

PolygonView.swift

PolygonView.swift

File Recent HelloPoly > HelloPoly > PolygonView.swift > drawRect(rect:)

Project Navigator HelloPoly AppDelegate.swift PolygonVC.swift PolygonView.swift PolygonShape.swift Main.storyboard Assets.xcassets LaunchScreen.storyboard Info.plist

Products project.app

```
1 import UIKit
2
3 class PolygonView: UIView {
4     var delegate: PolygonProtocol?
5
6     override func draw(_ rect: CGRect) {}
7
8     func drawRect(rect: CGRect) {
9         let shape = CAShapeLayer()
10        layer.addSublayer(shape)
11        shape.opacity = 0.5
12        shape.name = "polygon"
13        shape.lineWidth = 2
14        shape.lineJoin = CAShapeLayerLineJoin.miter
15        shape.strokeColor = UIColor(hue: 0.786, saturation: 0.79, brightness: 0.53, alpha: 1.0).cgColor
16        shape.fillColor = UIColor(hue: 0.786, saturation: 0.15, brightness: 0.89, alpha: 1.0).cgColor
17
18        let path = UIBezierPath()
19
20        for (index, point) in delegate!.pointsInRect(rect: rect).enumerated() {
21            if index == 0 {
22                path.move(to: point)
23            } else {
24                path.addLine(to: point)
25            }
26        }
27
28        path.close()
29        shape.path = path.cgPath
30    }
31}
32}
33}
```

PolygonVC.swift

PolygonView

```
1 import UIKit
2
3 class PolygonVC: UIViewController {
4     // SCREEN LABELS
5     @IBOutlet var sidesLbl: UILabel!
6     @IBOutlet var decreaseBtn: UIButton!
7     @IBOutlet var increaseBtn: UIButton!
8
9     // POLYGON VIEW WITH CUSTOM CLASS TYPE
10    @IBOutlet var polygonView: PolygonView!
11
12    // POLYGON SHAPE INSTANCE
13    var polygon = PolygonShape()
14
15    // POLYGON FRAME FOR THE SCREEN.
16    var polygonFrame: CGRect?
17
18    // CONSTANTS AND KEY VALUES
19    var MIN_SIDES = 3
20    var MAX_SIDES = 12
21    var SIDES_KEY = "sides"
22
23    override func viewDidLoad() {
24        super.viewDidLoad()
25
26        polygonFrame = CGRect(x: polygonView.center.x / 2, y: polygonView.center.y / 3, width: 100, height: 100)
27        polygonView.delegate = polygon
28
29        let defaults = UserDefaults.standard
30        let sides = defaults.integer(forKey: SIDES_KEY)
31        if sides != 0 { polygon.numberOfSides = sides }
32
33        drawPolygon()
34        checkBtnState()
35
36        let swipeLeft = UISwipeGestureRecognizer(target: self, action: #selector(decreaseSide))
37        swipeLeft.direction = .left
38        view.addGestureRecognizer(swipeLeft)
39
40        let swipeRight = UISwipeGestureRecognizer(target: self, action: #selector(increaseSide))
41        swipeRight.direction = .right
42        view.addGestureRecognizer(swipeRight)
43    }
44
45    func drawPolygon() {
46        let defaults = UserDefaults.standard
47        defaults.set(polygon.numberOfSides, forKey: SIDES_KEY)
48        _ = polygonView.layer.sublayers?.popLast()
49        polygonView.drawRect(rect: polygonFrame!)
50        sidesLbl.text = String(polygon.numberOfSides)
51    }
52
53    @IBAction func decreaseSide(_: Any) {
54        if polygon.numberOfSides > MIN_SIDES {
55            polygon.numberOfSides -= 1
56        }
57        checkBtnState()
58        drawPolygon()
59    }
60
61    @IBAction func increaseSide(_: Any) {
62        if polygon.numberOfSides < MAX_SIDES {
63            polygon.numberOfSides += 1
64        }
65        checkBtnState()
66        drawPolygon()
67    }
68
69    func checkBtnState() {
70        if polygon.numberOfSides > MIN_SIDES { decreaseBtn.isEnabled = true
71        } else { decreaseBtn.isEnabled = false
72        }
73
74        if polygon.numberOfSides < MAX_SIDES { increaseBtn.isEnabled = true
75        } else { increaseBtn.isEnabled = false
76        }
77    }
78}
79}
80}
```

PolygonShape.swift

HelloPoly

HelloPoly

AppDelegate.swift

PolygonVC.swift

PolygonView.swift

PolygonShape.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

Products

project.app

```
1 import CoreGraphics
2 import Foundation
3
4 protocol PolygonProtocol {
5     func pointsInRect(rect: CGRect) -> [CGPoint]
6 }
7
8
9 class PolygonShape: PolygonProtocol {
10    public var numberOfSides: Int = 6
11    public var name: String = "name"
12
13    func pointsInRect(rect: CGRect) -> [CGPoint] {
14        let center = rect.center
15        let radius = min(rect.size.width, rect.size.height)
16        let arc = 2 * CGFloat.pi / CGFloat(numberOfSides)
17
18        var vertexArray = [CGPoint]()
19
20        for i in 0 ..< numberOfSides {
21            var vertex = center
22            vertex.x += cos(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
23            vertex.y += sin(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
24            vertexArray.append(vertex)
25        }
26        return vertexArray
27    }
28 }
29
30 extension CGRect {
31     var center: CGPoint { return CGPoint(x: centerX, y: centerY) } // COR OF THE RECTANGLE CENTER
32
33     var centerX: CGFloat { // RETURNS THE XCORD OF THE CENTER
34         get { return midX }
35         set { origin.x = newValue - width * 0.5 }
36     }
37
38     var centerY: CGFloat { // THE Y-COORDINATE = RECTANGLES CENTER -> NOTE: ACTS AS A SETTABLE MIDY -> RETURNS: THE Y-COORDINATE OF THE CENTER
39         get { return midY }
40         set { origin.y = newValue - height * 0.5 }
41     }
42 }
43
```

