

## **Ziele**

- Initiierung des Projekts
- iOS Human Interface Guidelines kennenlernen
- Verstehen fremden Programmcodes
- Die Location API kennenlernen
- Interaktionen mit Beschleunigungssensor und Gyroskop entwickeln

## **Aufgabe 1**

Dokumentieren Sie die Ergebnisse des Brainstormings (z.B. durch Einscannen der Ideenblätter). Einigen Sie sich innerhalb der Gruppe auf eine der drei vorgestellten Projektideen, die Sie im Praktikum umsetzen. Lösen Sie dann individuell die folgenden Aufgaben:

- a) Beschreiben Sie einen typischen Benutzer Ihrer Anwendung. Entwerfen Sie dazu eine „Persona“. Eine „Persona“ ist ein fiktiver aber typischer Nutzer, der die Zielgruppe repräsentiert. Dazu wird zunächst die Zielgruppe definiert und davon ausgehend typische Eigenschaften isoliert, die dann als „Persona“ beschrieben werden. In den Folien finden Sie einige weitergehende Informationen zum Thema „Personas“. Beschreiben Sie die entworfene Persona so knapp wie möglich.
- b) Überlegen Sie sich die Hauptfunktionen Ihrer Anwendung, sowie interessante Features und Entwurfsideen. Schreiben Sie diese als Liste knapp auf. Die Liste ist als Diskussionsgrundlage für das nächste Gruppentreffen (am 9.6.) gedacht, bei dem die Liste der Features dann innerhalb der Gruppe konsolidiert werden soll.
- c) Lesen Sie die Kapitel 2 und 3 der „iOS Human Interface Guidelines“ [1].

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

## **Aufgabe 2**

Laden Sie die „LocateMe“ Beispielanwendung herunter [2] und öffnen Sie sie in Xcode. Die Anwendung sollte idealerweise auf einem iPhone und nicht im Emulator ausgeführt werden. Wählen Sie daher bei über USB verbundenem Gerät Ihr iPhone als „Active Scheme“ aus.

- a) Analysieren Sie zunächst die Datei MainWindow.xib. Vergleichen Sie die Struktur (besonders innerhalb des Tab Bar Controllers) mit der Struktur Ihrer Tab-Bar-Anwendung aus der letzten Übung. Worin bestehen die Unterschiede und was wird durch die Struktur in „LocateMe“ zusätzlich ermöglicht?

- b) Begründen Sie, warum GetLocationView.xib einen TableView enthält und warum er in der xib-Datei nicht sichtbar ist. Erklären Sie grob die Schritte, die das Programm nach Drücken des Start-Buttons ausführt.
- c) Machen Sie sich mit der Funktionsweise des „Picker Controls“ in GetLocationSetupView.xib vertraut und erklären Sie die grundlegende Funktionsweise (Woher kommen die Daten, die im Picker zu sehen sind? Was bewirkt die Interaktion des Benutzers?). Sie müssen nicht alle Funktionen des „Picker Controls“ erklären, sondern nur die, die in der Controller-Klasse (SetupViewController.{h,m}) verwendet werden. Wozu wird das Dictionary „setupInfo“ in der Klasse SetupViewController verwendet? Wie werden die Daten vom SetupViewController zurück zum GetLocationSetupView transferiert? Hinweis: [obj respondsToSelector:@selector(method)] überprüft, ob das Objekt obj die Methode method implementiert.
- d) „CLLocation (Strings)“ ist ein Beispiel für eine Objective-C „Category“. Damit lässt sich die Funktionalität existierender Klassen durch Hinzufügen neuer Methoden erweitern (in diesem Fall CLLocation), ohne dass man (wie bei Vererbung) die existierenden Methoden der Klasse ändert. Categories können nur Methoden, aber keine Klassen- oder Instanzvariablen hinzufügen. Beschreiben Sie die Funktionalität der Category „CLLocation (Strings)“ und die Bedeutung der Localized Strings.
- e) Machen Sie sich mit der Verwendung des location managers in GetLocationViewController.{h,m} vertraut. Wie lange bleibt der location manager aktiv? Hinweis: [self performSelector:@selector(method) withObject:argument afterDelay:delay] führt nach „delay“ Sekunden die Methode „method“ von Objekt „self“ mit Argument „argument“ aus.

[1] <http://developer.apple.com/library/ios/#samplecode/LocateMe/Introduction/Intro.html>

### Aufgabe 3

Schreiben Sie eine einfache Anwendung, die eine Kompassnadel auf dem Bildschirm darstellt, die immer nach Norden ausgerichtet ist. Verwenden Sie die CMMotionManager-API aus dem CoreMotion-Framework (Framework einbinden!), um die dazu nötigen Sensorwerte zu erhalten. Geben Sie auch die Werte für pitch, roll, und yaw aus. Diese Begriffe werden in einer der Praktikums-Folien erklärt. Hinweis: Die Kompassnadel können Sie in der xib-Datei als in einem UIImageView hinzufügen (Mode = Center). Im Programmcode können Sie dann je nach yaw-Wert die Rotation des UIImageView wie folgt setzen:

```
CGAffineTransform t = CGAffineTransformMakeRotation(motion.attitude.yaw);
imageView.transform = t;
```

## Abgabe

- Geben Sie die Lösung bis zum 9.6.2011 – 14:59 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 „Exercise03\_[CIP-Kennung]“.
- Laden Sie über SVN ebenfalls den XCode-Projekt-Ordner 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>