Ludwig-Maximilians-Universität München Institut für Informatik Lehrstuhl für Mobile und Verteilte Systeme Prof. Dr. Claudia Linnhoff-Popien



Praktikum iOS-Entwicklung im Wintersemester 2016/2017 Übungsblatt 5

Abgabetermin:29.11.2016, 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:30.11.2016

Aufgabe 10: (H) Visualisierung von Multitouch-Sequenzen

In dieser Aufgabe sollen sie eine iOS-App zur Visualisierung einer Multitouch-Sequenz implementieren. Die Anwendung soll folgendermaßen funktionieren:

- Trifft ein Finger auf den Bildschirm, wird an dieser Stelle ein ein blauer Kreis erzeugt.
- Die nun folgenden Bewegungen des Fingers werden als Track aufeinanderfolgender roter Kreise visualisiert.
- Verlässt der Finger den Bildschirm, wird der zugehörig Track vom Bildschirm entfernt

Die Anwendung soll zduem die Visualisierung einer beliebigen Anzahl von interagierenden Fingern unterstützen.

Abbildung 1 illustriert beispielhaft einige aufeinanderfolgende Zustände der Anwendung. Abbildung 1a zeigt die Anwendung, während sich zwei Finger gleichzeitig auf dem Bildschirm nach unten bewegen. In Abbildung 1b wurde der rechte Finger vom Bildschirm entfernt. Der linke Finger bewegt sich weiterhin abwärts. In Abbildung 1c bfindet sich auch der rechte Finger wieder auf dem Bildschirm und bewegt sich ebenfalls abwärts.

Gehen sie bei der Umsetzung der Anwendung folgendermaßen vor:

- Implementieren sie die Anwendung ohne Storyboard / Xib-Datei programmatisch mit Swift.
- Implementieren sie die Logik zur Behandlung der Multitouch-Sequenz in einer eigenen View.
 Überschreiben sie dazu die Methoden drawRect, touchesBegan, touchesMoved, touchesEnded und touchesCancelled.



Abbildung 1: Darstellung der Anwendungen in unterschiedlichen Zuständen auf dem iPhone 4s.

Aufgabe 11: (H) Custom Gesture Recognizer

In dieser Aufgabe sollen sie eine iOS-App entwickeln, die die Rotation eines Images auf der Basis einer eigenen Geste unterstützt. Erkannt werden sollen die beiden in Abbildung 2 dargestellten Gesten.



Abbildung 2: Darstellung der Gesten, die zur Rotation des Images führen sollen.

Folgende bedingte Abfolge einer Touch-Sequenz wird als Geste zur Einleitung der Rechtsrotation erkannt:

- Genau ein Finger trift auf das Display.
- Der Finger bewegt sich kontinuierlich mindestens 25 Punkte nach rechts. Relativ zum Punkt der ersten Berührung darf sich der Finger hier auch nach unten jedoch nicht nach oben bewegen.
- Nach Erfüllung der vorhergehenden Bedingung bewegt sich der Finger mindestens 25 Punkte nach unten. Der Finger darf sich dabei weiterhin nach rechts, jedoch nicht nach links bewegen.

Die bedingte Abfolge einer Touch-Sequenz zur Erkennung der Geste, die eine Linksrotation einleitet, erfolgt analog. Halten sie sich bei der Umsetzung der Anwendung an folgende Anforderungen:

- Implementieren sie die Anwendung mit Swift.
- Implementieren sie ihren Gesture Recognizer als Klasse RotateImageRecognizer, welche von der Klasse UIGestureRecognizer erbt. Überschreiben sie dazu die Methoden touchesBegan, touchesMoved, touchesEnded, touchesCancelled und reset. Verwenden sie dabei die gererbte Property state zur Verwaltung der Phasen.

Hinweise: In der ursprünglichen Objective-C Klasse UIGestureRecognizer handelt es sich bei state um eine read-only Property. Um die Property in ihrer Implementierung verändern zu können, müssen sie diese mit Hilfe eines Bridging-Headers neu Deklarieren. Erzeugen sie dazu in ihrem Projekt eine Objective-C Klasse mit dem Namen Bridging-Header. Beantworten sie dabei die Frage, ob sie einen Bridging-Header erstellen wollen mit ja. Fügen sie in der Datei RotateImageRecognizer-Bridging-Header.h das Import-Statement

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

hinzu und löschen sie die Datei Bridging-Header.m.

Achten sie bei der Darstellung des Images darauf, dass es für alle möglichen Rotation (0°, 90°, 180° und 270°) immer die gleiche jedoch maximale Größe besitzt. Das Image soll stets komplett, unverzerrt und zentriert dargestellt werden. Verwenden sie dazu geeignete Layout-Constraints. Abbildung 3 veranschaulicht beispielhaft das gewünschte Ergbnise



Abbildung 3: Darstellung der Anwendung in seinen unterschiedlichen Zuständen auf dem iPhone 6.