MapLocations

Description: Create an iOS Swift single view app that contains a map that fits the full screen and is centered on a specific geo-coordinate (latitude and longitude) and shows a specific area of the map (latitude delta and longitude delta). Once that is complete implement a map annotation to mark a location on the map and display information about the location when it is tapped.

Purpose: To learn how to work with a MapKit View and geo-coordinate data, add a framework to an Xcode project, use `import` and practice implementing a Custom Class for a UIViewController and an IBOutlet.

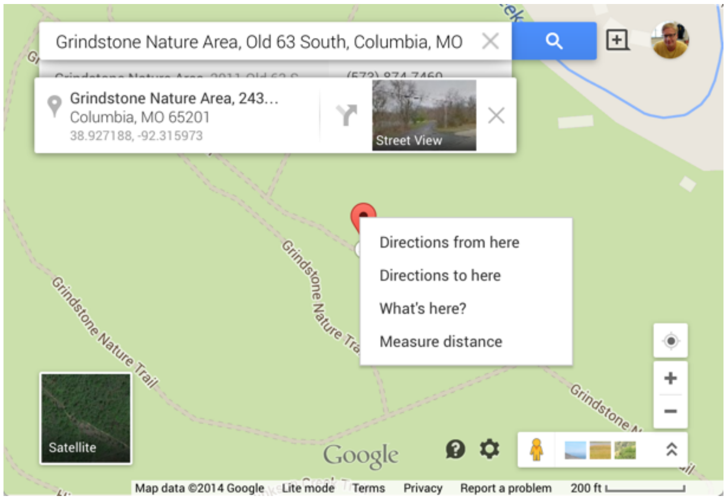
Requirements:

Project Name: MapLocations
Devices: Universal

For this part of the challenge implement a single view app in Swift that contains a map that fits the full screen and is centered on a specific geo-coordinate (latitude and longitude) and shows a specific area of the map (latitude delta and longitude delta).

Here is an example location you can use. It is centered on Grindstone Nature Area just south of the University of Missouri campus.
Latitude: **`38.927246`** Latitude: **`-92.315984`**
For the area of the map to display use the following deltas.
Latitude Delta: **`.03`** Longitude Delta: **`.03`**

You can use a different latitude and longitude and a different set of deltas if you want, but I'd prefer it wasn't a random choice. You can get geo-coordinates from Google maps. If you search to a location and then right-click (or ctrl-click on a Mac) on the location marker and choose "What's here?" it will display the latitude and longitude in an information box at the top of the page.

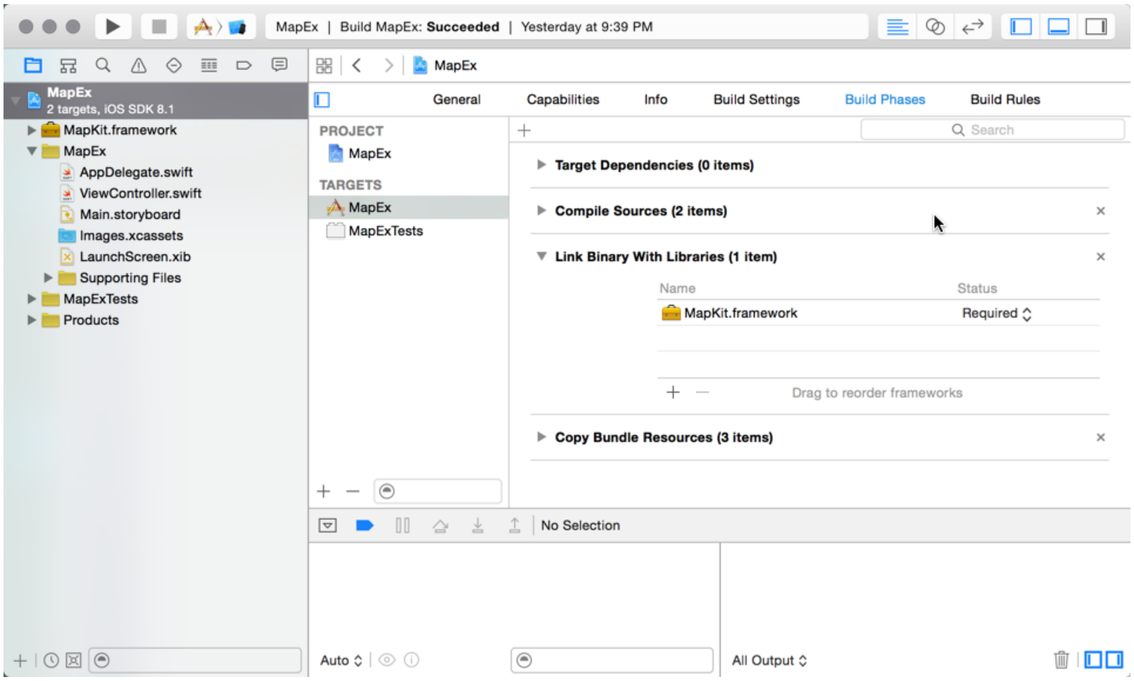


Grindstone Nature Area, Old 63 South, Columbia, MO

Grindstone Nature Area, 243...
Columbia, MO 65201
38.927188, -92.315973

MapLocations

To implement a map you will need to use MapKit. You will need to add MapKit to the project. You add MapKit in the Build Phases > Link Binary with Libraries area of the project in Xcode. See the screenshot below. Click the **+** to add a framework and then select the MapKit.framework from the list of available frameworks.



The UI element you need to place in the interface is a MapKit View (`MKMapView`). Auto Layout Constraints are to be set on the MapKit View so this it is pinned to all four edges of the Superview. Remember that the ability to do this is in the Editor > Pin menu.

To get a reference to the map in your code you will need to create an IBOutlet to the MapKit View. To use MapKit in your code you will need to use `import MapKit` at the top of the UIViewController Custom Class.

The following is sample code that will set the location and the amount of latitude and longitude displayed. This code along with the `import MapKit` and the IBOutlet to the MapKit View are placed in the Custom Class for the UIViewController that contains the MapKit View. The code below can be placed in the `viewDidLoad` method. NOTE: do not use the default ViewController Custom Class. Delete ViewController.swift and create your own properly named Custom Class and associate it with the UIViewController in Interface Builder.

```
var latitude:CLLocationDegrees = 38.927246
var longitude:CLLocationDegrees = -92.315984
var latDelta:CLLocationDegrees = 0.03
var lonDelta:CLLocationDegrees = 0.03
var span:MKCoordinateSpan =
MKCoordinateSpan(latitudeDelta:latDelta, longitudeDelta:lonDelta)
var location:CLLocationCoordinate2D =
CLLocationCoordinate2DMake(latitude, longitude)
var region:MKCoordinateRegion = MKCoordinateRegionMake(location,
span)
mapView.setRegion(region, animated:true)
```

In the above code `mapView` is the variable that is the IBOutlet to the MapKit View.

The region that is displayed is a combination of a location and a span. The span is comprised of a latitude delta and a longitude delta. Experiment with these values to see how they change the amount of the map you can see. Also experiment with choosing a location on Google maps from a web browser and setting that location in the code to verify it goes to the correct location.

Experiment in the simulator with moving the map around and pinching and zooming to see more or less area on the map. ***You can pinch and zoom in the simulator by holding down the option key while clicking and moving the mouse.***

Additional Requirements: Add an Annotation to the Map

Once you have the map displaying correctly and you have an understanding of the code for creating and using locations, spans, and regions, try to add an annotation to the map as a specific location and set a title and subtitle in the annotation. The title and subtitle you set should describe the location you have chosen.

Hint: look in the documentation for `MKPointAnnotation` and check out methods of the map view that provide a means for adding an annotation.

Yes, you are not being given a lot of information for this additional requirement. You are expected to use the documentation available to you as well as the code completion abilities in Xcode to figure this out.

Notes:

The way you are to name your Xcode project based on the Project Name and how to create a ZIP file for submission of your work on Blackboard is covered in the document “Xcode Project Naming and ZIP Files.pdf”.

The ZIP file of your Xcode project folder is to be submitted on Blackboard in the assignment submission that is included in the challenge.