

## Praktikum iOS-Entwicklung im Wintersemester 2016/2017 Übungsblatt 3

- Abgabetermin:** 15.11.2016, spätestens 23:59 Uhr  
(Gilt nur für die Teilnehmer, die ihre Lösungen am Besprechungstermin vorstellen. Dies geschieht in Form einer ca. 5-minütigen Powerpoint-Präsentation. Die Abgabe muss daher die Präsentationsdatei enthalten und erfolgt über Uniworx.)
- Besprechung:** 16.11.2016

### Aufgabe 5: (H) Gradienten mit Core Graphics

In der Vorlesung haben Sie mit **UIKit** und **Core Graphics** zwei Möglichkeiten kennengelernt, individuelle zweidimensionale Views zu erstellen. Einige Funktionalitäten von Core Graphics wie z.B. das Zeichnen von Gradienten lassen sich jedoch nicht auf objektorientierte Art und Weise mit Hilfe der Klassen und Methoden von UIKit realisieren.

In dieser Aufgabe sollen Sie eine iOS-App mit Objective C bzw. Swift erstellen, die in einer eigenen View geometrische Figuren mit unterschiedlichen Gradienten visualisiert. Halten Sie sich bei der Lösung der nachfolgenden Aufgaben an folgende Vorgehensweise:

- Erstellen Sie mit Xcode ein Projekt vom Typ Single View Application.
  - Fügen Sie dem Projekt eine Klasse `GradientView` hinzu, die von der Klasse `UIView` erbt.
  - Überschreiben Sie die Methode `drawRect` der Klasse `GradientView`, indem Sie die entsprechenden Anweisungen zum Zeichnen der geometrischen Figuren implementieren.
  - Modifizieren Sie das automatisch generierte Storyboard, so dass es als Custom View eine Instanz der Klasse `GradientView` verwendet.
- a. Setzen Sie die iOS-App in Objective C unter Beachtung der oben genannten Vorgehensweise um. Die Methode `drawRect` der Klasse `GradientView` soll ein Dreieck mit einem Farbverlauf von blau nach rot visualisieren. Als Ergebnis soll Ihre Anwendung eine Anzeige generieren, die der aus Abbildung 1a entspricht.
- b. Setzen Sie die iOS-App in Swift unter Beachtung der oben genannten Vorgehensweise um. Die Methode `drawRect` der Klasse `GradientView` soll ein Rechteck mit einem Farbverlauf von gelb nach grün visualisieren. Als Ergebnis soll Ihre Anwendung eine Anzeige generieren, die der aus Abbildung 1b entspricht.

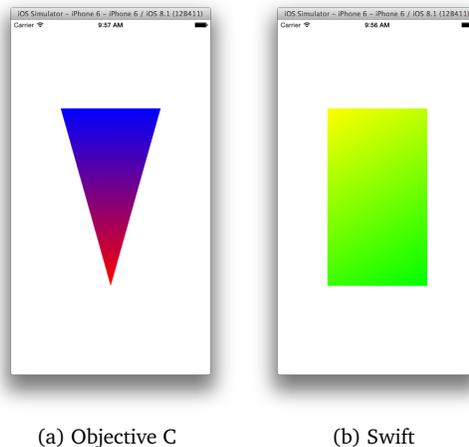


Abbildung 1: Darstellung der Anwendungen auf dem iPhone 6.

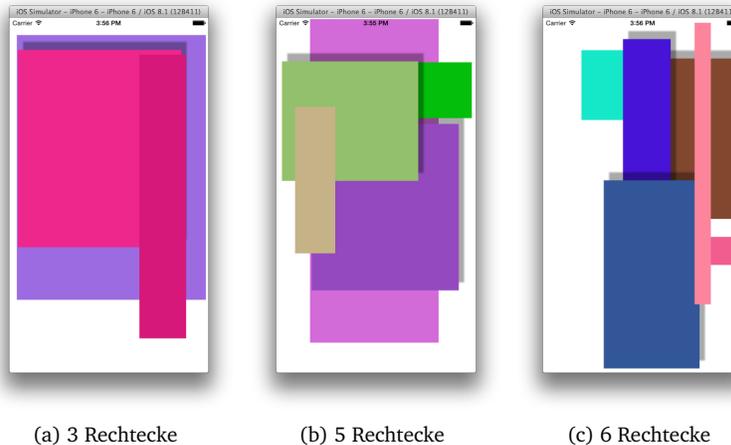
## Aufgabe 6: (H) Schattierungen mit Core Graphics

In dieser Aufgabe sollen Sie eine iOS-App mit Objective C bzw. Swift erstellen, die in einer eigenen View zufällig erzeugte Rechtecke mit und ohne Schattierung visualisiert. Halten Sie sich bei der Lösung der nachfolgenden Aufgaben an folgende Vorgehensweise:

- Erstellen Sie mit Xcode ein Projekt vom Typ Single View Application.
- Fügen Sie dem Projekt eine Klasse `ShadowView` hinzu, die von der Klasse `UIView` erbt.
- Überschreiben Sie die Methode `drawRect` der Klasse `ShadowView`, so dass diese zufallsbasiert zwischen 1 und 50 Rechtecke generiert. Verwenden Sie für Ihre Lösung ausschließlich Funktionen aus dem **Core Graphics Framework** (also nicht **UIKit** o.ä.). Implementieren Sie Ihre Lösung so, dass jedes Rechteck eine zufallsbasierte Position, eine zufallsbasierte Größe und eine zufallsbasierte Füllfarbe erhält. Achten Sie darauf, dass alle Rechtecke komplett innerhalb der durch die View vorgegebene Zeichenfläche liegen. Zudem soll jedes **zweite** erzeugte Rechteck einen Schatten generieren. Greifen Sie dazu auf die Funktion `CTContextSetShadow` zurück. Lagern Sie alle sich wiederholenden Schritte wie die Erstellung der zufallsbasierten Rechtecke, die Erstellung der zufallsbasierten Farben und das wiederholte Zeichnen der Rechtecke in geeignet Hilfsfunktionen aus.

**Hinweis:** Um den Grafikkontext Ihrer View zwischenzuspeichern, bietet sich die Verwendung der Funktionen `CGContextSaveGState` und `CGContextRestoreGState` an.

- a. Setzen Sie die iOS-App in Objective C unter Beachtung der oben genannten Vorgehensweise um. Als Ergebnis soll Ihre Anwendung Anzeigen generieren, die denen aus Abbildung 2 entsprechen.
- b. Setzen Sie die iOS-App in Swift unter Beachtung der oben genannten Vorgehensweise um. Als Ergebnis soll Ihre Anwendung Anzeigen generieren, die denen aus Abbildung 2 entsprechen.



(a) 3 Rechtecke

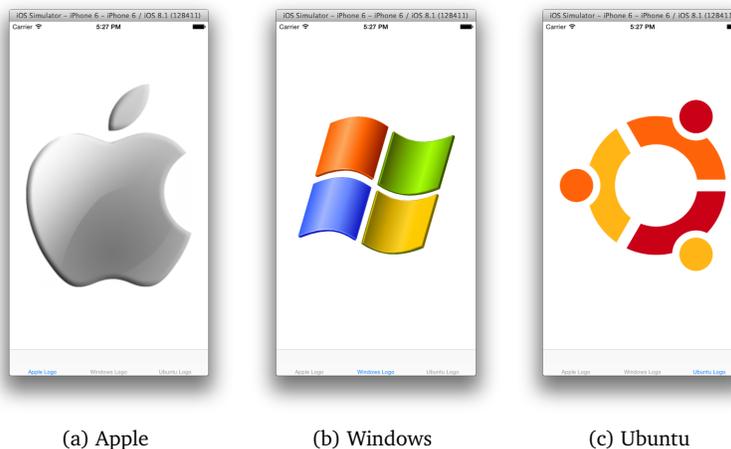
(b) 5 Rechtecke

(c) 6 Rechtecke

Abbildung 2: Darstellung der Anwendung für eine unterschiedliche Anzahl an Rechtecken auf dem iPhone 6.

## Aufgabe 7: (H) Tab Bar Controller

Erstellen Sie eine iOS-App, die unter Verwendung einer Instanz vom Typ `TabBarController`, die Logos von Apple, Windows und Ubuntu visualisiert. Verwenden Sie dazu eine Vorgehensweise Ihrer Wahl (z.B. programmatisch mit Swift oder Objective C, mit Storybord, mit Xib-Files, ...). Achten Sie bei der Umsetzung auf die Verwendung sinnvoller Layout-Constraints. Als Ergebnis soll Ihre Anwendung Anzeigen generieren, die der aus Abbildung 3 entsprechen.



(a) Apple

(b) Windows

(c) Ubuntu

Abbildung 3: Darstellung der Anwendungen auf dem iPhone 6.