iOS Assignment #2
Points of Interest

CS275 Fall 2021
15 points
due Wednesday, Dec. 1st, 11:59 pm

1 Points of Interest

Do the Bronze Challenge, on p. 112 of BNRG. You may do this assignment individually or with a partner.

1.1 Description of UI

Here's how you should modify the interface:

- put the switch 8 points below the segmented control
- put the label 8 points below the segmented control
- constrain the leading anchor of the label to the margins leading anchor
- constrain the switch leading anchor to the label trailing anchor + 12 points

Add the switch and the label and associated constraints in code, similar to the way that the `UISegmentedControl` and its constraints are added.

To complete this assignment, you'll first have had to go through Ch. 3-5 in BNRG.

You'll need to make these two calls in your code:

- `mapView.pointOfInterestFilter = MKPointOfInterestFilter(excluding: [])`
- `mapView.pointOfInterestFilter = MKPointOfInterestFilter(including: [])`

The first one turns on the points of interest, and the second turns off the points of interest on the map.

Name your action for the switch `poiDisplay(_:)`, so you should have this call:

```
poiSwitch.addTarget(self,
                    action: #selector(poiDisplay(_:)),
                    for: .valueChanged)
```

and this declaration:

```
@objc func poiDisplay(_ poiSwitch: UISwitch) { }
```

Set the state of the switch to "on" initially, because by default the map will show points of interest.

2 Locations

Add a button, below the Points of Interest label and switch, called Find Me. This should cause the map to zoom in to an interesting location (you can decide what location to zoom to).

Details:

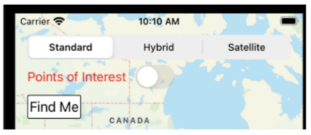
- put the button 8 points below the "Points of Interest" label
- constrain the leading edge of the button to the margins leading anchor
- set the border width for the button to be one point
- set the corner rounding for the button to be two points
- set the background color for the button to be `UIColor.white`

You can pad the button a little around its title by doing this:

```
let fmButtonWidthConstraint = fmButton.widthAnchor.constraint(
      equalToConstant: fmButton.titleLabel!.intrinsicContentSize.width + 2.0 * 3)
```

and then making that constraint active.

Here's what mine looks like:



Read about `UIButton`. You'll also need to read about `MKCoordinateSpan` and `MKCoordinateRegion`.

3 Graduate Students

Graduate students, and undergraduates who would like additional credit: make the Find Me button show where you are in the world. To do this, you'll have to get a location fix, using `CLLocationManager`. Have your map view controller conform to `CLLocationManagerDelegate`. You will have to add two entries to your info.plist:

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

Create an instance of `CLLocationManager` and start getting position fixes this way:

```
locationManager = CLLocationManager()
locationManager.delegate = self
locationManager.desiredAccuracy = kCLLocationAccuracyBest
locationManager.requestAlwaysAuthorization()
locationManager.startUpdatingLocation()
```

You should then implement `locationManager(_:didUpdateLocation)`. This function will be called whenever a new position fix is available. Retrieve the last location from the array that is passed to this function and save it in your own class property. You can then use it when the user touches the Find Me button.

Also, drop an `MKPointAnnotation` on the map, and delete the previous `MKPointAnnotation` marker (if there is one). You can tell the simulator to simulate various locations using the options in Features -> Location.

I've learned that there is a bug in iOS 15 simulators that prevents them from enabling location services. You'll have to build for iOS 14.x and then install an iOS 14.x simulator (if you don't already have one) and run on that simulator.

Graduate students must work individually.

4 What to Submit

If you have your project in github, then submit a link to your project. Otherwise, you can zip up your project, starting at the top level (so there should be a folder *ProjectName*, and that folder should have two subfolders: *ProjectName* and *ProjectName.xcodeproj*).