CAGradientLayer

Apple,bilibili,swift

基础

今天的内容是线性渐变图层的实现
要用到的是 UIKit 中的 CAGradientLayer

```
import UIKit
class HueGradientLayer: CAGradientLayer { }
```

为了能够在 Storyboard 中实时看到效果，加上

```
@IBDesignable
```

修改 CAGradientLayer 的属性来更改渐变的颜色和方向

```
func setup() {
    var controls = [CGFloat]()
    controls = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]
    locations = controls as [NSNumber]?
    colors = []
    for stop in controls {
        colors!.append(
            UIColor(hue: stop, saturation: 1, brightness: 1, alpha: 1).cgColor
        )
    }
    startPoint = CGPoint(x: 0, y: 0.5)
    endPoint = CGPoint(x: 1, y: 0.5)
}
```

让这段代码在初始化的时候都执行

```
override init() {
    super.init()
    setup()
}
override init(layer: Any) {
    super.init(layer: layer)
    setup()
}
required init?(coder aDecoder: NSCoder) {
    super.init(coder: aDecoder)
    setup()
}
```

怎么使用呢?

```
class HueGradientView: UIView {
    override open class var layerClass: AnyClass {
```

```
        return HueGradientLayer.self
    }
}
```

```
let view = HueGradientView(CGRect(x:0,y:0,width:685,height:30))
```

优化

关键点的地方打起来太麻烦。如果我要更精确，每个 0.01 有一个关键点, 有没有什么简单的方法?

```
controls = Array(stride(from: 0, through: 1, by: 0.1))
```

直接把 controls 转换成 [CGColor]?

```
colors = controls.map({ (hue) -> CGColor in
    return UIColor(hue: hue, saturation: 1, brightness: 1, alpha: 1).cgColor
})
```

```
colors = controls.map { UIColor(hue: $0, saturation: 1, brightness: 1, alpha: 1).cgColor }
```