three items: 'sample_video.mp4', 'Preview Assets.xcassets ...in HelloVideo/Preview Content', and 'Assets.xcassets ...in HelloVideo'. A red box highlights the '+' button located at the bottom left of this list, indicating where new resources can be added.



ContentView

- AVKit、VideoPlayer、Video Name

```
import SwiftUI
import AVKit

struct ContentView: View {
    var body: some View {
        VideoPlayer(player: AVPlayer(
            url: Bundle.main.url(
                forResource: "sample_video",
                withExtension: "mp4")!))
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```



播放網路影片

- URL(string:" ")

```
struct ContentView: View {  
    var body: some View {  
        /*  
         VideoPlayer(player:  
             AVPlayer(url:  
                 Bundle.main.url(  
                     forResource: "sample_video",  
                     withExtension: "mp4"))!))*/  
        VideoPlayer(player:  
            AVPlayer(url:  
                URL(string:  
                    "https://www.learningcontainer.com/wp-  
content/uploads/2020/05/sample-mp4-file.mp4")))  
    }  
}
```

