

# Praktikum iOS-Entwicklung

Wintersemester 2017 / 2018

Prof. Dr. Linnhoff-Popien

Kyrill Schmid, Markus Friedrich

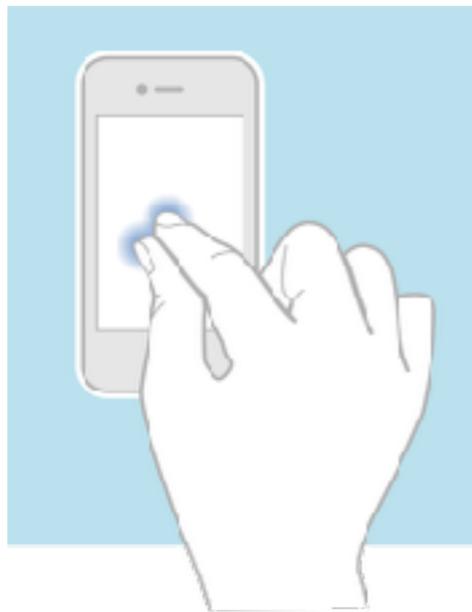


# Events

Der Nutzer kann durch Interaktion mit seinem Gerät verschiedene Events in iOS auslösen

Es gibt

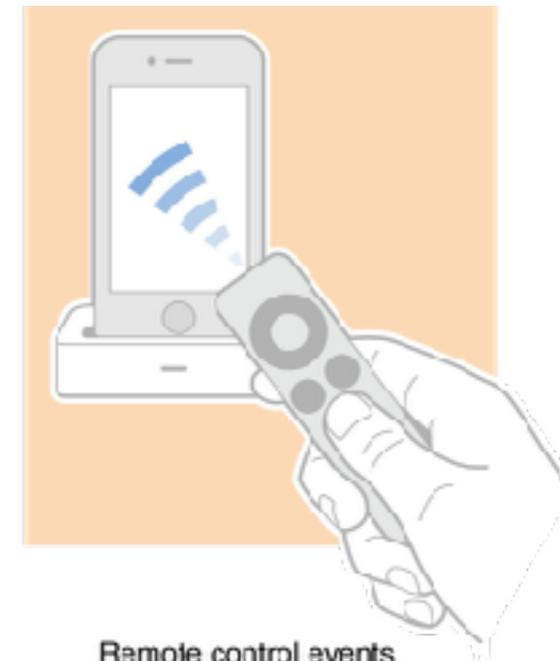
- (Multi-)Touch-Events
- Motion-Events
- Remote-Control-Events (zur Bedienung von Multimedia-Komponenten)



Multitouch events



Accelerometer events



Remote control events

# Events

Events sind Objekte, die an eine Anwendung gesendet werden, um sie über Interaktionen eines Nutzers zu informieren

Events, die durch den Nutzer ausgelöst werden, sind Instanzen der Klasse `UIEvent`

- enthalten Informationen, um auf das Event entsprechend zu reagieren
- haben einen Typ, z.B.
  - touch
  - shaking motion
  - remote control
- und einen Subtyp, z.B. für remote control
  - play
  - pause
  - stop
  - ...

# Gesten

Behandlung einzelner Berührungen der Finger mit dem Display (**Touch-Events**) ist in vielen Fällen viel zu aufwendig

Viele Interaktionen erfolgen nach dem gleichen Schema (Gesten)

- Tippen (Tap)
- Wischen (Swipe)
- Klammern (Pinch)
- Lange Drücken (LongPress)
- ...

→ Verwendung einer "High-Level-API" zur Behandlung von Gesten

# Gesten

Geste = Kombinationen/Abfolgen von Touch-Events

Intuitive Gesten werden von iOS über "Gesture Recognition" direkt erkannt

- Tippen (`UITapGestureRecognizer`)
- Pinchen (`UIPinchGestureRecognizer`)
- Wischen (`UISwipeGestureRecognizer`)
- Ziehen/Verschieben (`UIPanGestureRecognizer`)
- ...

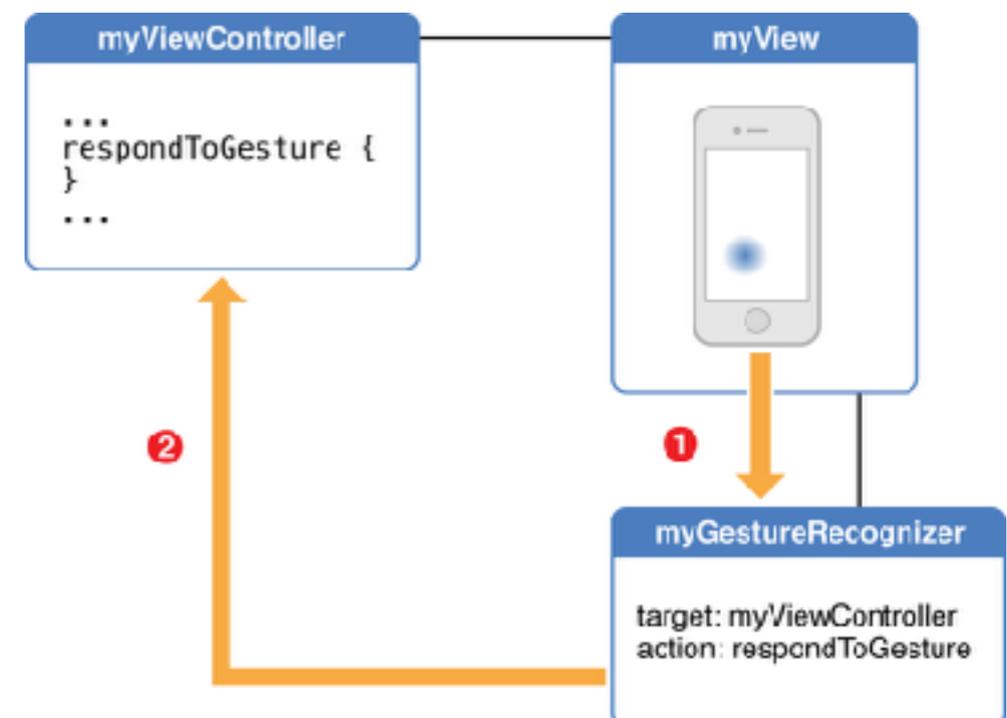
Für viele Gesten gibt es in Kombination mit typischen UI-Elementen eine direkte Integration in `UIKit`:

- Beispiel: `UIScrollView`
  - Pinch-Geste → Zoom
  - Wisch-Geste → Scroll

# Behandeln von Events mit Gesture Recognizern

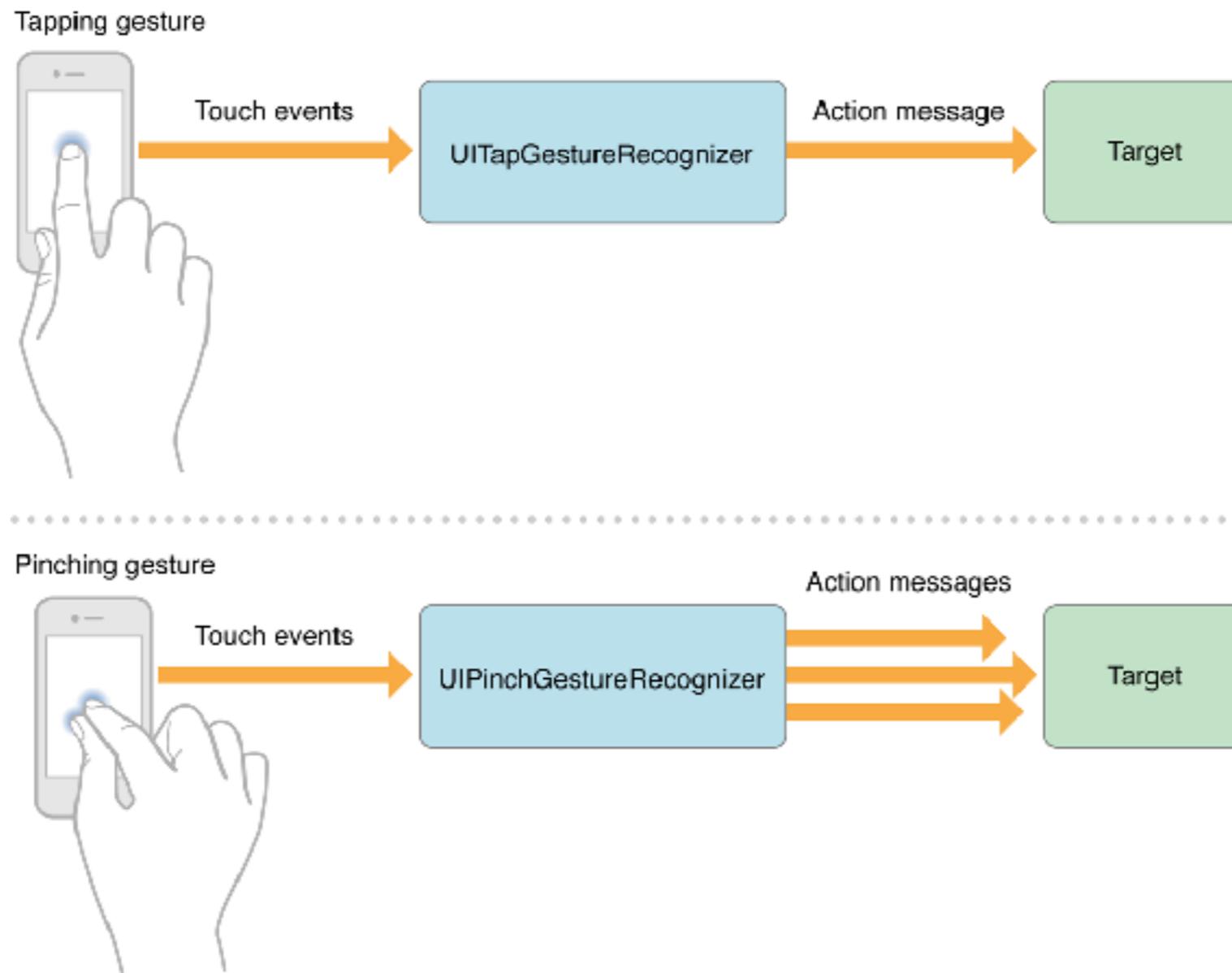
Hinzufügen eines Built-In Gesture Recognizers

- Erzeugen und konfigurieren einer Instanz von `UIGestureRecognizer`
  - Zuweisen eines Targets und einer Action
  - Zuweisen gestenspezifischer Attribute (optional; z.B. `numberOfTapsRequired`)
- Registrieren des Gesture Recognizers mit einer View
- Implementieren der Action-Methode



# Unterscheidung von Gesten

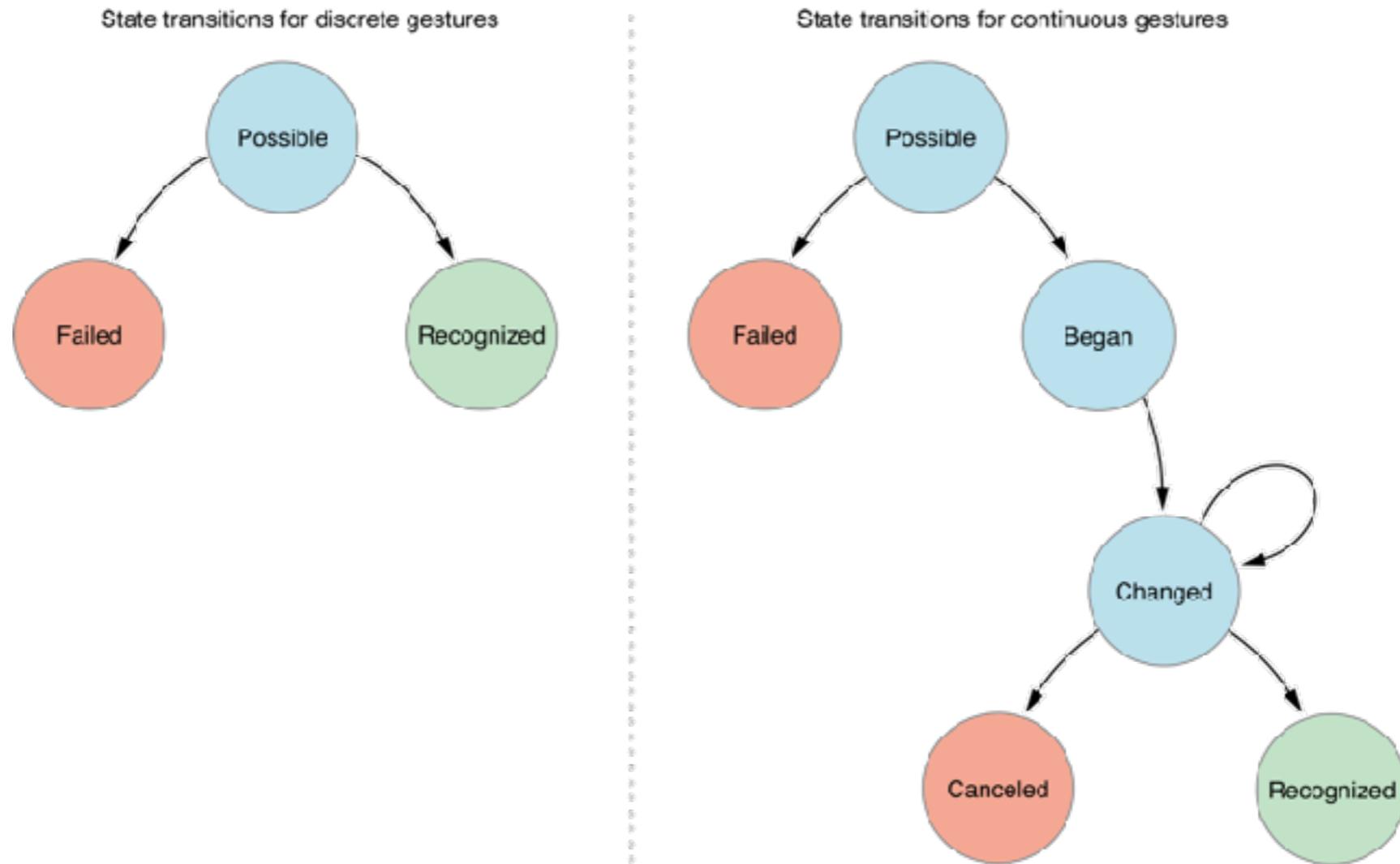
Es existieren **diskrete** und **kontinuierliche** Gesten



# Gesture Recognizer: Zustandsmodell

Gesture Recognizer operieren in einer Zustandsmaschine

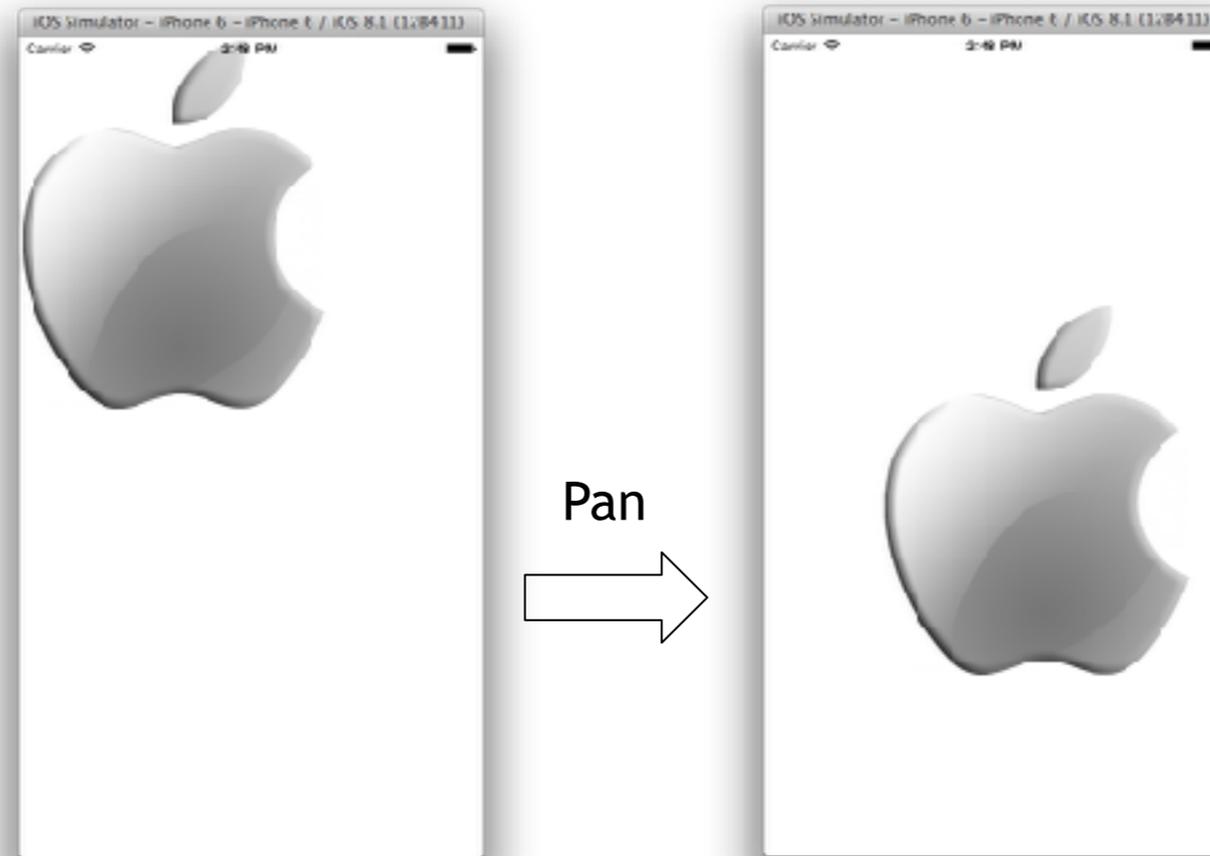
- mehr dazu im *"Event Handling Guide for iOS"*



Testen des gegenwärtigen Zustands über die Konstanten

- `UIGestureRecognizerStatePossible` (Default State)
  - Es wurden eventuell Touch-Events registriert, aber noch keine Geste erkannt
- `UIGestureRecognizerStateBegan`
  - Es wurde ein Touch-Event empfangen und eine **kontinuierliche** Geste erkannt
    - Löst eine Action-Nachricht aus
- `UIGestureRecognizerStateChanged`
  - Es wurden Touch-Events erkannt, die eine Veränderung in der **kontinuierlichen** Geste signalisieren
    - Löst eine Action-Nachricht aus
- `UIGestureRecognizerStateEnded`
  - Es wurden Touch-Events erkannt, die das Ende einer (**kontinuierlichen**) Geste signalisieren
    - Löst eine Action-Nachricht aus
    - Reset auf `UIGestureRecognizerStatePossible`

# Beispiel: ImageTranslationRecognition



# Erkennen paralleler Gesten: DragAndDrop

