

# Übungsblatt 1

## Ziele

- Xcode aufsetzen, als Entwickler registrieren
- iOS Human Interface Guidelines kennenlernen
- Eine erste iOS-Anwendung erstellen
- Einsatz von Xcode und Storyboards erlernen
- Realisierbare Projektidee entwickeln

## Aufgabe 1

Registrieren Sie sich im Universitäts-Account der Medieninformatik. Schicken Sie dazu eine Email (mit der Emailadresse, die Sie als iOS-Entwickler verwenden wollen) an michael.rohs@ifi.lmu.de sowie, falls vorhanden, die Geräte-ID Ihres iOS-Geräts. Beispiel Geräte-ID: 0102be49e0047a42a8427d8d4632205cbb2650a3. Diese lässt sich über den Xcode-Organizer oder über iTunes herausfinden. Wir senden Ihnen daraufhin eine Einladung zum iOS Dev Center sowie ein Provisioning Profile. Laden Sie Xcode herunter und installieren Sie es inklusive der aktuellen iOS-Version und der Dokumentation.

## Aufgabe 2

Die „iOS Human Interface Guidelines“ [1] geben grundlegende Empfehlungen zur Entwicklung von iOS-Anwendungen und legen verbindliche Regeln für die Verwendung der einzelnen Benutzungsschnittstellen-Elemente fest.

- a) Lesen Sie die Kapitel 2 und 3 der „iOS Human Interface Guidelines“.
- b) Schauen Sie sich exemplarisch die Details der Beschreibung von zwei Benutzungsschnittstellen-Elementen Ihrer Wahl in Kapitel 7 an.
- c) (für Master-Studierende) Lesen Sie in Kapitel 5 die Seiten 47 bis 65.

[1] <http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf>

## Aufgabe 3

Erstellen Sie eine einfache Anwendung an Hand der nachfolgenden Schritte. Hierbei geht es in erster Linie darum, die Handhabung von Xcode kennenzulernen und grundlegende Möglichkeiten der GUI-Programmierung unter iOS kennenzulernen. Die Anwendung soll nur im Emulator ausgeführt werden.

- Erstellen Sie ein neues Projekt namens „HelloWorld“ vom Typ „Single View Application“. Verwenden Sie Storyboards, ARC und Git.

- Der Project Navigator auf der linken Seite zeigt die verschiedenen Dateien an, die zum Projekt gehören. Gehen Sie diese Dateien durch und versuchen Sie deren Funktion zu verstehen.
- Öffnen Sie das Storyboard und löschen Sie die vorhandene Szene (eine „Szene“ entspricht einem Displayinhalt). Das Storyboard sollte jetzt leer sein.
- Ziehen Sie aus der Object Library (rechts unten) einen „Navigation Controller“ auf das Storyboard. Markieren Sie den „Root View Controller“ (rechte Szene) und ändern Sie im Identity Inspector den Klassennamen von UIViewController auf ViewController. Die Klasse ViewController sollte bereits automatisch erstellt worden und im Project Navigator sichtbar sein.
- Fügen Sie ein Label und einen Round Rect Button dem Root View Controller (rechte Szene) hinzu, in dem Sie diese aus der Object Library ziehen.
- Ziehen Sie einen weiteren „View Controller“ auf das Storyboard. Ziehen Sie ein Label auf die neue Szene und vergeben Sie ein sinnvolles Label. Erstellen Sie für die neue Szene eine eigene Klasse (File | New | New File...), abgeleitet von UIViewController. Nennen Sie die neue Klasse „SecondViewController“. Wechseln Sie wieder in das Storyboard, markieren Sie die neue Szene und ändern Sie im Identity Inspector die Klasse in die neu erstellte Klasse „SecondViewController.“
- Verknüpfen Sie per Ctrl-Drag den zuvor erstellten Button mit der zweiten Szene. Definieren Sie den Übergang („Segue“) zwischen der „Root View Controller“ Szene und der zweiten Szene als „Push-Segue“.
- Lassen Sie die Anwendung nun im Emulator laufen. Bei Druck auf den Button müsste die zweite Szene sichtbar werden. Dort müsste ein Back-Button für die Rückkehr in die erste Szene auftauchen.
- Fügen Sie jetzt einen Round Rect Button der zweiten Szene hinzu. Bei Drücken dieses Buttons soll der Label-Text geändert werden. Es soll die Anzahl der getätigten Klicks angezeigt werden. Gehen Sie dazu wie folgt vor. Schalten Sie mit der Toolbarbutton rechts oben den Assistant-Editor ein. Führen Sie dann ein Ctrl-Drag vom Button auf die Klassendeklaration aus, bis folgendes Tooltip erscheint: „Insert Outlet, Action, or Outlet Collection“. Erstellen Sie für den Button eine Action. Erstellen Sie auf gleiche Weise ein Outlet für das Label. Diese Verknüpfungen zwischen UI-Elementen im Storyboard und Methoden bzw. Variablen im Quelltext können auch programmatisch erstellt werden, was aber mehr Tipparbeit bedeutet.
- Erzeugen Sie in der Action-Methode eine Logausgabe mit NSLog, wie in den Folien beschrieben.
- Fügen Sie der Klasse SecondViewController eine Instanzvariable (keine Property) hinzu, die die Anzahl der Buttonclicks speichert. Bei jedem Klick soll diese Variable inkrementiert werden. Instanzvariablen werden in Objective C übrigens automatisch auf 0 initialisiert. Benutzen Sie die Methode NSString stringWithFormat um in der Action-Methode die Anzahl der Klicks auszugeben.
- Fügen Sie eine dritte Szene hinzu, die von der zweiten aus über einen Button erreichbar ist. Die dritte Szene soll ihren eigenen ViewController

haben (Name: ThirdViewController). In der dritten Szene sollen sich ein Label und ein Slider befinden. Der Slider soll Werte von 0 bis 100 annehmen können (Attributes Inspektor). Definieren Sie eine Action-Methode für den Slider (mit Argument sender), die den aktuellen Wert ausgibt. Welche Rolle spielt hierbei das per Methode übergebene Argument „sender“? Welchen Datentyp hat der Wert? Welcher Formatstring sollte also verwendet werden? (Hinweis: mögliche Formatstrings sind (u.a.) „%d“, „%f“, „%@“)

- Geben Sie nun den aktuellen Wert auch im Label aus. Definieren Sie dazu diesmal kein Outlet, sondern vergeben Sie ein „Tag“ für das Label (im Attributes Inspector). Der Zugriff auf das Label erfolgt dann so:  
`UILabel *label = (UILabel*)[self.view viewWithTag:100];`

#### **Aufgabe 4**

Überlegen Sie sich in Ihrem Team ein praktikables Thema für Ihr Projekt. Die Themen sollen beim nächsten Treffen von jeder Gruppe kurz mündlich vorgestellt werden. Die Themen müssen dabei noch nicht vollständig ausgearbeitet sein. Beachten Sie folgende Aspekte:

- Was? Funktionalität
- Wer? Zielgruppe
- Wo? Wann? Mit wem? Szenario, Situation, Benutzungskontext
- Warum? Warum sollten Personen aus der Zielgruppe motiviert sein, Ihre App zu benutzen?
- Probleme? technische Machbarkeit, Implementierbarkeit
- Neuheit? ist das Konzept innovativ? existierende Apps?

Diskutieren Sie diese Aspekte für Ihr Thema.

#### **Abgabe**

- Geben Sie die Lösung (zu Aufgabe 3) bis zum 24.5.2012 – 12:00 Uhr mittels Upload in UniWorX ab.
- Die Aufgaben 1-3 sind von jedem Teilnehmer einzeln zu bearbeiten, Aufgabe 5 in der Gruppe. Bitte beachten Sie dazu auch die Hinweise zu Plagiaten. <http://www.medien.ifi.lmu.de/lehre/Plagiate-Ifl.pdf>