调色板

Apollonian

预 习

LiulietLee

UP教你做个画图程序

av5293327

主要内容

- Swift 3 (语言) 和 Xcode 8 (IDE)
- 简化，单例模式
- 整理，computed property – {get set}

Demo

Swift 2 -> Swift 3

2 -> 3: drawRect

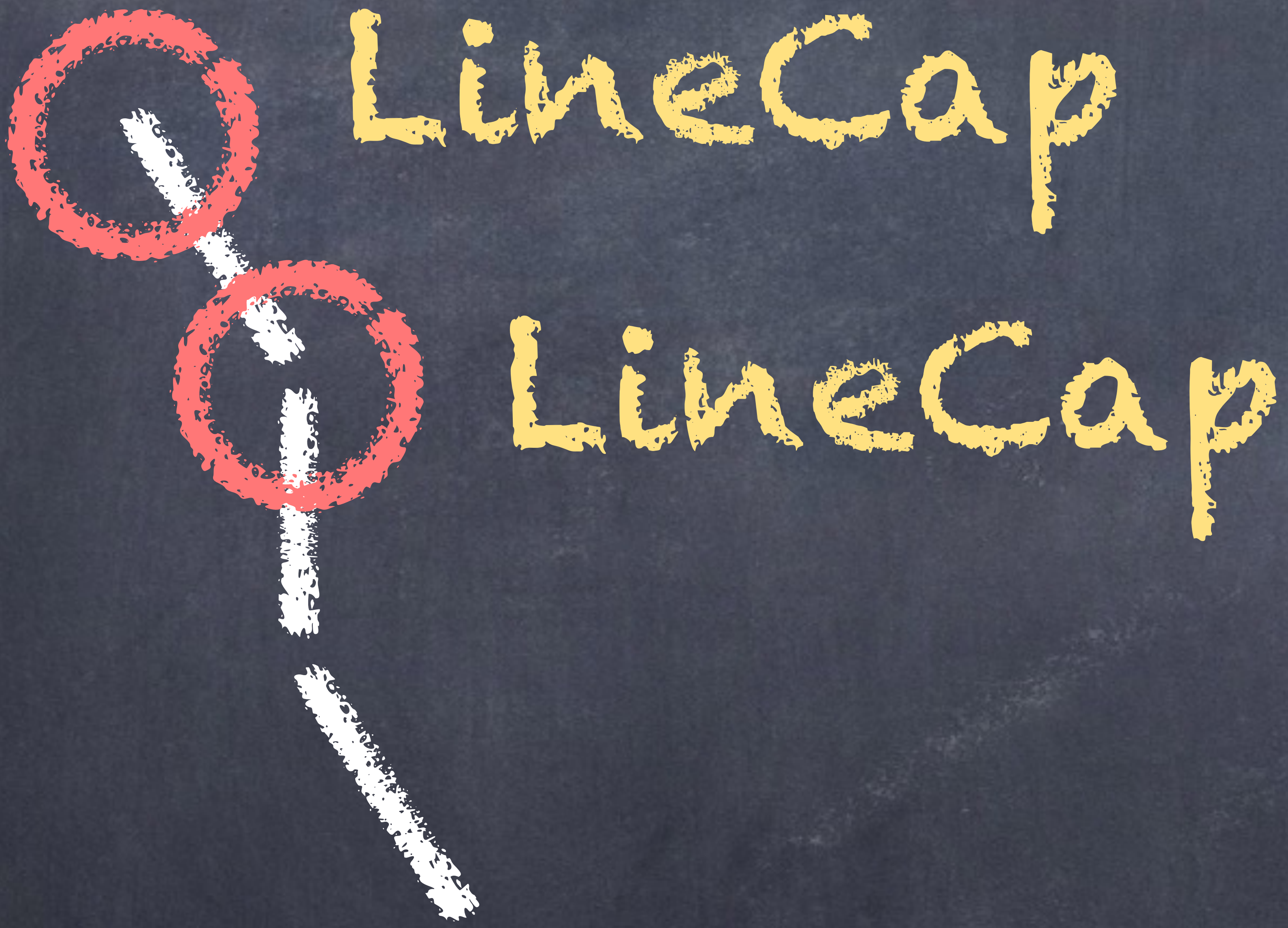
```
let context = UIGraphicsGetCurrentContext()
CGContextSetLineCap(context, .Round)
CGContextSetLineWidth(context, 5)
CGContextSetStrokeColorWithColor(context, color.CGColor)
CGContextMoveToPoint(context, x, y)
CGContextAddLineToPoint(context, x, y)
CGContextStrokePath(context)
```

```
let context = UIGraphicsGetCurrentContext()
context?.setLineCap(.round)
context?.setLineWidth(5)
context?.setStrokeColor(color.cgColor)
context?.moveTo(x: x, y: y)
context?.addLineTo(x: x, y: y)
context?.strokePath()
```

简化

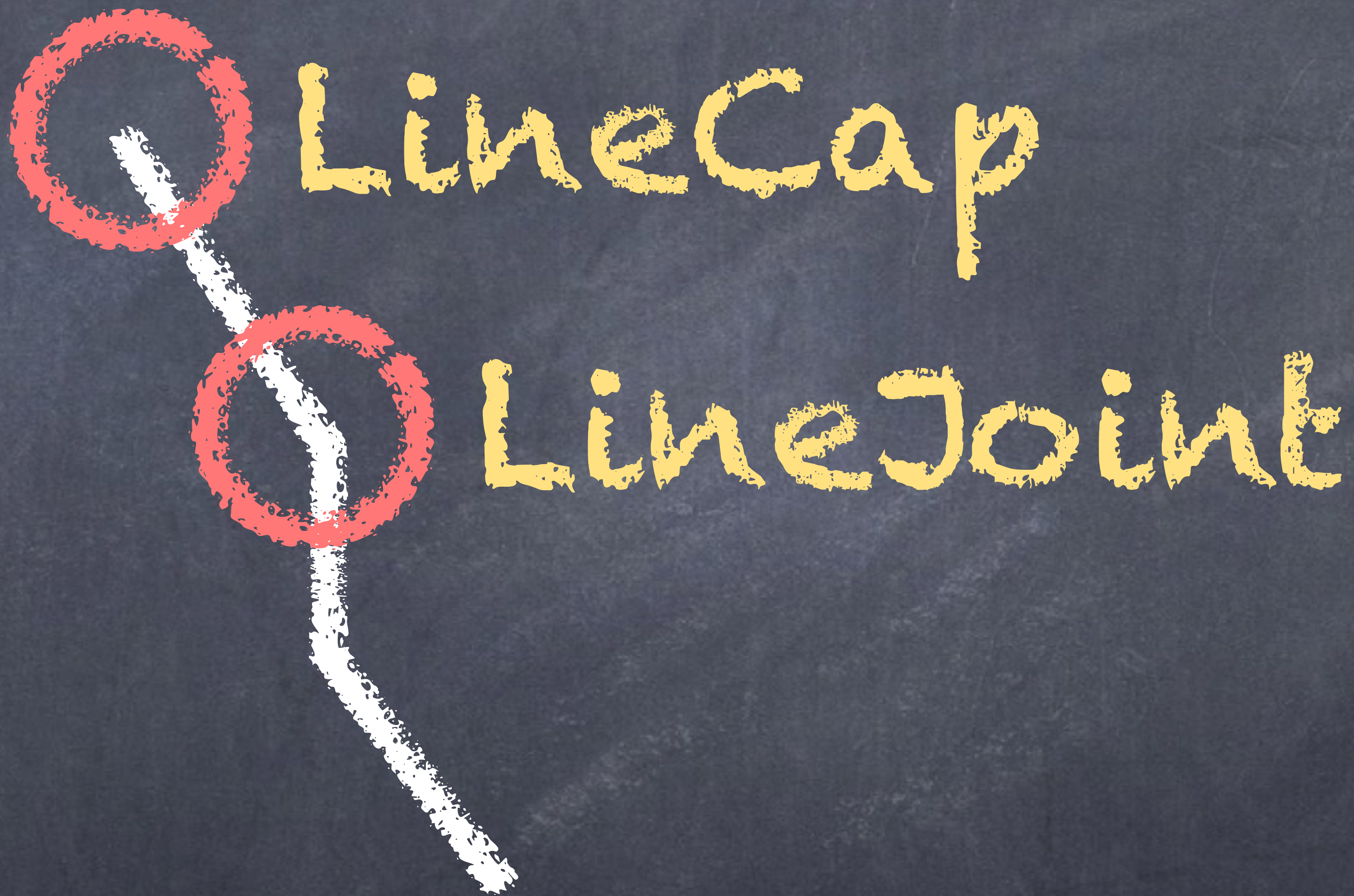
Line 和 UIBezierPath

Line 和 UIBezierPath



LineCap

LineCap

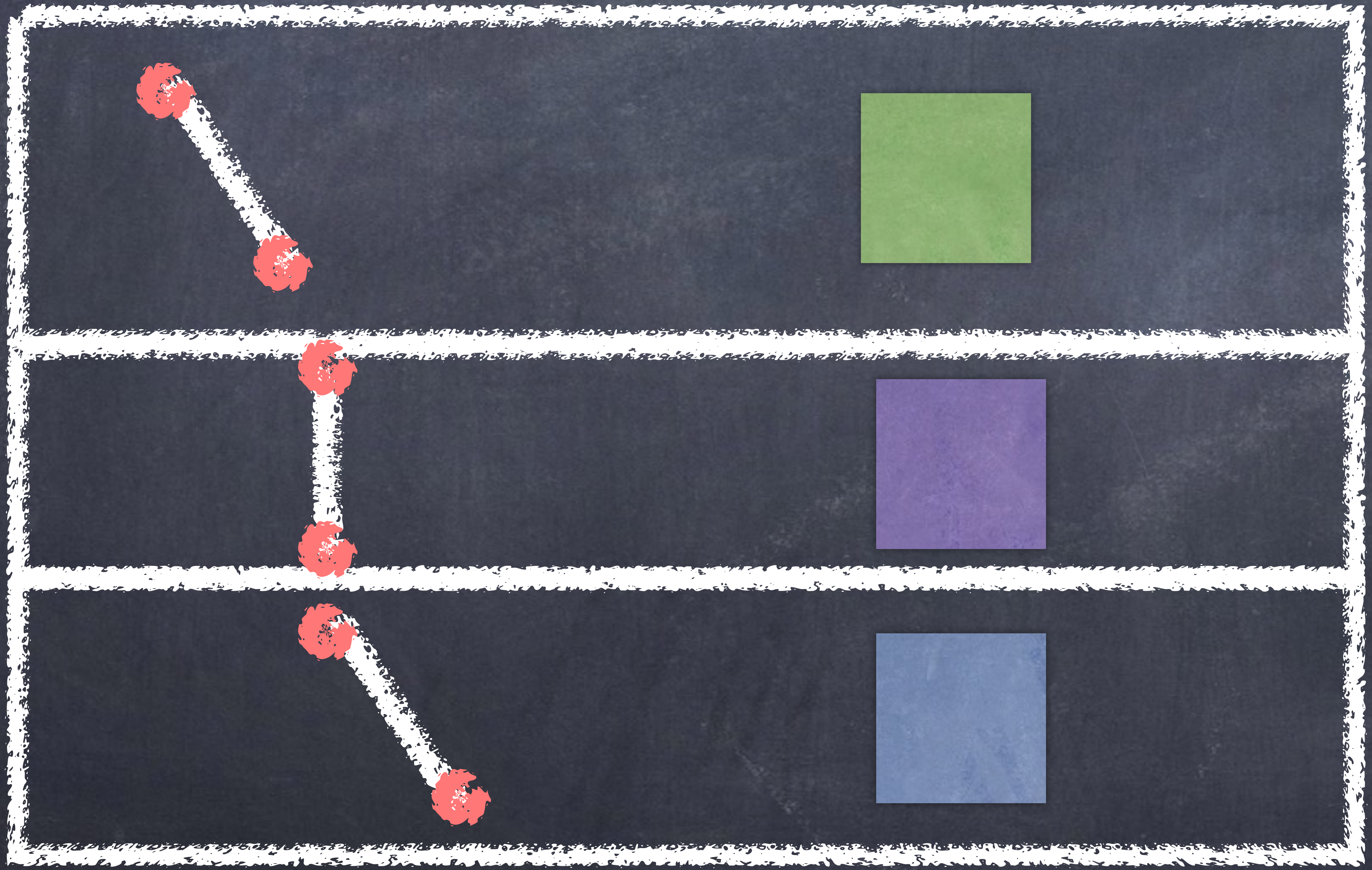


LineCap

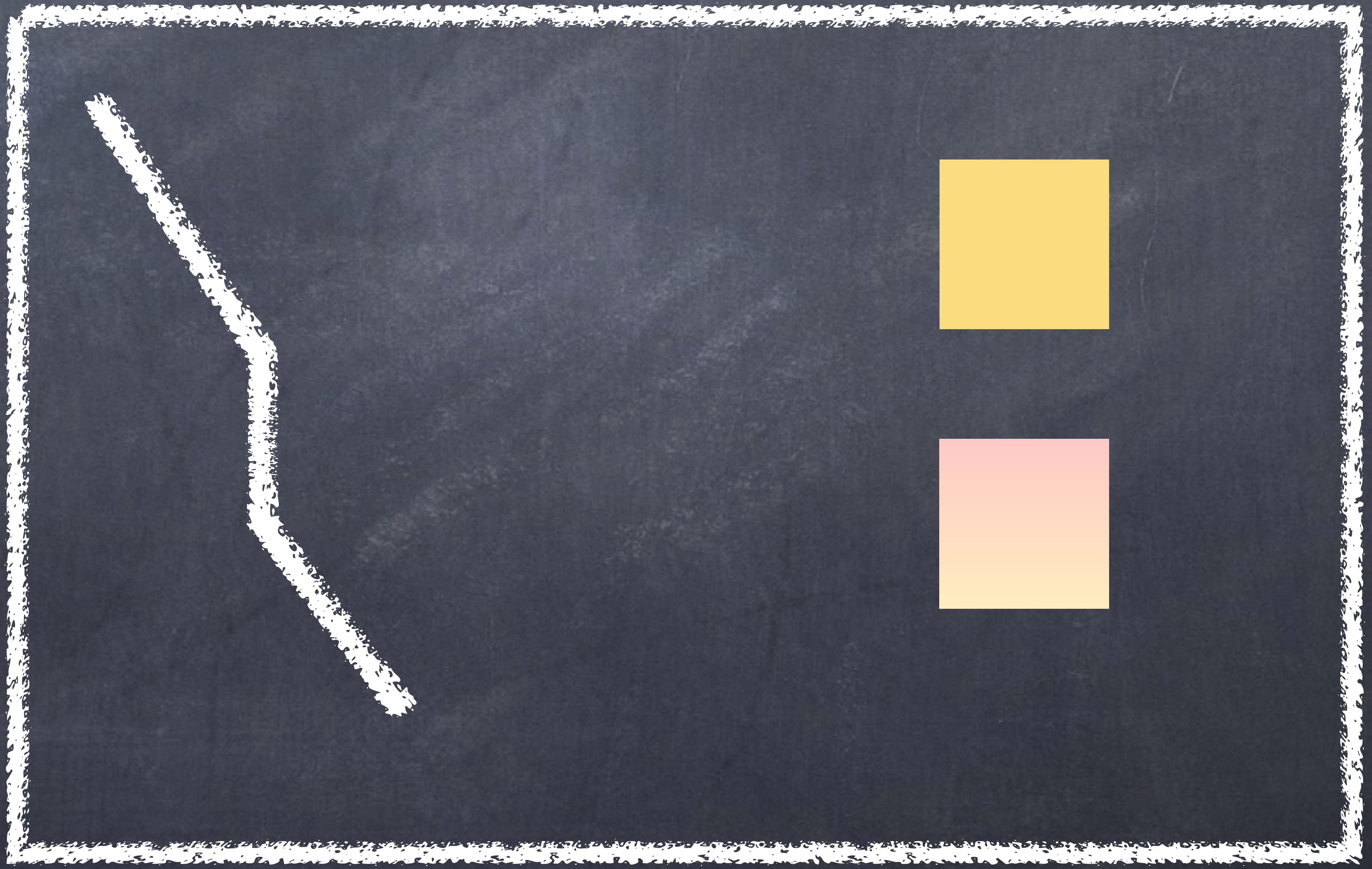
LineJoint

Line 和 UIBezierPath

- class Line(start, end, color)



- typealias Line = (path, color)



Line 和 UIBezierPath

- class Line(start, end, color)
- lines =

[

  - [{CGPoint, CGPoint, UIColor}, ...]
  - ...

]

- typealias Line = (path, color)
- lines =

[

  - (UIBezierPath, UIColor)
  - ...

]

Line 和 UIBezierPath

- lines.append([Line]())
- lastPoint = location

- lines.append((UIBeizierPath(), lineColor))
- lines.last?.path.move(to: location)

Line 和 UIBezierPath

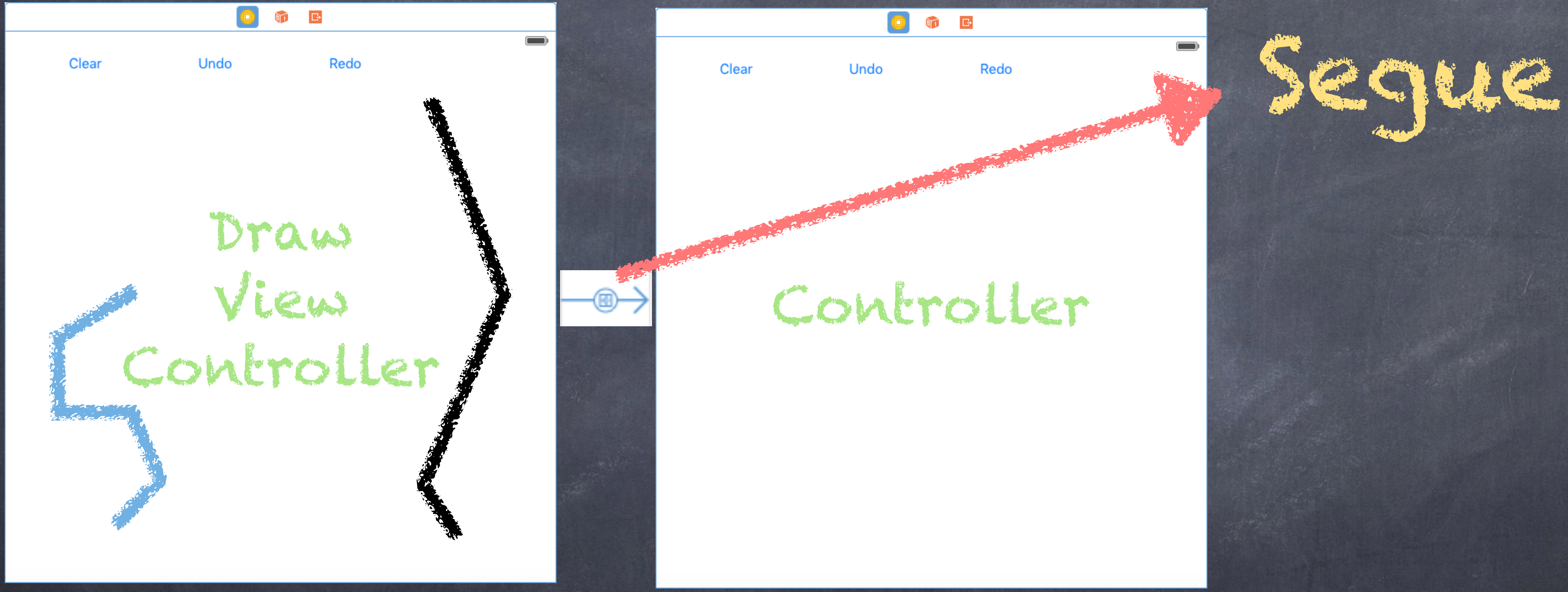
- newPoint = location
- lines[lines.count - 1].append(Line(lastPoint, newPoint, lineColor))
- lastPoint = newPoint

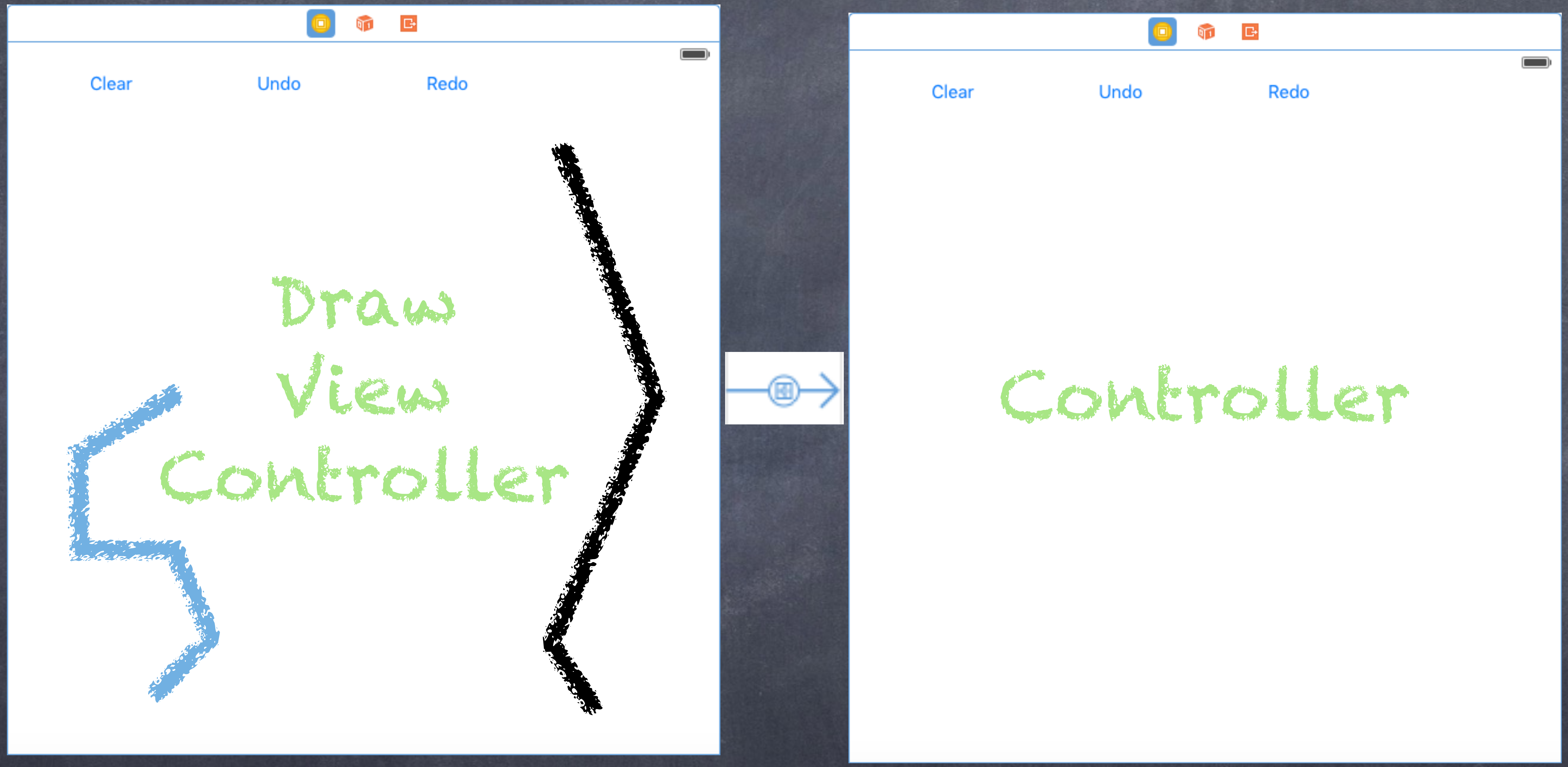
- lines.last?.path.addLine(to: location)

Demo

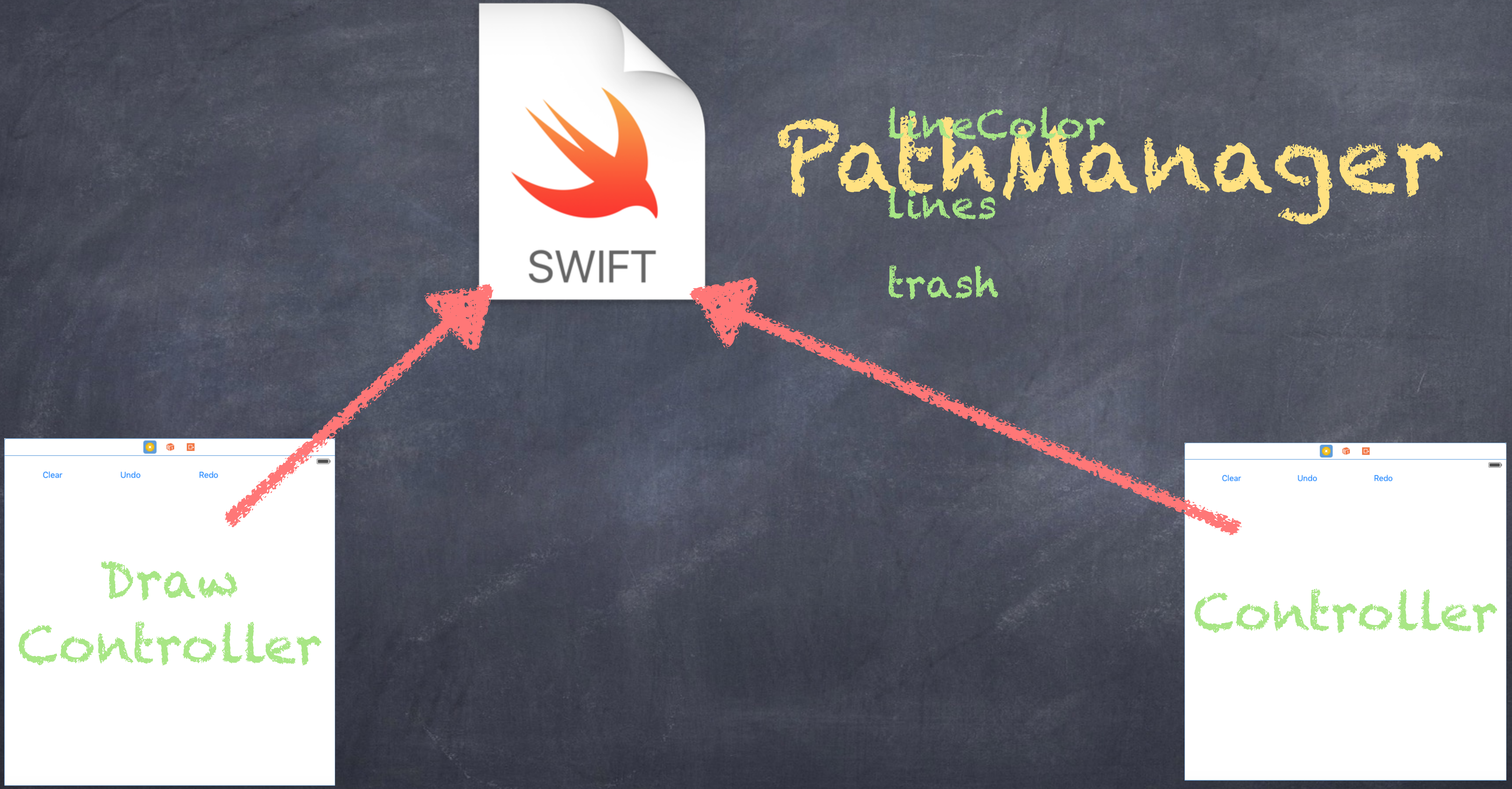
Line -> UIBezierPath

Segue


Clear
Undo
Redo
Draw View Controller
Clear
Undo
Redo
Controller
Segue



Clear
Undo
Redo
Draw
View
Controller
Clear
Undo
Redo
Controller



PathManager

lineColor

lines

trash

Draw Controller

Controller

PathManager

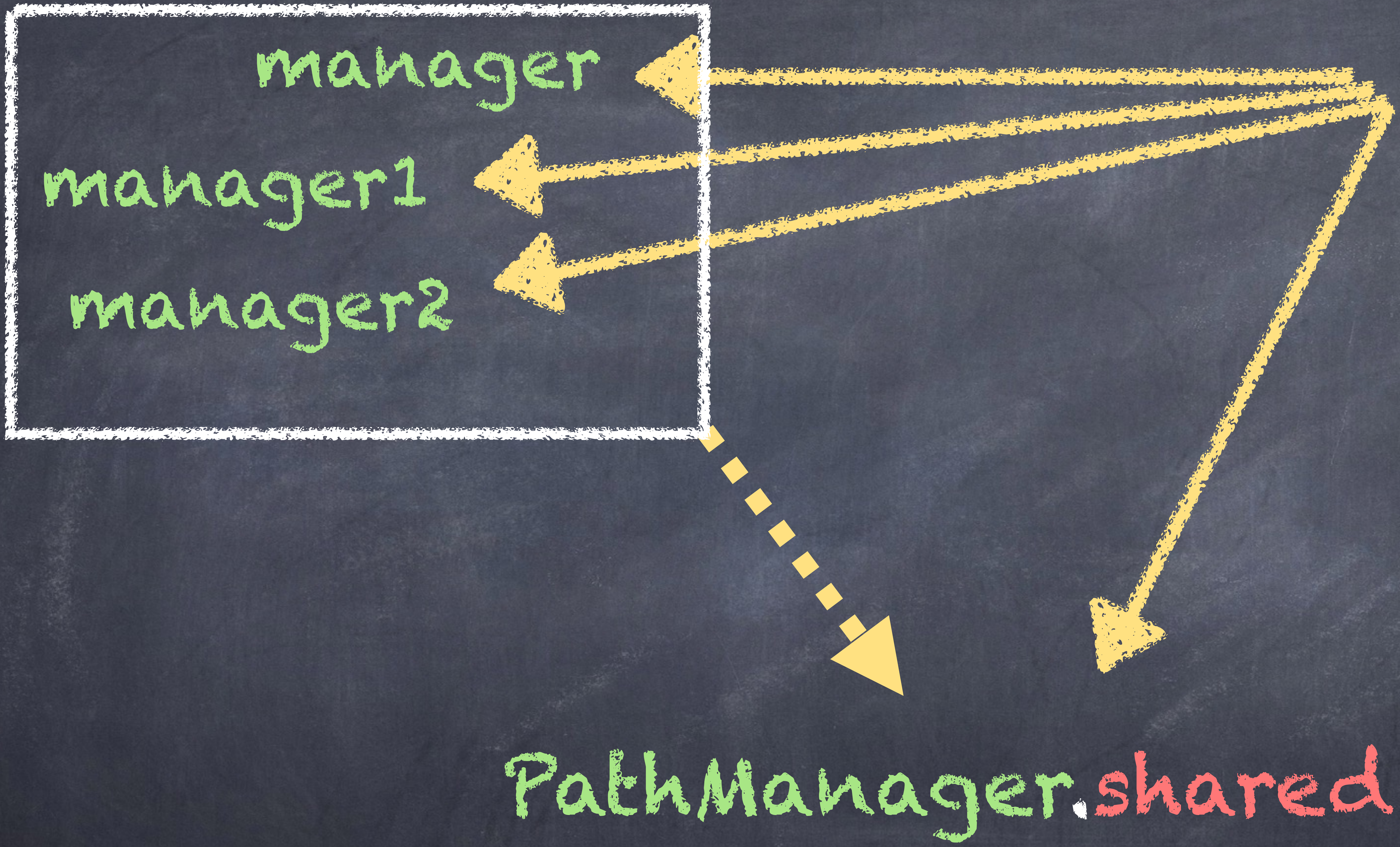
保证仅有一个实例

单例模式

PathManager -> Singleton

```
class PathManager{
    static let shared = PathManager()
    private init(){}
}
```

```
let manager = PathManager.shared
```



manager

manager1

manager2

PathManager.shared

Demo

PathManager

Computed Property

Computed Property

```
var name: Type{
 get{ /*当获取值时*/}
 set{ /*当赋新值(newValue)时*/}
}
```

Computed Property

```
willGet{ /*当即将获取值前*/}

willSet{ /*当即将赋新值(newValue)前*/}

didGet{ /*当已经获取值后*/}

didSet{ /*当已经赋新值后(oldValue)*/}
```

UILabel $ UISlider

UIKit

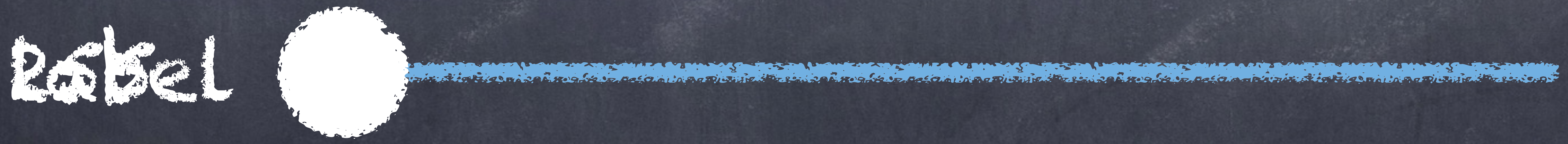
```
label.text = String

slider.setValue(Float, animated: Bool)
```

UIKit

```
label.text = "255"

slider.setValue(1, animated: true)
```



欢迎关注，收藏，丢硬币

github.com/ApolloZhu