Praktikum Entwicklung von Mediensystemen (iOS) **25.5.2012**

SS 2012 **Prof. Dr. Michael Rohs**

Übungsblatt 2

Ziele

- Mit Tabbbed-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

Implementieren Sie eine Tabbbed-Anwendung. Beginnen Sie mit dem „Tabbed Application" Template. Wählen Sie „Use Storyboard" sowie „Use Automatic Reference Counting" aus. Wählen Sie „Device Family": iPhone. Schauen Sie sich die Datei „MainStoryboard.storyboard" und machen Sie sich mit den Objekten vertraut, die es enthält. Alle Screens („Scenes") sowie Übergänge dazwischen („Segues") sind im Storyboard dargestellt. Ein Doppelklick auf den Hintergrund zoomt heraus, so dass alle Scenes zu sehen sind. Ein Doppelklick auf eine Scene zoomt herein, so dass eine Szene verändert werden kann. (Achtung: Wenn nicht eingezoomt, können keine neuen Elemente zu einer Szene hinzugefügt werden.) Die initiale Szene hat einen von links eingehenden Pfeil und den Titel „Tab Bar Controller Scene". Lesen Sie die Dokumentation der zugehörigen Klassen, d.h. UITabBarController, UITabBar, und UITabBarItem. Lesen Sie außerdem die Dokumentation der Klasse UIViewController. Implementieren Sie die Tabbed-Anwendung wie folgt:

- Machen Sie die AppDelegate-Klasse zu einem „delegate" des Tab Bar Controllers. Deklarieren Sie dazu das Protokoll „UITabBarControllerDelegate" in der Klasse „AppDelegate" und fügen Sie in der Methode didFinishLaunching... folgendes ein:

  ```
  UITabBarController *tbc = (UITabBarController*)
                                self.window.rootViewController;
  tbc.delegate = self;
  ```

  Implementieren Sie nun die Methode „didSelectViewController" so, dass als Debug-Ausgabe (NSLog) der Index des aktuell ausgewählten ViewControllers angezeigt wird.
  Hinweis: Wenn der Cursor auf einem Klassennamen steht, wird rechts oben „QuickHelp" angezeigt, so dass man schnell einen Überblick über die Methoden und Properties der Klasse bekommen kann.
- Fügen Sie ein UISwitch-Element (samt IBOutlet im FirstViewController) zum FirstView hinzu, sowie ein Label mit der Angabe: „Zweiter Reiter ist auswählbar". Falls der Schalter aus ist, soll der zweite Tab nicht auswählbar sein. Hinweis: Dies kann durch Implementierung einer weiteren UITabBarDelegate-Methode erreicht werden.
- Erstellen Sie eine neue von UIViewController abgeleitete Klasse namens WebViewController. Ziehen Sie im Storyboard einen neuen „View

Controller“ („Objects library“, rechts unten) in das Storyboard, um eine neue Szene zu erstellen. Ändern Sie im „Identity Inspector“ die Klasse in „WebViewController“.Verknüpfen Sie die Szene „Tab Bar Controller“ mit der Szene „Web View Controller“ mit Hilfe eines Übergangs („Segues“). Dies kann durch ctrl-drag von der Szene „Tab Bar Controller“ auf die Szene „Web View Controller“ erreicht werden. Wählen Sie als Typ: „Relationship – viewControllers“. Im TabBarController erscheint daraufhin ein dritter Reiter. Ändern Sie den Titel in „Web“ (Web View Controller, Tab Bar Item). Ziehen Sie ein „Web View“ control aus der Bibliothek auf den View. Skalieren Sie es so, dass es den gesamten View (außer der Tab Bar am unteren Rand) ausfüllt. Fügen Sie im WebViewController ein IBOutlet für den Web View hinzu. Verknüpfen Sie dieses outlet mit dem Web View (connections inspector).
Hinweis: Dies geht besonders einfach, wenn Sie über die Werkzeugleiste rechts oben „Show the Assistant editor“ einschalten, so dass links das Storyboard und rechts die Header-Datei für den WebViewController zu sehen sind. Per Ctrol-Drag vom WebView in den Code kann dann das IBOutlet erstellt werden.

- Überschreiben Sie im WebViewController eine geeignete Methode, um statischen Inhalt zum Web View hinzuzufügen. Eine Möglichkeit, dies zu tun, lautet:

```
NSString *html = @"some HTML content in a single string";
[webView loadHTMLString:html baseURL:nil];
```

- Zeigen Sie nun statt statischen Inhalts wirkliche Webseiten an. Kommentieren Sie zunächst den vorherigen Code zum Anzeigen statischen Inhalts aus. Erzeugen Sie dann ein NSURL-Objekt für http://www.medien.ifi.lmu.de, sowie ein NSURLRequest-Objekt. Finden Sie eine geeignete Methode der Klasse UIWebView, um die Seite zu laden.
- Fügen Sie zwei Knöpfe zur „Web Scene“hinzu, um vorwärts und zurück zu gehen. Der WebView muss dazu etwas verkleinert werden. Erstellen Sie geeignete IBAction-Methoden, verknüpfen Sie die UIButtons damit und rufen Sie die entsprechenden Methoden im web view auf.
- Fügen Sie nun ein „map view“ als vierten und letzten Reiter hinzu. Erstellen Sie zunächst eine Klasse namens MapViewController, abgeleitet von UIViewcontroller. Erstellen Sie im Storyboard durch Ziehen eines „View Controllers“ aus der „Object Library“ eine neue Szene. Ändern Sie im „Identity Inspector“ die Klasse in MapViewController (die zuvor erstellte Controller-Klasse). Fügen Sie per Ctrl-Drag einen Übergang („Segue“) von der Tab Bar Controller Scene zur Map Scene hinzu. Am unteren Rand sollte automatisch ein weiteres Tab erscheinen. Ändern Sie den Tab Titel in „Map“.
- Ziehen Sie ein MapKit Map View auf den vorhandenen View. Erzeugen Sie ein IBOutlet für den map view. Verknüpfen Sie den map view mit dem outlet. Sie müssen nun eine Header-Datei importieren (#import <MapKit/MKMapView.h>) sowie die Bibliothek MapKit.framework einbinden (project (App-Name, ganz links oben im Project Navigator), targets, build phases, link binary with libraries, +, MapKit.framework).

- Setzen Sie die Mitte der Karte auf (latitude, longitude)= (48.137369, 11.576109) mit einer Ausdehnung von (Δlat, Δlon)=(0.005, 0.005). *Hinweis:* Verwenden Sie die Methode setRegion der Klasse MKMapView.
- **Für Master-Studierende:** Fügen Sie zwei Textfelder sowie Labels zum zweiten Reiter hinzu – eines zum Setzen des Breitengrads (latitude) und eines zum Setzen des Längengrads (longitude). Diese Koordinaten sollen dann die Mitte der Karte im vierten Reiter bestimmen. Dazu sollten im SecondViewController entsprechende outlets sowie das Protokoll UITextFieldDelegate deklariert werden sowie. Der Controller kann dann im Storyboard als delegate der beiden Textfelder verknüpft werden. Die Methode textFieldDidEndEditing erlaubt das Auslesen des Inhaltes des gerade editierten Textfeldes. Definieren Sie im AppDelegate properties zum Speichern der latitude und longitude, so dass diese Daten zwischen den ViewControllern ausgetauscht werden können. Auf den AppDelegate lässt sich wie folgt zugreifen (#import "AppDelegate.h" nicht vergessen):
  MyAppDelegate *delegate = (MyAppDelegate*)
  [UIApplication sharedApplication].delegate;
  NSString-Objekte können so in double-Werte konvertiert werden:
  NSString *latString = @"48.123"; double lat = [latString doubleValue];

Aufgabe 2 (für Master-Studierende)

Machen Sie sich mit der Funktionalität von „autorotation“ in der Klasse UIViewController vertraut. Beschreiben Sie kurz, wie die Anwendung auf eine Änderung der Geräte-Orientierung reagieren kann. Wie kann ein animierter Übergang in die neue Geräte-Orientierung erreicht werden? Wann macht es Sinn, die möglichen Orientierungen in der Anwendung einzuschränken (z.B. nur Portrait oder nur Landscape)? Welche Unterschiede gibt es in den Design Guidelines zwischen iPhone und iPad? Beschreiben Sie diese Aspekte und fügen Sie evtl. Screenshots und Illustrationen hinzu. (Länge: max. 1 Seite)

Abgabe

- Geben Sie die Lösung bis zum 1.6.2012 – 12:00 Uhr mittels Upload in UniWorX ab.
- Die Aufgaben sind von jedem Teilnehmer einzeln zu bearbeiten. Bitte beachten Sie dazu auch die Hinweise zu Plagiaten.
  http://www.medien.ifi.lmu.de/lehre/Plagiate-IfI.pdf