

Praktikum iOS-Entwicklung im Sommersemester 2017

Übungsblatt 2

- Abgabetermin:** 23.05.2017, spätestens 23:59
(Gilt nur für die, die ihre Lösungen am Besprechungstermin vorstellen. Dies geschieht in Form einer ca. 5-minütigen Powerpoint-Präsentation. Die Abgabe muss daher die Präsentationsdatei enthalten (bitte per Email an kyrill.schmid@ifi.lmu.de.)
- Besprechung:** 24.05.2017

Aufgabe 3: (H) Adaptive Layout

Zur besseren Unterstützung unterschiedlicher Bildschirmauflösungen führt Apple mit iOS 8 das Paradigma **adaptiver Layouts** ein. Damit ist es beispielsweise nicht mehr notwendig, für iPhone und iPad jeweils ein eigenes Layout zu erzeugen. Das Design einer Benutzeroberfläche erfolgt fortan in ein und derselben View.

In dieser Aufgabe sollen sie für die in der Vorlesung vorgestellte App **MyQuiz** ein adaptives Layout erstellen. Eine gute Einführung zur Erstellung adaptiver Layouts finden sie unter

<http://www.raywenderlich.com/83276/beginning-adaptive-layout-tutorial>

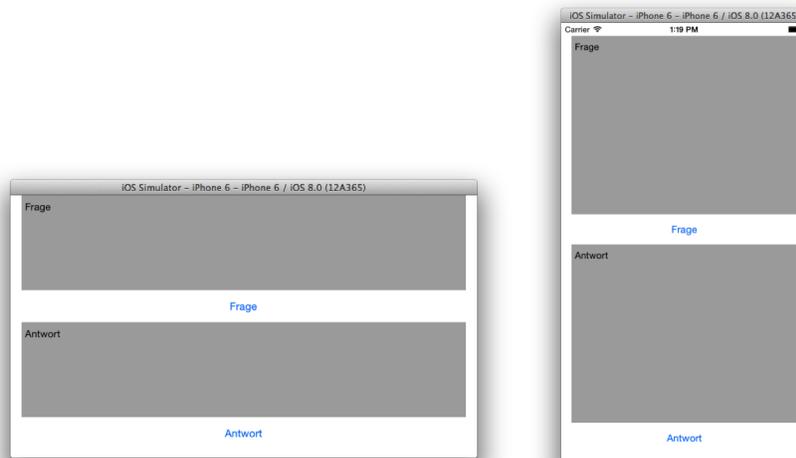
Halten sie sich beim Design an folgende Vorgaben:

- Alle UI-Komponenten sollen stets horizontal zentriert erscheinen.
- Die Buttons Frage und Antwort sollen stets die gleiche Höhe von jeweils 30 Punkten besitzen.
- Die Textfelder sollen für jede Bildschirmauflösung und -orientierung den gesamten noch zur Verfügung stehenden Platz ausfüllen. Dabei sollen die Textfelder Frage und Antwort stets die gleiche Höhe besitzen.
- Verwenden sie keine **Size Classes**.
- Achten sie darauf, dass die von ihnen eingesetzten Constraints keine Konflikte generieren.

Abbildungen 1 zeigt beispielhaft die gewünschte Darstellung auf einem iPhone 6.

Aufgabe 4: (H) Verwendung von Xib-Dateien

Bei der Entwicklung einer Benutzeroberfläche im Team bietet es sich an, die einzelnen Views einer iOS-Anwendung in jeweils einer eigenen Xib-Datei zu implementieren.



(a) Portrait

(b) Landscape

Abbildung 1: Unterschiedliche Spielzustände der Anwendung.

In dieser Aufgabe sollen sie die Benutzeroberfläche der in der Vorlesung vorgestellten **MyQuiz** mit Hilfe einer Xib-Datei in Objective-C implementieren.

Hinweis: Die dazu notwendigen Schritte für Objective-C entsprechen semantisch denjenigen, die in der Vorlesung anhand des Programmierbeispiels für Swift erklärt wurden