

Übungsblatt 1

Abgabe

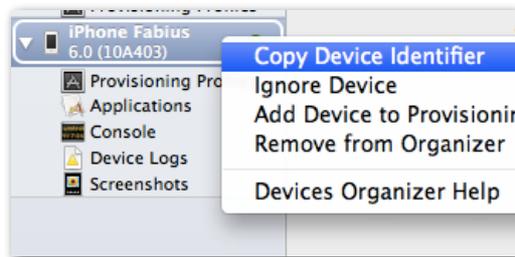
Geben Sie Ihre Lösung (Aufgabe 2 und 4) bis zum 24.4.2014 10:00 Uhr in Uniworx ab. Alle Aufgaben sind einzeln zu bearbeiten.

Aufgabe 1

Installieren Sie die neueste Version von Xcode über den Mac App Store und laden Sie ggf. Updates herunter, um die neuesten Versionen von APIs, Dokumentation und Simulator verwenden zu können.

Registrieren Sie sich im iOS Developer Center und schalten Sie Ihr iOS-Gerät zum Testen frei. Schicken Sie dazu **bis zum 24.4.2013 10:00 Uhr** eine Email an steinberger@cip.ifl.lmu.de mit dem Betreff "PEM iOS Developer Center" mit:

- Email-Adresse, mit der Sie als Developer registriert sein möchten (z.B. Ihre Apple-ID).
- Device-Identifizierer Ihres iOS-Geräts (iPhone, iPod Touch, iPad). Diese lässt sich über den Xcode Organizer herausfinden, wenn das Gerät am Computer angeschlossen ist (siehe Abbildung).
- Sollten Sie und/oder Ihr iOS-Gerät bereits im Medieninformatik-Team sein, dann vermerken Sie das bitte.



Ich trage dann Ihr Gerät ein und lade Sie ins Team ein. Sind Sie dem Team beigetreten, müssen Sie als nächstes ein Certificate Request durchführen. Dies erledigt Xcode für Sie (siehe "Request Signing Certificates" in der [Dokumentation](#)). Falls Sie damit Probleme haben, können Sie das auch manuell tun (iOS Developer Center Website > Member Center > Certificates, Identifiers & Profiles > Certificates > +). Anschließend kann ich Ihren Certificate Request bestätigen und Ihr Provisioning Profile erzeugen und an Sie schicken, das es Ihnen ermöglicht, eigene Apps auf Ihrem iOS-Gerät zu testen. Mit dem Developer Account können Sie außerdem auf weitere Ressourcen zugreifen, wie z.B. die [Developer Videos von Apple's Worldwide Developers Conference](#).

Fragen, Probleme? Nächstes Mal im Kurs zu mir kommen (am besten mit Laptop und iOS-Gerät).

Aufgabe 2

Erstellen Sie eine erste iPhone App anhand der folgenden Schritte. Hierbei geht es in erster Linie darum, die Handhabung von Xcode kennenzulernen und grundlegende Möglichkeiten der GUI-Programmierung unter iOS kennenzulernen.

Hinweise: Die folgenden Aufgaben können auch ohne Developer Account gelöst werden. Für dieses Übungsblatt können Sie Ihre App im Simulator testen.

- Erstellen Sie ein neues XCode-Projekt namens HelloWorld vom Typ Single View Application.
- Der Navigator auf der linken Seite zeigt die verschiedenen Dateien, die zum Projekt gehören. Gehen Sie diese Dateien durch und versuchen Sie deren Funktion zu verstehen.
- Öffnen Sie das Storyboard, dort finden Sie eine Scene (ein Displayinhalt). Markieren Sie die Scene und klicken Sie rechts auf den Identity Inspector. Dort sehen Sie, dass es sich hier um die Klasse ViewController handelt, die im Navigator sichtbar ist.
- Fügen Sie der Scene einen Button hinzu, indem Sie ihn aus der Object Library unten links ziehen. Fügen Sie außerdem ein Label hinzu und setzen Sie den Text auf “ViewController”.
- Ziehen Sie einen weiteren View Controller auf das Storyboard. Erstellen Sie dafür eine neue Klasse (File > New > New File) namens SecondViewController abgeleitet von UIViewController. Wechseln Sie wieder zum Storyboard, markieren Sie die neue Scene und ändern Sie im Identity Inspector die Klasse zu SecondViewController.
- Ziehen Sie ein Label (“SecondViewController”) auf die zweite Szene.
- Verknüpfen Sie per Ctrl-Drag den zuvor erstellten Button mit der zweiten Scene (Modal). Die Verknüpfung zwischen zwei Scenes wird als Segue bezeichnet.
- Lassen Sie die Anwendung nun im Simulator laufen. Durch den Button müsste man nun zur zweiten Scene gelangen.
- Ziehen Sie einen Button auf die zweite Scene und erstellen Sie eine Modal-Segue, so dass man über den Button zur ersten Scene gelangt.
- Ziehen Sie auf die zweite Scene ein weiteres Label sowie einen Stepper. Das Label soll zu Beginn 0 anzeigen, der Wert soll über den Stepper verändert werden. Gehen Sie dazu wie folgt vor.
- Blenden Sie den Assistant Editor ein, dieser zeigt den Code von SecondViewController.h an. Führen Sie ein Ctrl-Drag vom neuen Label zum Code durch und erstellen Sie dadurch ein Outlet für das Label. Erstellen Sie auf die gleiche Weise ein Outlet sowie eine Action (Value Changed) für den Stepper.

```
#import <UIKit/UIKit.h>

@interface SecondViewController : UIViewController

// UI elements
@property (weak, nonatomic) IBOutlet UILabel *stepperLabel;
@property (weak, nonatomic) IBOutlet UIStepper *stepper;

// UI actions
- (IBAction)stepperValueChanged:(id)sender;

@end
```

- Wechseln Sie zu SecondViewController.m und erzeugen Sie in der IBAction-Methode eine Logausgabe mit NSLog. Setzen Sie in der gleichen Methode den Labeltext auf den Stepper-Value. Benutzen Sie dafür die NSString-Methode stringWithFormat. Lassen Sie jetzt Ihre Anwendung laufen.

- **Für Master-Studierende:** Fügen Sie Ihrer Abgabe eine Datei namens Master.txt hinzu. Erstellen Sie eine dritte Scene und verwenden Sie ein weiteres Label sowie ein weiteres UI Element Ihrer Wahl (z.b. Slider). Erstellen Sie diesmal kein Outlet für das Label, sondern vergeben Sie ein Tag für das Label im Attributes Inspector. Der Zugriff auf das Label erfolgt dann über
`UILabel *label = (UILabel*) [self.view viewWithTag:100];`

Aufgabe 3

Die [Human Interface Guidelines](#) geben grundlegende Empfehlungen zur Entwicklung von iOS-Anwendungen und legen verbindliche Regeln für die Verwendung der UI-Elemente fest.

- a) Lesen Sie sich *Designing for iOS 7* und *Interactivity and Feedback* durch und verschaffen Sie sich einen Überblick über die restlichen Kapitel.
- b) Schauen Sie sich exemplarisch die Beschreibung eines UI Elements unter *UI Elements* an.

Aufgabe 4

Erstellen Sie eine iPhone App mit folgender Funktionalität.

- Beim Starten der Anwendung sieht der Benutzer eine Tabelle, in der jeder Eintrag für ein Festival Ihrer Wahl steht. (Oder Film, Fußballmannschaft, ... bleibt Ihnen überlassen.)
- Die Auswahl eines Tabelleneintrags führt zu einer weiteren Tabelle. Jeder Tabelleneintrag steht für eine Band des ausgewählten Festivals. Die Tabelleneinträge sollen ein Vorschaubild für die Band enthalten sowie einen Bandamen.
- Die Auswahl eines Tabelleneintrags liefert eine Detailansicht der ausgewählten Band mit einem Titel und einem Bild. Der Benutzer der Anwendung kann außerdem die Band bewerten (mit einem UI-Element Ihrer Wahl).
- Optional: Fügen Sie in der Detailansicht die Möglichkeit hinzu, mit der Kamera weitere Bilder hinzuzufügen.

Hinweise:

- Die Daten für die Table können Sie z.B. in einem Array oder Dictionary speichern. Der nächsten Tabelle bzw. der Detail View muss dann nur der jeweilige Eintrag übergeben werden (welcher wiederum ein Array oder Dictionary sein könnte).
- Kopieren Sie Ihre Bilder in Ihr Xcode-Projekt. Ansonsten ist Ihre App nur auf Ihrem Computer lauffähig.