Praktikum Entwicklung von Mediensystemen mit Android und iOS 19.5.2011

Sommersemester 2011

Ziele

- Mit Tab-Bar-Anwendungen vertraut werden, die wesentlichen Klassen und Protokolle kennenlernen, die Navigation zwischen Tabs kontrollieren
- Eine Webseite in einem Tab anzeigen und in der Historie navigieren
- Eine Karte in einem Tab anzeigen, Position und Zoom-Level festlegen

Aufgabe 1

Implement a tab bar application. Start with the "Tab Bar Application" template. Have a look at MainWindow.xib and familiarize yourself with the objects it contains. Read the documentation on the corresponding classes, i.e. UITabBarController, UITabBar, and UITabBarItem. Also review the documentation of UIViewController.

Create a Tab Bar Application as follows:

- Make the ApplicationDelegate a delegate of the tab bar controller. Hint: This can be done with the "Connections Inspector" in MainWindow.xib. The AppDelegate should already conform to the UITabBarDelegate protocol (see header file). In the AppDelegate's implementation file add the protocol's "didSelectViewController" method and produce NSLog output as views are selected showing the selected tab index.
- Add a UISwitch to FirstView and a label that says: "Can select SecondView". If the switch is off, the second view should not be selectable. Hint: This can be achieved by implementing UITabBarDelegate methods.
- Create a new subclass of UIViewController called WebViewController, including the corresponding xib file. In the xib file's attribute inspector set "Bottom Bar" to "Tab Bar". This adjusts the metrics of the view, as it will be part of a tab bar. Drag a Web View control onto the view. Make it fill the entire available space. Add an outlet for the Web View control to the header file. In MainWindow.xib add a UIViewController as the third child of the Tab Bar Controller. It should automatically get a Tab Bar Item Child – set its Title to "Web". In the Identity Inspector, change the class to the just created WebViewController. In the WebViewController, override a suitable method to set some static content of the Web View:

  ```
  NSString *html = @"some HTML content in a single string";
  [webView loadHTMLString:html baseURL:nil];
  ```
- Now, go on to load a page from the Web. First, create a NSURL object for http://www.medien.ifi.lmu.de. Then create an NSURLRequest to load the page. Comment out the previous code to load static content.
- Add two buttons to WebViewController.xib to allow the user to go forward and back in the web page, respectively.
- Disable or enable each button depending on whether it is possible to go forward or backward. Also handle the case in which this condition

changes when a new page has been loaded (and hence it become possible to navigate backward).

- Now, add a map view to a fourth tab. First, create a new class MapViewController as a subclass of UIViewController, including a xib file. As before, set "Bottom Bar" to "Tab Bar" in the Attributes Inspector under Simulated Metrics. Drag a MapKit Map View onto the view. Create an outlet for the map view. (You have to #import <MapKit/MKMapView.h> in the header file.) The map view needs its own framework (= library), named MapKit.framework. (Xcode 3: add existing framework. Xcode 4 : select project, targets, build phases, link binary with libraries, +, MapKit.framework). In MainView.xib create a fourth tab for the map view. Make sure class of the new controller is MapViewController.
- When the map appears, set the center position of the map to (lat, lon)= (48.146988, 11.576108). Show a region centered on that position with a (Δlat, Δlon)=(0.005, 0.005). Use an animated transition.
- Add two text fields and corresponding labels to the second view – one for setting latitude and one for setting longitude. The coordinates entered should be used as the new default center coordinates for the map view.

Aufgabe 2 (optional)

Familiarize yourself with the autorotation property of the view controller and present your findings in class. You can show a few slides or step through source code to explain how a program can handle rotation changes.

Abgabe

- Geben Sie die Lösung bis zum 26.05.2011 – 23:59 Uhr mittels Upload im Team-Repository ab.
- Legen Sie im SVN-Repository ihres Teams einen Ordner „Nachname" an und legen sie alle Projektdaten dort drin in einen Ordner „Exercise02_[CIP-Kennung]".
- Laden Sie den XCode-Projekt-Ordner als zip-Datei hoch.
- Es ist nicht erlaubt die Aufgaben gemeinsam zu bearbeiten. Bitte beachten Sie dazu auch die Hinweise zu Plagiaten.
  http://www.medien.ifi.lmu.de/lehre/Plagiate-IfI.pdf