

Gesture recognition

Fabrice.Kordon@lip6.fr



As an introduction...



Gestures

- Complex to handle directly with UIResponder



A dedicated class, UIGestureRecognizer

- Since iOS 3.2
- Handles «standard gesture»
 - ▶ Tap
 - ▶ Long press
 - ▶ Pinch
 - ▶ Rotation
 - ▶ Swipe
 - ▶ Pan
 - ▶ Screen edge
- Usable to invent new gestures too
 - ▶ New one or... composition of existing ones



RTFM!

Principles

As usual

- Creating a `UIGestureRecognizer`

- Main methods to support

 - `func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent)`

 - `func touchesMoved(_ touches: Set<UITouch>, with event: UIEvent)`

 - `func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent)`

 - `func touchesCancelled(_ touches: Set<UITouch>, with event: UIEvent)`

- Property state

 - ▶ possible, began, changed, ended, cancelled, failed

- About the delegate (`UIGestureRecognizerDelegate`)

 - ▶ Fine management for activation / deactivation

 - ▶ Fine management to deter and combine gestures



RTFM!

This is only a a basis for subclasses

- Numerous ones dedicated to predefined gestures

 - ▶ No need to work for this ones

UITapGestureRecognizer

Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UITapGestureRecognizer)
```

- ▶ Inserted in the list of recognisers

Configuration

- numberOfTapsRequired

- ▶ Default is 1

- numberOfTouchesRequired

- ▶ Default is 1

UILongPressGestureRecognizer

Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UILongPressGestureRecognizer)
```

- ▶ Inserted in the list of recognisers

Configuration

- numberOfTapsRequired, numberOfTouchesRequired

- ▶ Default is 1

- minimumPressDuration

- ▶ TimeInterval (seconds), default is 0.5

- allowableMovement

- ▶ CGFloat (points), default is 10

UIPinchGestureRecognizer



Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UIPinchGestureRecognizer)
```

- ▶ Inserted in the list of recognisers



Configuration

- scale

- ▶ Change since last call

- velocity

- ▶ Point/seconds

UIRotationGestureRecognizer

7



Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UIRotationGestureRecognizer)
```

- ▶ Inserted in the list of recognisers



Configuration

- rotation

- ▶ Absolute value (CGFloat)

- velocity

- ▶ Radians/seconds

UISwipeGestureRecognizer



Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UISwipeGestureRecognizer)
```

- ▶ Inserted in the list of recognisers



Configuration

- direction

- ▶ right, left, up, down

- numberOfTouchesRequired

- ▶ Default value is 1

UIPanGestureRecognizer

Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UISwipeGestureRecognizer)
```

- ▶ Inserted in the list of recognisers

Configuration

- `minimumNumberOfTouches` , `maximumNumberOfTouches`

- ▶ Respective default values is 1 and `NSUIntegerMax`

Extra functions

```
func translation(in view: UIView?) -> CGPoint
```

```
func setTranslation(_ translation: CGPoint, in view: UIView?)
```

```
func velocity(in view: UIView?) -> CGPoint
```


UIScreenEdgePanGestureRecognizer

10



Creation

- Association to a target + a method

```
init(target: Any?, action: Selector?)
```

- Associated method's prototype

```
func myMethod (sender : UIScreenEdgePanGestureRecognizer)
```

- ▶ Inserted in the list of recognisers



Configuration

- edges

- ▶ top, left, bottom, right, all

- Extra functions

```
func translation(in view: UIView?) -> CGPoint
```

```
func setTranslation(_ translation: CGPoint, in view: UIView?)
```

```
func velocity(in view: UIView?) -> CGPoint
```


As a conclusion...



A large catalog of gestures available



Useful when you used standard gestures



Be aware...

- ⦿ Might be more complex if you use them all
 - ▶ Conflicts possible
- ⦿ But so much simpler than directly using UIResponder...
 - ▶ Like shake motion

