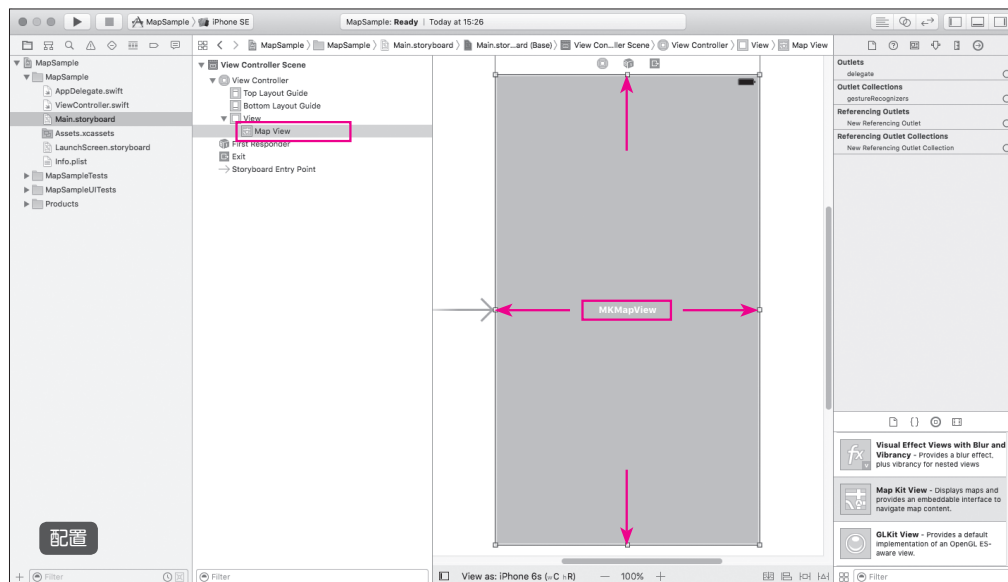
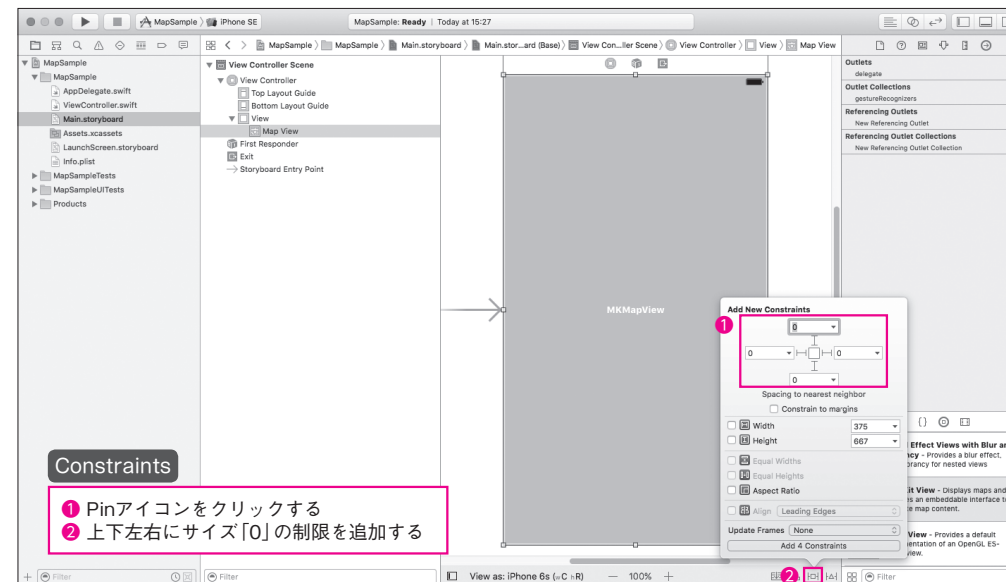


MKMapViewの4辺をドラッグし、画面一杯に拡大します。

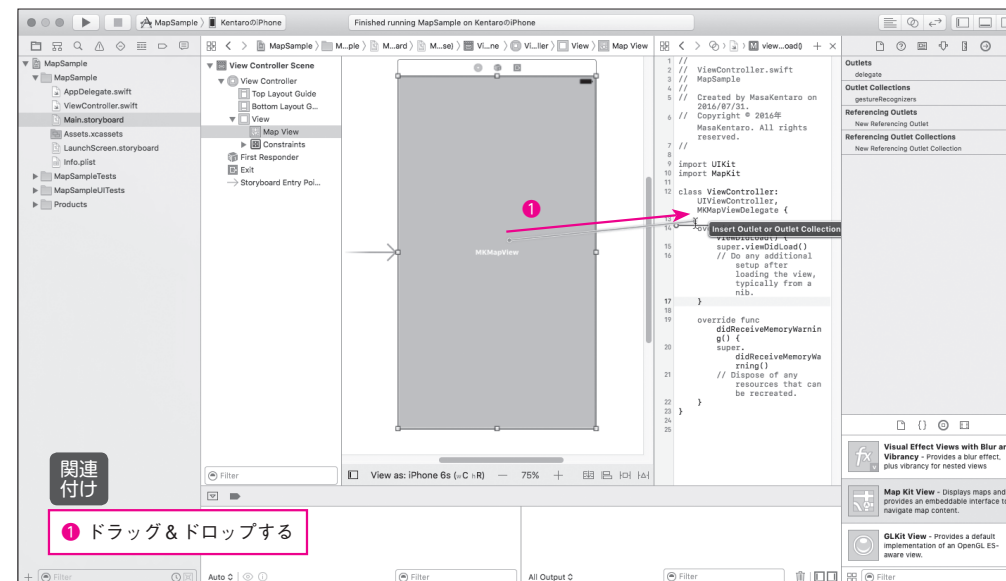


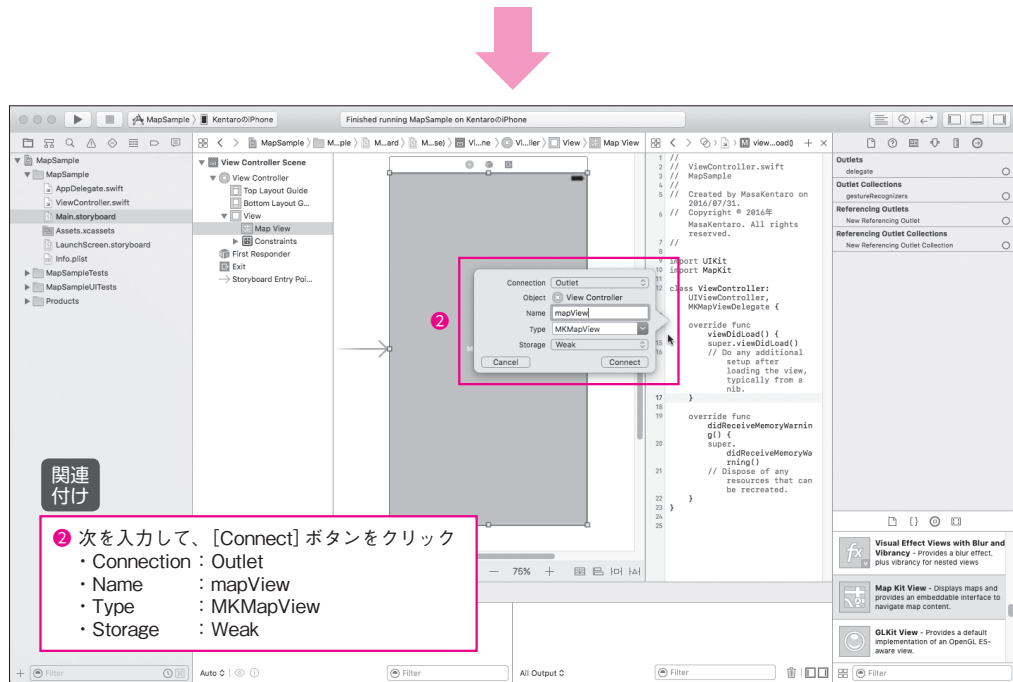
いろいろな画面サイズに対応するために Map View に Constraints の制限を追加します。



プログラムとマップビューを関連付ける

左のStoryboardの [Map View] を右のプログラムにドラッグ&ドロップします。このとき、ドロップする位置は次図のように、「override func viewDidLoad() {」の行のすぐ上にします。ドラッグ&ドロップすると小さなウィンドウが表示されるので次のように入力して [Connect] ボタンをクリックします。





右のプログラム側に、「@IBOutlet weak var mapView: MKMapView!」という1行が追加されれば成功です。

地図を表示するためのプログラムを追加する

ViewController.swift にリスト 5-1 のようなプログラムを追加します。

○リスト 5-1 : ViewController.swift

```
import UIKit
+ import MapKit ①

class ViewController: UIViewController {

+   var mapType: MKMapType = .standard ②
+
+   @IBOutlet weak var mapView: MKMapView! ③

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

+       let centerCoordinate = CLLocationCoordinate2D(latitude: 35.658581,
+       longitude: 139.745433) ④
```

(続く)

(続き)

```
+ let camera = MKMapCamera(lookingAtCenter: centerCoordinate,
+                           fromDistance: 700.0,
+                           pitch: 0.0,
+                           heading: 0.0) ⑤
+ mapView.setCamera(camera, animated: true) ⑥

}

override func didReceiveMemoryWarning() {

... 略 ...
```

- ① 地図を扱うプログラムのファイルの先頭に必要
- ② 地図の種類の初期値を「標準」に設定する
- ③ Storyboardに配置したマップビューと関連付ける
- ④ 地図に表示する地点の緯度・経度を指定する
- ⑤ どのような視点から地図を見るか、という制御のために擬似的なカメラを使う（高度700から垂直に見下ろしているところを想像してください）
- ⑥ マップビューに⑤で作成したカメラをセットする

ここで一旦アプリを実行してみましょう。編集したファイルを保存して、実行してみます。次のような地図が表示されます。



5-4 設定画面を追加する

iOSのマップビューには、便利なオプションがいくつも用意されています。このアプリでは、それらのいくつかを実際に試すために、設定画面を追加します。同時に、これを題材にある画面から別の画面への表示の変更（画面遷移）も練習します。

ツールバーを追加する

まずはじめに、地図画面から設定画面に画面遷移するための「設定」ボタンをツールバー