

«Praxinoscope»

Fabrice.Kordon@lip6.fr

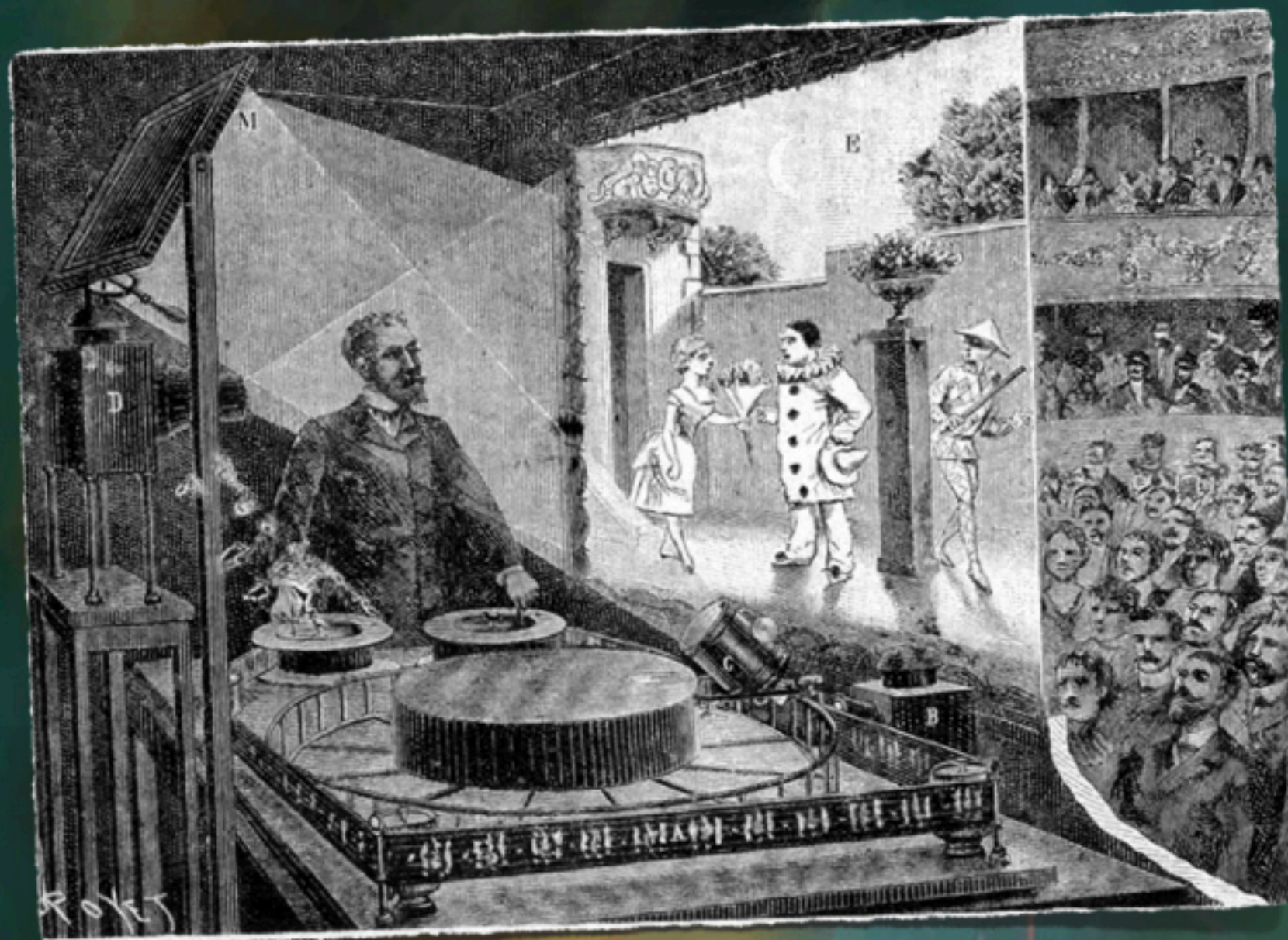


As an introduction...



Invented by Émile Reynaud (1876)

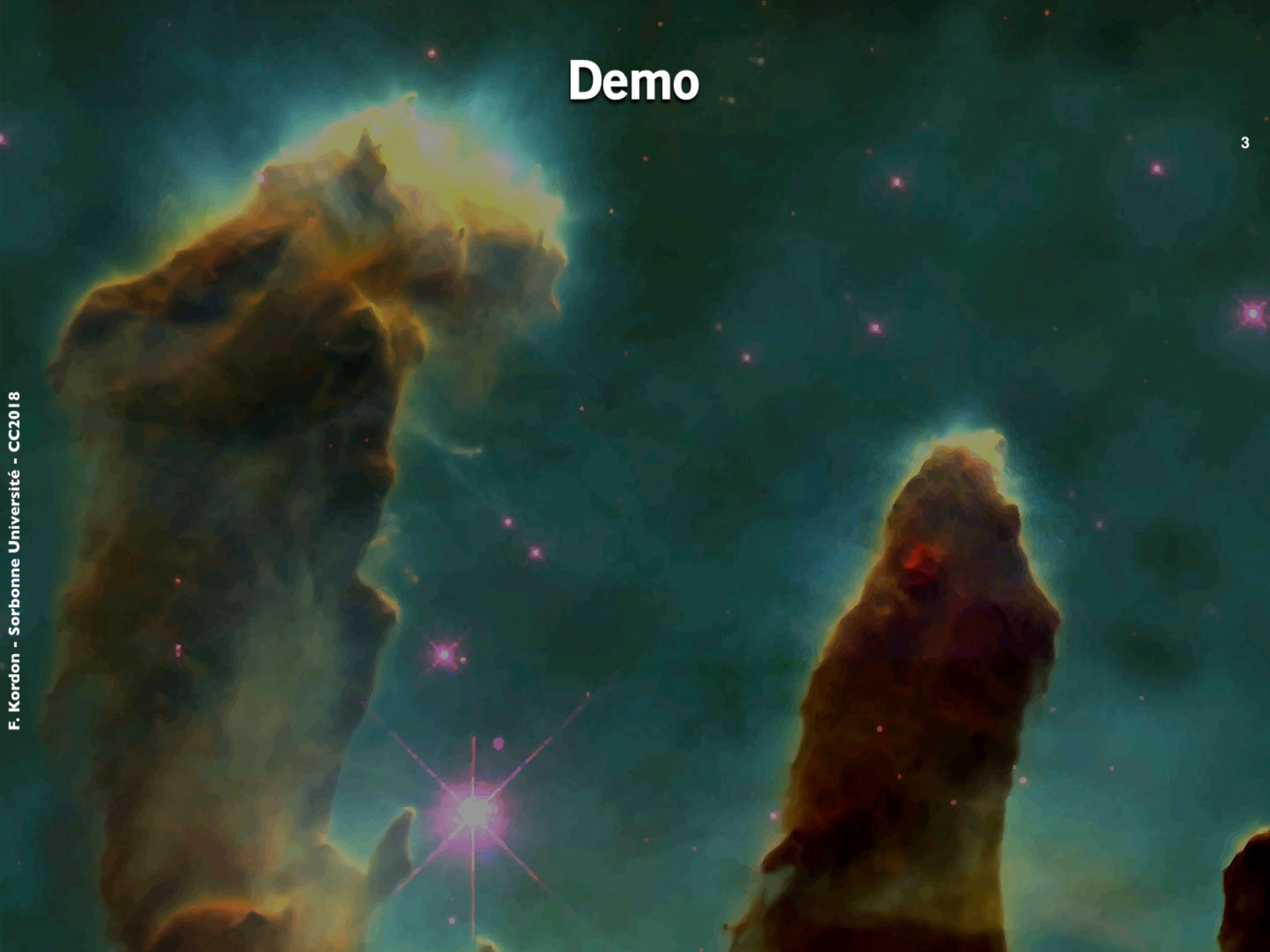
- Extension of the Zoetrope
- Creates an illusion of movement
- Presentation in «optical theaters»
- Huge success at the 1878 universal exposition in Paris
- Disappears with Cinema...



Goal

- Rotate a wheel
- Start/Stops
- Several speeds

Demo



ViewController



Sake of simplicity...

Code located in a
ViewController

ViewController

```
import UIKit

class ViewController: UIViewController {

    private let wheel = UIImageView(image: UIImage(named: "wheel"))
    private let backg = UIImageView(image: UIImage(named: "backimg"))
    private let b = UIButton(type: .system)
    private var moving = false
    private let f = UISegmentedControl(items: ["10Hz", "50Hz", "100Hz"])
    private weak var t : Timer? // remind that here, default is "nil"
    private var angle = CGFloat(0)
```

ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    backg.center = CGPoint(x: UIScreen.main.bounds.size.width / 2.0,
                           y: UIScreen.main.bounds.size.height / 2.0)

    backg.alpha = 0.5
    b.setTitle("Start", for: .normal)
    b.frame = CGRect(x: UIScreen.main.bounds.size.width / 2.0 - 50,
                    y: 50, width: 100, height: 30)
    wheel.center = CGPoint(x: UIScreen.main.bounds.size.width / 2.0,
                           y: UIScreen.main.bounds.size.height / 2.0)

    f.frame = CGRect(x: UIScreen.main.bounds.size.width / 2.0 - 120,
                    y: 100, width: 240, height: 30)
    f.selectedSegmentIndex = 0
    self.view = UIView()
    self.view.backgroundColor = UIColor.white
    b.addTarget(self, action: #selector(action), for: .touchDown)
    f.addTarget(self, action: #selector(actionSelector),
               for: .valueChanged)

    self.view.addSubview(backg)
    self.view.addSubview(b)
    self.view.addSubview(f)
    self.view.addSubview(wheel)
}
```

ViewController

```
@objc func action() {
    if moving {
        b.setTitle("Start", for: .normal)
        moving = false
        t?.invalidate()
        t = nil
    } else {
        b.setTitle("Stop", for: .normal)
        moving = true
        self.resetTimer()
    }
}

@objc func actionSelector() {
    if moving {
        self.resetTimer()
    }
}
```

ViewController

```
func resetTimer() {
    var interval = TimeInterval(0.0)
    switch f.selectedSegmentIndex {
        case 0: interval = 0.1
        case 1: interval = 0.02
        case 2: interval = 0.01
        default: () // NOP
    }
    t?.invalidate() // Remind t is optional
    t = nil
    t = Timer.scheduledTimer(timeInterval: interval,
                             target: self,
                             selector: #selector(rotate),
                             userInfo: nil,
                             repeats: true)
}


func resetTimer2() { // Alternative version with an instruction block
    ...
    t = Timer.scheduledTimer(withTimeInterval: interval,
                             repeats: true,
                             block: { (tm: Timer) -> Void in
                                     self.rotate(t: tm)
                                 })
}
```


ViewController

```
@objc func rotate(t : Timer) {  
    angle = angle + 0.157  
    if angle > 6.28 {  
        angle = 0.0  
    }  
    wheel.transform = CGAffineTransform(rotationAngle: angle)  
}  
}
```

As a conclusion...

Simple and useful!

 For more informations...



Shall we suppress the timer when our App enters in background?

 Not clear

 Nice to know

- ▶ Timers are stopped in background
- ▶ They restart when the App enters foreground