

8: CAMPUS WALK I

DUE: NOON, OCTOBER 22

Description

The remaining assignments involve designing and implementing an app to map out different walking paths between buildings on the Penn State campus. The first part involves manipulating and annotating maps using the MapKit and CoreLocation frameworks. A significant aspect of this assignment is the design of the app. We will give you only an app definition statement and you must decide how to realize the required functionality while providing a good user experience. There are many possible ways you can do this. Spend time carefully designing the user interface, perhaps laying it all out on a storyboard before implementing any code.

Parts II and III of the app will provide additional functionality and user interaction, including plotting walking routes, giving step-by-step directions and allowing the user to set preferences. As always, good code organization (MVC) and robust code is expected. Assets will be provided for each campus building.

App Definition

The Campus Walk app provides the user with a map of campus and the ability to find (and plot) campus buildings and the ability to maintain (and plot) a list of favorite buildings. More specifically the app must support the following

1. Initial view is a map of campus centered approximately on the center of campus.
2. The user can view a list of building names (see provided assets), select one, and have it plotted on the map with a pin and the map centered on this building and zoomed in a bit from the default zoom level. The centering and zooming of the map should be animated so that the user sees the change in the map from its previous view. The list of building names should be easy to navigate around, e.g., should include a quick way to jump to the buildings beginning with the letter "W." Building names should be sorted alphabetically. The annotation for the building should be a colored pin with a callout containing the building's name. The user should be able to remove the pin.
3. The user can maintain (by adding and deleting) a list of favorite buildings. The user can choose to have the map show or hide all the pins on the map for these buildings. The pins should be a different color than used for any other kind of annotation. Pins displaying favorites should always be consistent with the user's current favorites list.
4. The user's current location should be plotted on the map. The user should have the option of selecting whether the map tracks their current location and heading.
5. The user should have the option of selecting the kind of map displayed (standard, satellite or hybrid).

Testing

Thoroughly test your app for all possible user interactions you can imagine, in all orientations and on all devices. Be sure to clean up any compiler or runtime warnings. Do not use any deprecated methods. Be sure user interaction is intuitive and not awkward. Adhere to the given location services permissions. You can toggle this setting in the privacy settings on the simulator (just like on a real device). Make sure that when your app is foregrounded it handles any change in settings appropriately.

8: CAMPUS WALK I

DUE: NOON, OCTOBER 22

Hints

1. Make sure to import MapKit in any Swift files that use the MKMapKit framework.
2. Be sure to add the appropriate entry to the info.plist file for location services.
3. An action sheet might be a nice way for the user to take some actions. Use an instance of UIAlertController specifying the action sheet style.
4. Embedding your main view controller in a navigation controller, even if you don't push any other view controllers, may be useful as it gives you a navigation bar on which you can place bar button items. A tool bar may also be integrated into your design. You'll need a place for several buttons over the next couple weeks. A tab bar is definitely not appropriate for this app.
5. When obtaining and checking for location services privileges, make sure to only ask for location permission while the app is in use. Your app must adhere to user preferences. If the user changes their permissions in their phone's privacy settings, your app must check for this and handle the change appropriately.
6. MapKit and CoreLocation both include many delegate functions. Make sure to read through the API documentation to find the appropriate functions to use in this project.

Troubleshooting

Set a custom location on the simulator so you can simulate your location on campus. If you run your app on a real device you can test out the support for showing and tracking user location.

Submission

Your submission should be pushed on the master branch. Be sure to verify that your project builds and remove all cruft and compiler warnings.