

Various elements about iOS development

Fabrice.Kordon@lip6.fr



What elements?

-  Random in iOS
-  Widget orientations
-  Reasons why Apps are rejected



Get random values in iOS

Swift

Recent (and nice) changes with Swift 4.2

▶ Integrated in the language facilities

```
let rnd1 = Int.random(in: 0 ..< 10)
let rnd2 = Double.random(in: 0.0 ... 10.0)
let rnd3 = Bool.random()
```

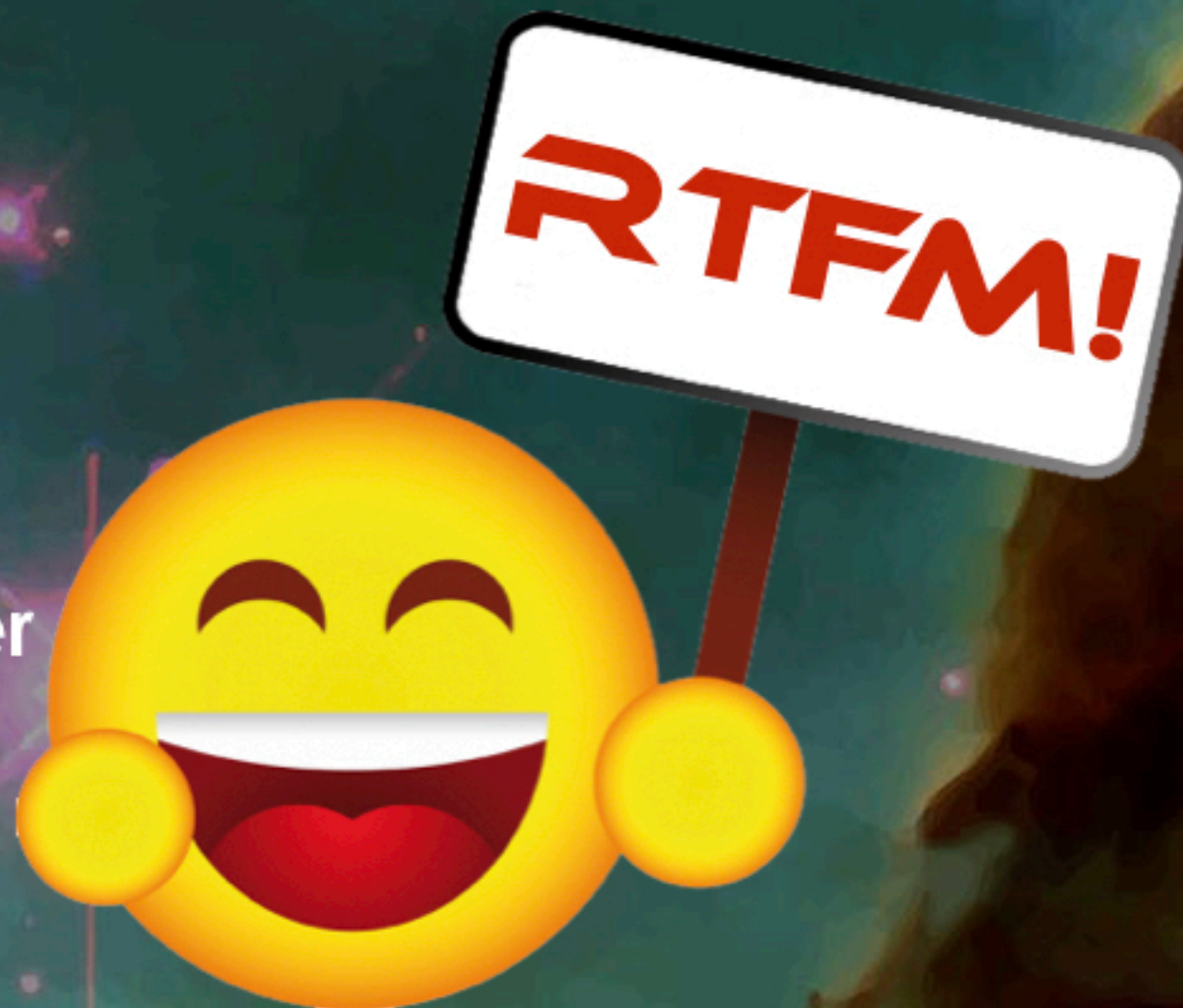
```
let tab = ["Paris", "Roma", "Berlin", "Brussel", "Madrid"]
let rnd4 = tab.randomElement()
```

Also some variants here you control the seed

Objective-C

A C function...

- ▶ `arc4random()`
- ▶ returns a random integer



Change of widgets orientation

```
import UIKit

class ViewController: UIViewController {
    let l = UILabel()
    let s = UISlider()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        self.view = UIView()
        self.view.backgroundColor = UIColor.white
        l.frame = CGRect(x:UIScreen.main.bounds.width / 2.0 - 50.0,
                        y:80.0, width:100.0, height: 30.0)
        l.textAlignment = .center
        l.text = "0%"
        self.view.addSubview(l)
        s.frame = CGRect(x: 0.0, y: 0.0, width: 300, height: 10.0)
        s.center = CGPoint(x:UIScreen.main.bounds.width / 2.0,
                          y:UIScreen.main.bounds.height / 2.0)
        s.transform = CGAffineTransform(rotationAngle: -3.14 / 2.0)
        s.addTarget(self, action: #selector(sliderAction), for: .valueChanged)
        self.view.addSubview(s)
    }

    @objc func sliderAction (slider : UISlider) {
        let val = Int(slider.value * 100)
        l.text = "\(val)%"
    }
}
```

Change of widgets orientation



```
ViewController {
```

```
  viewDidLoad() {
```

setup after loading the view, typically from a nib.

```
    let slider = UISlider(frame: CGRect(x: 0.0, y: 0.0, width: 300, height: 10.0),
```

```
                              value: 0.77, trackTintColor: UIColor.white,
```

```
                              minimumValue: 0.0, maximumValue: 1.0)
```

```
    slider.addTarget(self, action: #selector(sliderAction), for: .valueChanged)
```

```
  }
```

```
  @IBAction func sliderAction(_ sender: UISlider) {
```

```
    let value = sender.value
```

```
    let percentage = value * 100
```

```
    let label = UILabel(frame: CGRect(x: 0.0, y: 0.0, width: 300, height: 10.0),
```

```
                              text: String(format: "%.1f%%", percentage),
```

```
                              font: UIFont.systemFont(ofSize: 16))
```

```
    label.textAlignment = .center
```

```
    label.frame.origin.y = slider.frame.origin.y - 10
```

```
    label.frame.origin.x = slider.frame.origin.x
```

```
    label.frame.origin.z = slider.frame.origin.z
```

```
    label.frame.origin.w = slider.frame.origin.w
```

```
    label.frame.origin.h = slider.frame.origin.h
```

```
    label.frame.origin.t = slider.frame.origin.t
```

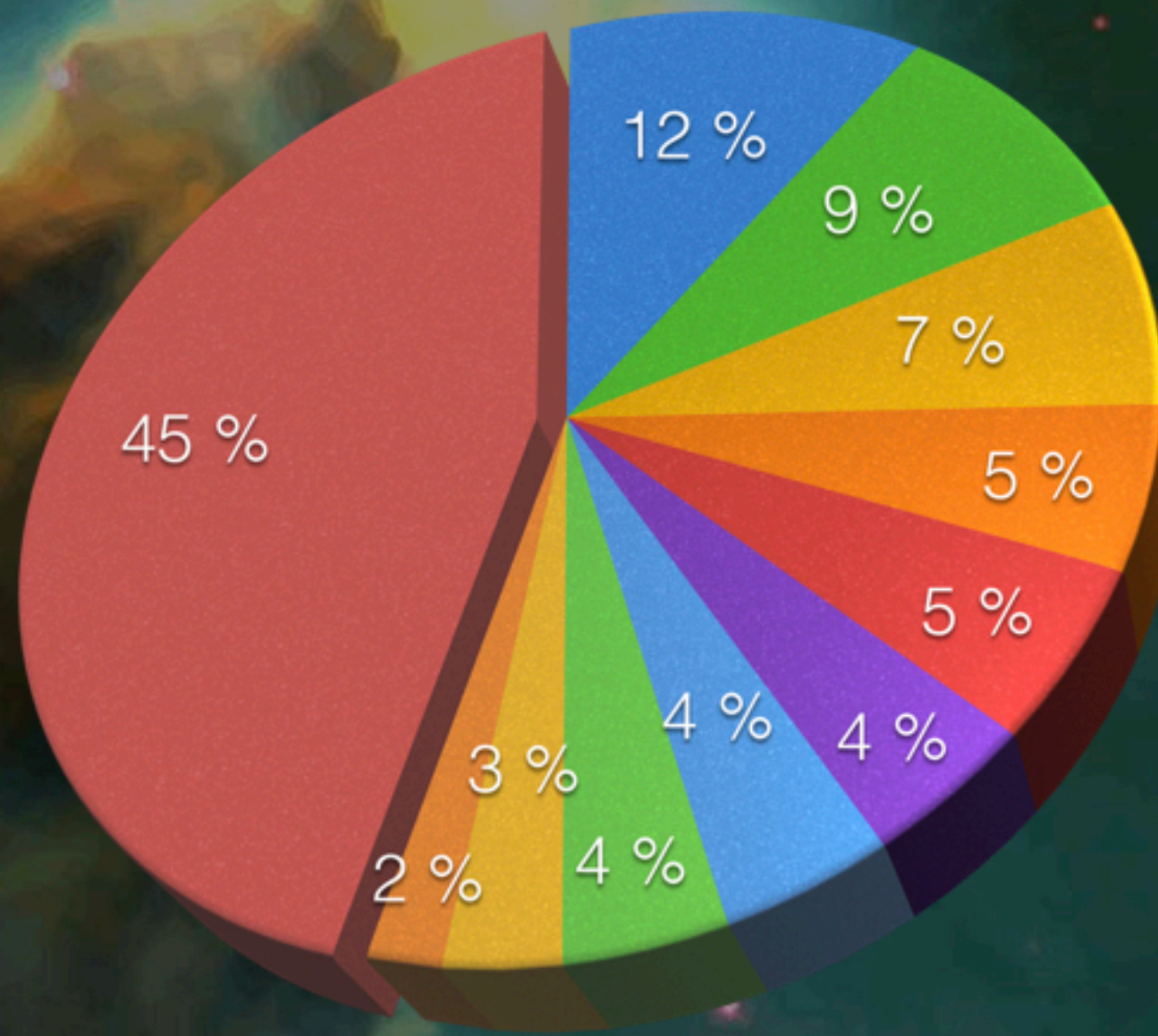
```
  }
```

```
}
```

Reported rejections reasons



2014



- Missing data
- Bugs found
- Privacy problems
- Problems with developer licence
- Interfaces too complex
- Crashes
- Mismatch between description and the App
- Naming troubles between iTunesConnect and the device
- Names or icons to close to Apple's ones
- Submission tests
- Other

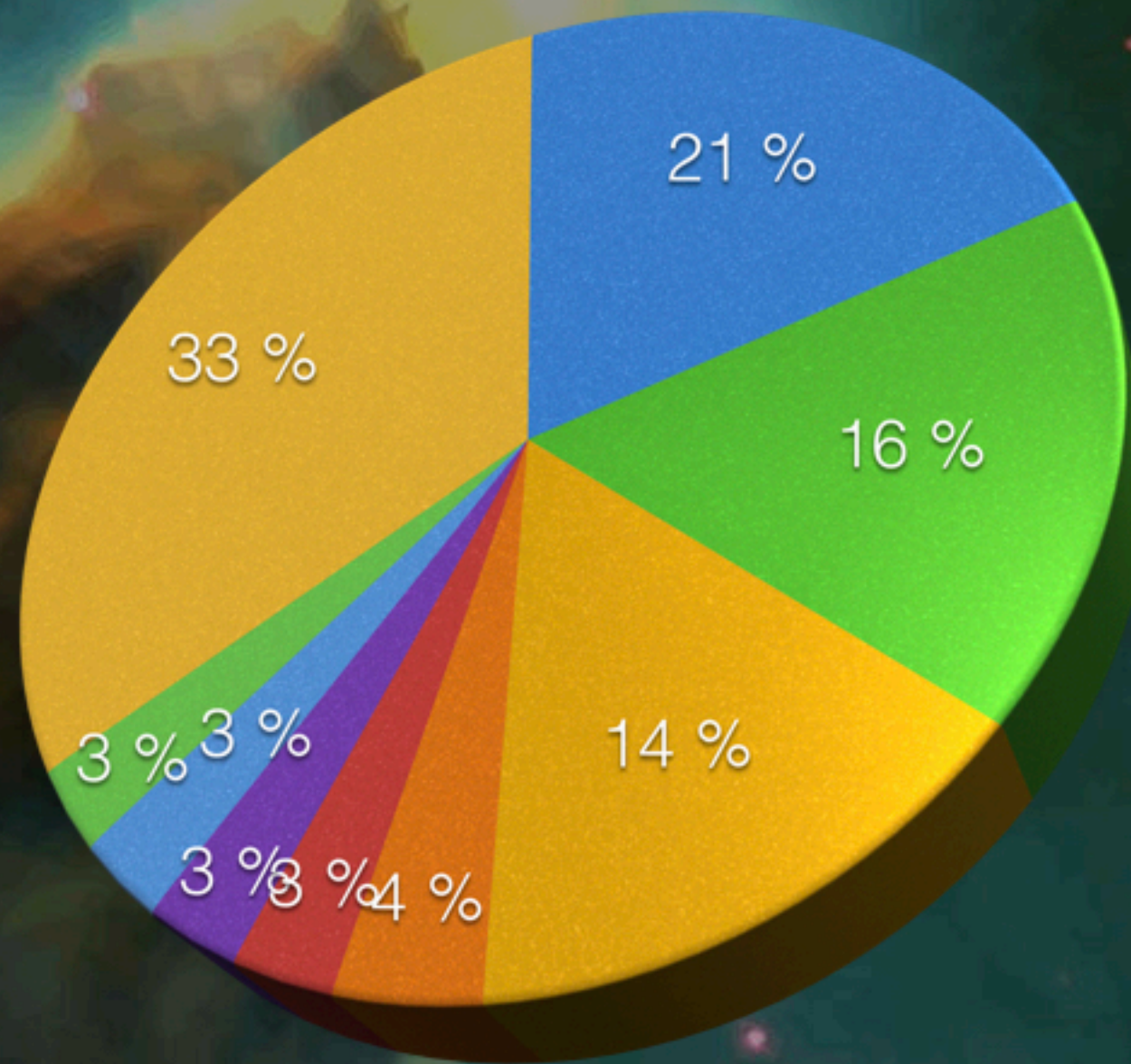
Reported rejections reasons



2014



2016



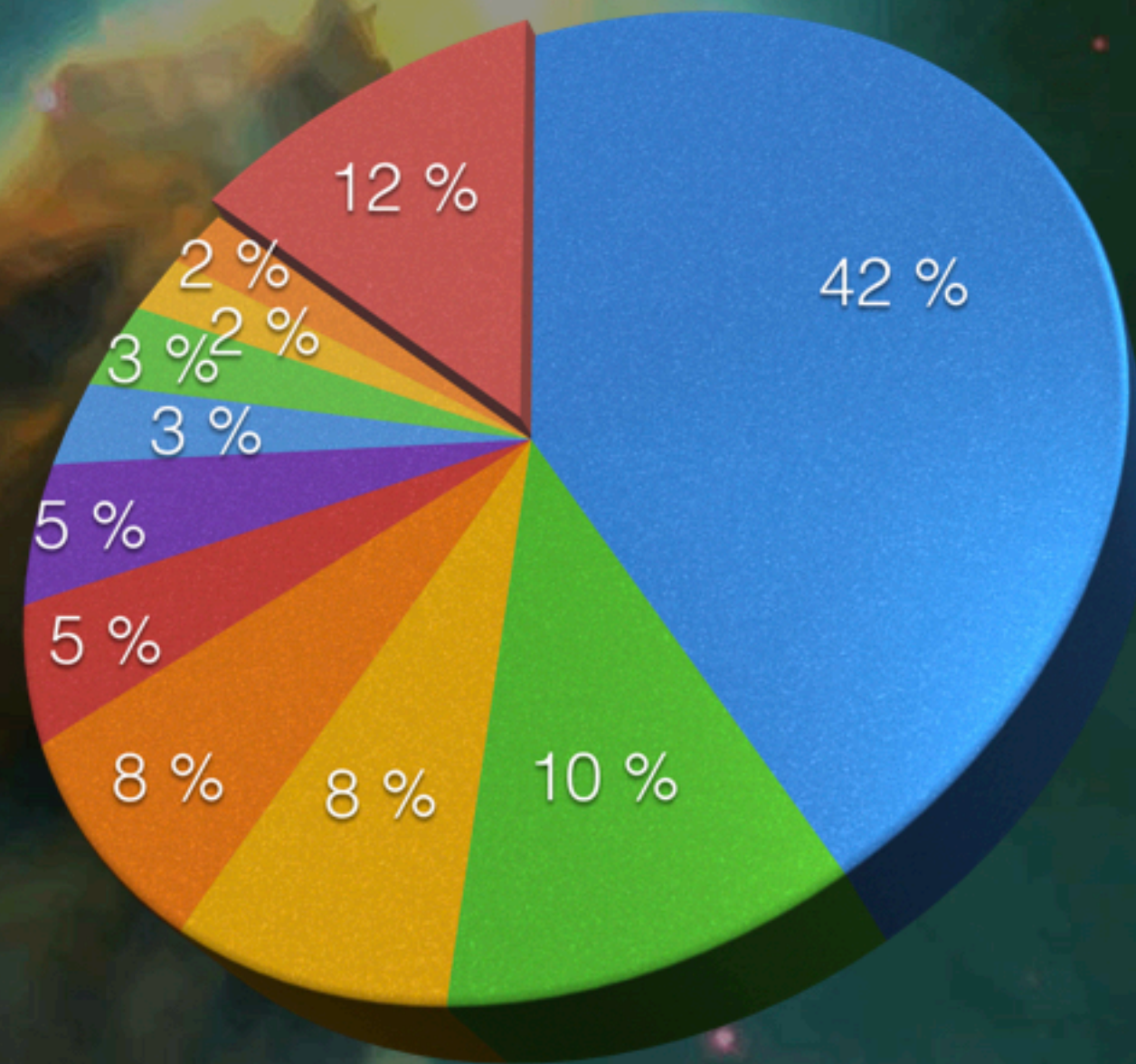
- Incomplete App
- Missing data
- Metadata
- Plagiarism
- Soft. requirements
- In-App purchase
- Design
- Privacy problems
- Other












Reported rejections reasons

 2014

 2016

 2018



-  Incomplete App
-  Design spam
-  Privacy Problems
-  Performances
-  Location services
-  In-App purchase
-  Metadata
-  Minimum fonctionnality
-  Business model issues
-  Performance
-  Other

As a conclusion...



Some advices

- Nice applications you will make
 - ▶ **User interface aspects are important**
- Apple's guidelines you will respect
 - ▶ **They stand on it**
- Quality you will achieve
 - ▶ **No bug, please consider code readability too**
- The developer licence you will read
 - ▶ **Detect what is forbidden (see rejections reasons)**



Remember, Swift is still moving

- But slower than a few years ago

