






«iGesture»

Fabrice.Kordon@lip6.fr



Goal of the example

Play with specialised gesture recognisers

-  Double tap
-  Long press
-  Pinch
-  Rotation
-  Swipe

Sometime, apply on some visible items

-  Just for the fun of it



Demo



ViewController

```
//  
// ViewController.swift  
// iGestures  
//  
// Created by Fabrice Kordon on 03/11/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    private let v = MyView(frame: UIScreen.main.bounds)  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
        self.view = v  
    }  
  
    override func viewWillTransition(to size: CGSize,  
                                     with coordinator: UIViewControllerTransitionCoordinator) {  
        v.displayInSize(size: size)  
    }  
  
    override var preferredStatusBarStyle: UIStatusBarStyle {  
        return .lightContent  
    }  
}
```

MyView

```
import UIKit

class MyView: UIView {

    private let deepSpace = UIImageView(image: UIImage(named: "deep-space"))
    private let planet = UIImageView(image: UIImage(named: "mars"))
    private let l = UILabel()
    var prevAngle : CGFloat = 0.0

    override init(frame: CGRect) {
        super.init(frame: frame)
        self.backgroundColor = UIColor.black
        self.addSubview(deepSpace)

        planet.isHidden = true
        self.addSubview(planet)

        l.text = "No recognized gesture"
        l.textColor = UIColor.white
        l.textAlignment = .center
        self.addSubview(l)
    }
}
```

MyView

```
let tapDetect = UITapGestureRecognizer(target: self,
                                     action: #selector(tapDetection(sender:)))
tapDetect.numberOfTapsRequired = 2
self.addGestureRecognizer(tapDetect)
let pressDetect = UILongPressGestureRecognizer(target: self,
                                               action: #selector(pressDetection(sender:)))
pressDetect.minimumPressDuration = 1.0
self.addGestureRecognizer(pressDetect)
let pinchDetect = UIPinchGestureRecognizer(target: self,
                                           action: #selector(pinchDetection(sender:)))
self.addGestureRecognizer(pinchDetect)
let rotDetect = UIRotationGestureRecognizer(target: self,
                                           action: #selector(rotDetection(sender:)))
self.addGestureRecognizer(rotDetect)
//let panDetect = UIPanGestureRecognizer(target: self,
//                                       action: #selector(panDetection(sender:)))
//self.addGestureRecognizer(panDetect) // Swipe et Pan not compatibles!!!
let swipeLeft = UISwipeGestureRecognizer(target: self,
                                       action: #selector(swipeDetection(sender:)))
swipeLeft.direction = .left
let swipeRight = UISwipeGestureRecognizer(target: self,
                                       action: #selector(swipeDetection(sender:)))
swipeRight.direction = .right
let swipeDown = UISwipeGestureRecognizer(target: self,
                                       action: #selector(swipeDetection(sender:)))
swipeDown.direction = .down
let swipeUp = UISwipeGestureRecognizer(target: self,
                                       action: #selector(swipeDetection(sender:)))
swipeUp.direction = .up
self.addGestureRecognizer(swipeLeft)
self.addGestureRecognizer(swipeRight)
self.addGestureRecognizer(swipeDown)
self.addGestureRecognizer(swipeUp)
self.displayInSize(size: UIScreen.main.bounds.size)
}
```

MyView

```
required init(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}

func displayInSize(size: CGSize) {
    deepSpace.center = CGPoint(x: size.width / 2.0, y: size.height / 2.0)
    l.frame = CGRect(x: 10.0, y: 50.0, width: size.width - 20.0, height: 20.0)
    planet.center = CGPoint(x: size.width / 2.0, y: size.height / 2.0)
}

// Double tap
@objc func tapDetection (sender : UITapGestureRecognizer) {
    if sender.state == .ended {
        if planet.isHidden {
            planet.isHidden = false
        } else {
            planet.isHidden = true
        }
    }
}

// Long-press
@objc func pressDetection (sender : UILongPressGestureRecognizer) {
    l.text = "«long press» detected"
    let s = deepSpace.image?.size
    deepSpace.frame = CGRect(x: 0.0, y: 0.0, width: s!.width,
                             height: s!.height)
    planet.transform = planet.transform.scaledBy(x: 1.0, y: 1.0)
    self.displayInSize(size: UIScreen.main.bounds.size)
}
```

MyView

```
// Pinch
@objc func pinchDetection (sender : UIPinchGestureRecognizer) {
    l.text = String (format: "«pinch» of %d%% detected",
                       Int(sender.scale * 100.0))
    if !planet.isHidden {
        planet.transform = planet.transform.scaledBy(x: sender.scale,
                                                    y: sender.scale)
    } else {
        let s = deepSpace.frame.size
        deepSpace.frame.size = CGSize(width: s.width * sender.scale,
                                       height: s.height * sender.scale)
    }
}

// Rotation
@objc func rotDetection (sender : UIRotationGestureRecognizer) {
    if sender.state == .ended {
        prevAngle = 0.0
    } else if sender.state == .changed {
        let rotation = 0 - (prevAngle - sender.rotation)
        let degrees = sender.rotation / CGFloat.pi * 180.0
        l.text = String (format: "«rotation» of %.0f° detected", degrees)
        if !planet.isHidden {
            planet.transform = planet.transform.rotated(by: rotation)
        }
        prevAngle = sender.rotation
    }
}
```


MyView

```
// Pan (deactivated because conflict with swipe)
@objc func panDetection (sender : UIPanGestureRecognizer) {
    if sender.state == .changed {
        l.text = "«pan» detected"
        if !planet.isHidden {
            let t = sender.translation(in: planet)
            planet.center = CGPoint(x: planet.center.x + t.x / 2.0,
                                    y: planet.center.y + t.y / 2.0)
            sender.setTranslation(CGPoint(x:0, y:0), in: self)
        }
    }
}


// Swipe
@objc func swipeDetection (sender : UISwipeGestureRecognizer) {
    if sender.direction == .right {
        l.text = "«right swipe» detected"
    }
    if sender.direction == .left {
        l.text = "«left swipe» detected"
    }
    if sender.direction == .up {
        l.text = "«swipe to top» detected"
    }
    if sender.direction == .down {
        l.text = "«swipe to bottom» detected"
    }
}
}
```

As a conclusion

 **Once again, it's easy!**



 **Huge value added to you Apps...**

 **You MUST use it!**