Daniel Kattan
CSCI 4237
December 10, 2012

Final Project Report
It’s Always Five O’Clock!

Introduction

For my final project I created an app that gives meaning to the phrase ‘it’s always five o’clock’. The app tells you how long it will be until it it is five o’clock, or how long ago five was, and tell you several cities where it is five. You can then decide to see where some bars near you are located. Moving to a new location in the map will load the bars for that location.

Implementation

The app utilizes several frameworks. The MapKit framework is used to display the locations of the cities where it is currently five o’clock. A MapView is also used to display the locations of the bars near your current location, or any location that you pan to.

The bar information is pulled from FourSquare using their web based API. FourSquare returns a json object which iOS is able to parse using an NSJSONSerialization object. This object returns foundation classes like NSArrays, NSStrings, and NSDictionaries which are very easy to handle in iOS.

The timezone in which it is currently five is determined based on the time on the iOS device and the timezone the device is in. This is obtained from an NSDate object. The timezone algorithm works off the UTC offset of the device’s current timezone.

An NSDictionary of cities was created and stored locally on the device. Each city dictionary has the city name, the country it is located in and the longitude and latitude of the city. This information is used to create the list of cities in a TableView and to display those cities on the MapView as MapViewAnnotations.

When a user clicks the button titles “Show me the bars!” the app segues from the main view controller to the map view controller and loads the bars from the FourSquare data. The data is fetched using the device’s current location as represented by a longitude and a latitude. If the user pans to a new location on the map the device will request a new set of bars from FourSquare based on the longitude and latitude of the center of the map.

Problems

Most of Apple’s native APIs are quite straightforward. The MKMapKit framework is easy to work with, as is parsing JSON data. The biggest hurdle was figuring out how to get the data from FourSquare. Their web API was not very clear in what data would be returned and returned a large amount of useless data along with the bar information. I had hoped to use Yelp to get the data however the Yelp API was much harder to understand than the FourSquare and was significantly less open than FourSquare. I also hoped to have more information about each bar than FourSquare provided. In the callout for the MapViewAnnotation (pin on the map) for each bar I display the bar’s name and phone number if it is available.

The data fetch from the web happens in an asynchronous thread which made debugging a little trickier as multiple things could be happening in parallel.

Pro Tips

Use native iOS API for whatever is possible. In iOS 6.1, Apple is set to release native API to search for points of interest in MapKit which would be able to replace FourSquare’s data for this project. Using native API is significantly easier to debug and almost always works better than third party libraries.

Many of the basics of iOS programming can be learned by watching the iOS App Development class from Stanford University on iTunesU. There is full lecture on MapKit and CoreLocation that goes through all the API needed to work with the device location and display points of interest on a map.

Reusable Classes

For this project I created two custom classes for the MapViewAnnotations. One for the cities where it is five o’clock and one for the bars. These custom classes made it incredibly easy to generate the annotations to each map whenever the app needs new annotations.