CoreLocation Exercise

Building on what we have seen during the explanation let's make an application that allows to get a suggested path to reach a destination from the current location

1. Prepare a view controller with a map in it using MapKit

   remember to have the view controller implement MKMapViewDelegate and set it as delegate for the map, you might need it later

2. Ask for user location and display his location

   you will need to use CLLocationManager and requestWhenInUseAuthorization(), your view controller should implement CLLocationManagerDelegate and respond to didUpdateLocation calls
   also remember to set NSLocationAlwaysUsageDescription in your .plist

3. Allow users to select a destination (i.e. using long taps)

   you will need to add a UILongPressGestureRecognizer to the map:
   when the gesture recognizer is triggered you can create a new MKPointAnnotation, set it's location and title and add it to the map with showAnnotation() or addAnnotation()

4. Compute a path between current location and selected destination

   For this task you can use MKDirectionRequest: you will need to provide to your request two MKMapItems with the coordinates for source and destination and the transportType (i.e. .Automobile) and finally call calculateDirections() with an appropriate completion

5. Display the path on the map

   each route returned from your direction request will contain a polyline that you can add to the map overlay but for this to work your view controller will have to return a MKOverlayRenderer on delegate calls to rendererForOverlay: this is easy as a MKPolylineRenderer is ready to be used, you just need to set properties such as color and width

Extras:

6. Show user-friendly address inside annotations

   you can use CLGeocoder and pass your CLLocation to reverseGeocodeLocation() with an appropriate completion block that changes the annotations title