

# «Shaker»

Fabrice.Kordon@lip6.fr



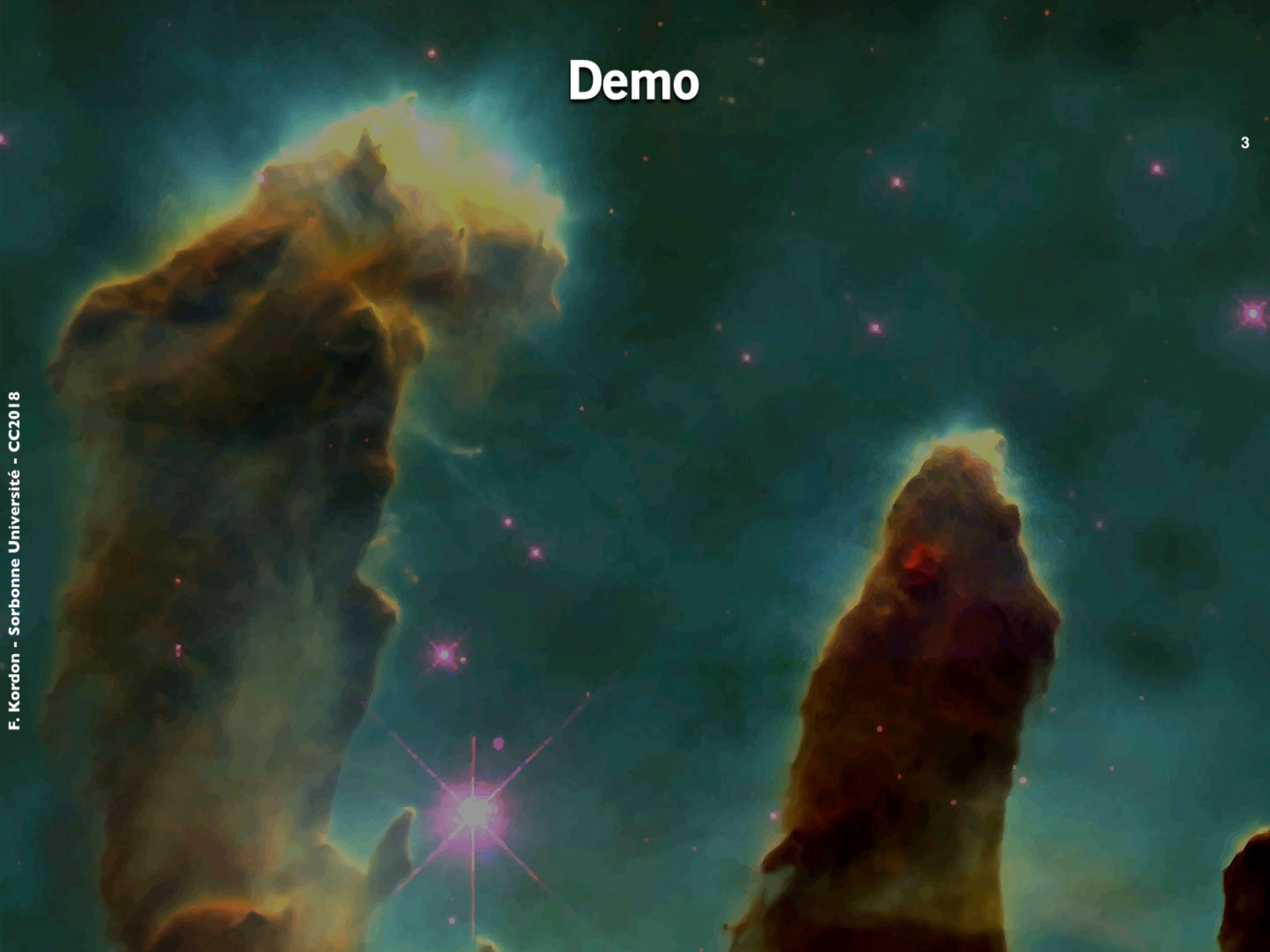
# Goal of the example

-  **Play with UIResponder**
  -  Understand how to handle events

 **Here, count shakes**



# Demo



# ViewController



**Sake of simplicity...**

Code located in a  
ViewController

# ViewController

```
import UIKit

class ViewController: UIViewController {

    private let v = UIView()
    private let shaked = UIImageView(image: UIImage(named: "vodka-martini"))
    private let backg = UIImageView(image: UIImage(named: "backg"))
    private let l1 = UILabel()
    private let l2 = UILabel()
    private var count = 0

    override var preferredStatusBarStyle : UIStatusBarStyle {
        return .lightContent
    }
}
```

# ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    let bounds = UIScreen.main.bounds
    v.frame = bounds
    self.view = v
    backg.frame = bounds
    v.addSubview(backg)
    backg.center = CGPoint(x: bounds.width / 2, y: bounds.height / 2)
    l1.textAlignment = .center
    l1.font = UIFont(name: "Arial", size: 20)
    l1.textColor = .white
    l1.text = "No shake detected"
    l1.frame = CGRect(x: 20, y: 70,
                     width: bounds.width - 40, height: 40)

    v.addSubview(l1)
    l2.textAlignment = .center
    l2.font = UIFont(name: "Arial", size: 20)
    l2.textColor = .white
    l2.text = "___"
    l2.frame = CGRect(x: 20, y: bounds.height - 80,
                     width: bounds.width - 40, height: 40)

    v.addSubview(l2)
    shaked.center = CGPoint(x: bounds.width / 2, y: bounds.height / 2)
    shaked.isHidden = true
    v.addSubview(shaked)
    // To catch events
    self.becomeFirstResponder()
}
```

# ViewController

```
// UIResponder

override func motionBegan(_ motion: UIEvent.EventSubtype,
                          with event: UIEvent?) {
    l2.text = "motionBegan"
    shaken.isHidden = false
}

override func motionEnded(_ motion: UIEvent.EventSubtype,
                          with event: UIEvent?) {
    l2.text = "motionEnded"
    count += 1
    if count == 1 {
        l1.text = "1 shake detected"
    } else {
        l1.text = String(format: "%d shakes detected", count)
    }
    shaken.isHidden = true
}

override func motionCancelled(_ motion: UIEvent.EventSubtype,
                              with event: UIEvent?) {
    l2.text = "motionCancelled"
    shaken.isHidden = true
}
}
```

# As a conclusion...



## Nice is'n't it?

- And useful too
- Remote command works similarly
  - ▶ You need to check for the type of event



## About «gestures»?

- Handled by UIGestureRecognizer
  - ▶ To be studied later

