

p1: 改进

Swift 2 -> 3

可以点击 Convert

如果点击了 Later, 可以通过 Edit -> Convert -> Current Swift Syntax 来转化

PPT

Line -> UIBezierPath

```
1 // 我们定义 Line, 更新数组, 删除 lastPoint
2 typealias Line = (path: UIBezierPath, color: UIColor)
3
4 var lines = [Line]()
5 var trash = [Line]()
6 // var lastPoint
```

drawRect

```
1 // lineCap, Width 等是 UIBezierPath 的属性
2 // 我们通过循环设置, 绘制
3 for line in lines{
4     line.color.setStroke()
5     line.path.lineJoinStyle = .round
6     line.path.lineCapStyle = .round
7     line.path.lineWidth = 5
8     line.path.stroke()
9 }
```

touches 系列简化

```
1 // 当用户开始触摸, , 将起始点移动到当前触摸位置
2 func touchesBegan(_:_:){
3     super.touchesBegan(_:_:){
4         // 我们可以不判断 touches, 而是触摸的位置是否为空
5         // 如果能够使得 location 为
6         // touches 中第一个在 view 里的位置
7         // 而不是nil
8         if let location = touches.first?.location(in: self){
```

```

9      // 那么新建 UIBezierPath 并添加到 lines
10     lines.append((UIBezierPath(), lineColor))
11     //
12     lines.last?.move(to: location)
13 }
14 }

1 // 将更新线位置变为一个统一的函数
2 private func update(for touches: Set<UITouch>){
3     if let location = touches.first?.location(in: self){
4         // 延长Path到当前触摸位置
5         lines.last?.addLine(to: location)
6         // 更新界面
7         setNeedsDisplay()
8     }
9 }

1 // 快捷键 touches
2 // 所以我们在 touchesMoved, ended, cancelled 执行 update(for: touches) 函数
3 func touchesXXX(_:_){
4     super.touchesXXX(_:_)
5     update(for: touches)
6 }

```

运行测试

p2: 调色板

Segue

PPT

添加返回按钮，讲解如何新建 Segue

Manager

```

1 // 新建 PathManager
2 import UIKit
3 typealias Line = (path: UIBezierPath, color: UIColor)
4 let manager = PathManager.shared
5 class PathManager{
6     static let shared = PathManager()
7     private init(){}
8     var lines = [Line]()
9     var trash = [Line]()
10    var lineColor = UIColor.black()
11    func clear(){
12        lines.removeAll()

```

```

13         trash.removeAll()
14     }
15     func undo(){
16         if !lines.isEmpty{
17             trash.append(lines.removeLast())
18         }
19     }
20     func redo(){
21         if !trash.isEmpty{
22             lines.append(trash.removeLast())
23         }
24     }
25
26     // 为了方便使用, 我们添加两个
27     func newLine(){
28         lines.append((UIBezierPath(), lineColor))
29     }
30     // 为了能够时刻更新, 我们使用 Computed Property
31     // > PPT
32     var current: (line: Line?, path: UIBezierPath?, lineColor: UIColor){
33         get{
34             // 当我们在获取的时候, 返回数组中当前的最后一个元素
35             return (lines.last, lines.last?.path, lineColor)
36         }
37     }
38 }

```

更改 DrawView

```

1 for line in manager.lines {}
2
3 touchesBegan(_:_:){
4     manager.newLine()
5     manager.current.path?.move(to: location)
6 }
7
8 update(_:){ manager.current.path?.addLine(to: location) }

```

RGBAViewController

新建 RGBAViewController, 删除原本代码

链接 — [rgba]Label, [rgba]Slider, preview

加快摆放控件

```

// Label: [Text:Black, Shadow:(1,1)] White;width=height=1

<Back preview
r -0- 255
g -0- 255
b -0- 255
a -0- 255

```

private RGBA

```

1 private var r: CGFloat = 0{
2     willSet{
3         rLabel.text = "\(newValue * 255)"
4         rSlider.setValue(Float(newValue), animated: true)
5     }
6 }
7 // 补全: defGBA

```

value

```

1 var value: UIColor{
2     get{
3         return UIColor(red: r, green: g, blue: b, alpha: a)
4     }
5     set{
6         // UnsafeMutablePointer<CGFloat>
7         // getRed(_ red: inout CGFloat, ...)
8         newValue.getRed(&r, green: &g, blue: &b, alpha: &a)
9         preview.backgroundColor = newValue
10    }
11 }

```

共用 IBAction

设置几个 slider 的 tag 属性: 0, 1, 2, 3

```

1 // 注意, 各个 Slider 都要关联到
2 @IBAction func valueChanged(_ sender: UIButton){
3     switch sender.tag{
4         case 0: r = CGFloat(sender.value)
5         // 补全: casesGBA
6     }
7     preview.backgroundColor = value
8 }

```

viewWillXXX -> Controller

```

1 // RGBA
2 func viewWillDisappear(_:){
3     super.viewWillDisappear(_:)
4     manager.lineColor = value
5 }
6
7 func viewWillAppear(_:){
8     super.viewWillAppear(_:)
9     preview.backgroundColor = manager.lineColor
10 }
11

```

```
12 // Draw, 更新 StoryBoard 链接
13 func viewWillAppear(_:){
14     super.viewWillAppear(_)
15     value = manager.lineColor
16 }
```

ViewController

```
1 func touchUpInside(sender: UIButton){
2     switch sender.tag{
3         case 1:
4             manager.clear()
5             // 补全: tui
6     }
7     drawView.setNeedsDisplay()
8 }
```

运行，会报错，然后删除原来的关联
尝试旋转，更改为 redraw