Swift CAShapeLayer Sample Code

ddas.tech/swift-cashapelayer-sample-code/

November 19, 2022

Created By: Debasis Das (19-Nov-2022)

Initializing the Root Layer

- In this post we will explore the CAShapeLayer and create different sample CAShapeLayer and in some case animate a few of the properties
- We will create and add a rootLayer which is a CALayer and set the rootLayer as the layer of the view.
- Once the rootLayer is created, we will call different function that will create different CAShapeLayer and will add to the root layer

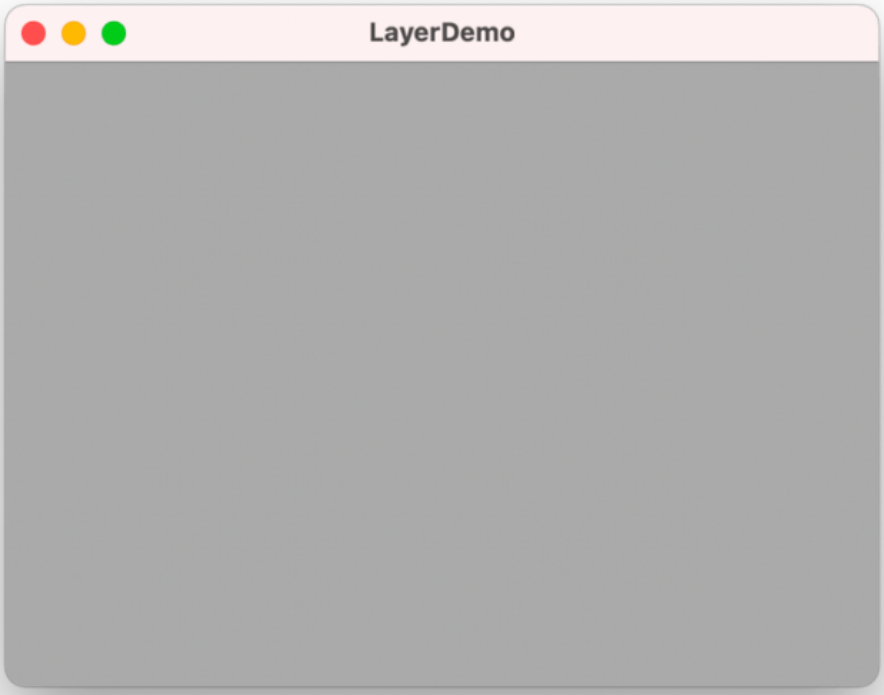
Lets begin with initializing the rootLayer in the make it the layer for the window.contentView

We will call the initializeViewWithRootLayer from the viewDidLoad or windowDidLoad based on whether the UI is driven by a NSViewController or a NSWindowController.

```
let rootLayer = CALayer()

func applicationDidFinishLaunching(_ aNotification: Notification) {
        self.initializeViewWithRootLayer()
//        self.linesShapeLayers()
//        self.linesShapeLayersLineDashPattern()
//        self.differentLineCaps()
//        self.marchingAntAnimationUsingCAShapeLayer()
//        self.circleShapeLayer()
//        self.triangleCAShapeLayer()
//        self.rectangeCAShapeLayer()
//        self.animateCAShapeLayerDrawing()
    }


func initializeViewWithRootLayer(){
        if let frame = self.window.contentView?.frame{
            rootLayer.frame = frame
            rootLayer.backgroundColor = NSColor.lightGray.cgColor
            self.window.contentView?.layer = rootLayer
            self.window.contentView?.wantsLayer = true
        }
}
```



After initializing the root layer and setting it as the layer for the window.contentView

Adding a Line as a path of the CAShapeLayer

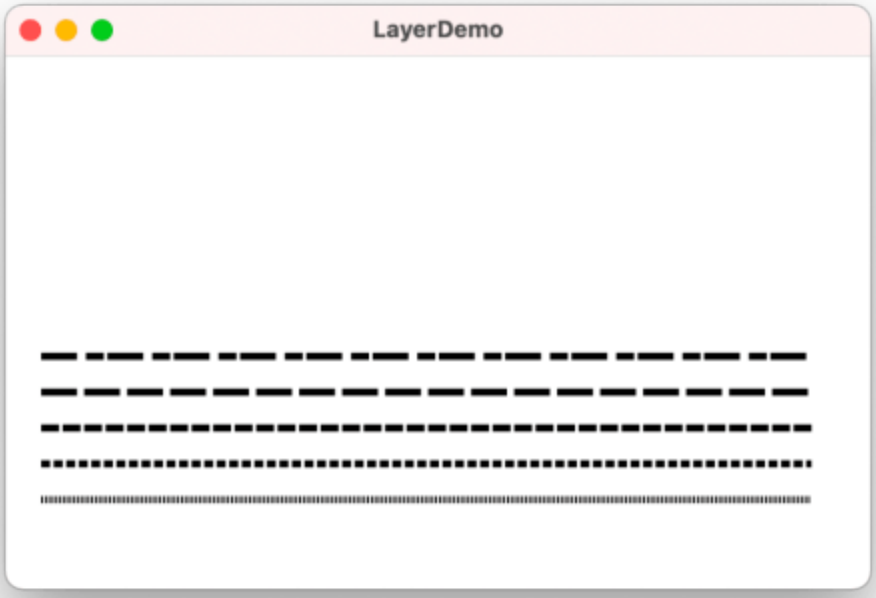
```
func linesShapeLayers(){
        let shapeLayer = CAShapeLayer()
        shapeLayer.strokeColor = NSColor.black.cgColor
        shapeLayer.lineWidth = 10
        let path = CGMutablePath()
        path.addLines(between: [CGPoint(x: 20, y: 50),CGPoint(x: 250, y: 50)])
        shapeLayer.path = path
        rootLayer.addSublayer(shapeLayer)
    }
```

Adding a line to the path property of a CAShapeLayer

Different Line Dash Patterns

```
func linesShapeLayersLineDashPattern(){
        let lineDashPatterns = [
            [NSNumber(1),NSNumber(1)],
            [NSNumber(5),NSNumber(2)],
            [NSNumber(10),NSNumber(2)],
            [NSNumber(20),NSNumber(4)],
            [NSNumber(20),NSNumber(5),NSNumber(10),NSNumber(2)],

        ]
        for (idx,pattern) in lineDashPatterns.enumerated(){
            let shapeLayer = CAShapeLayer()
            shapeLayer.strokeColor = NSColor.black.cgColor
            shapeLayer.lineWidth = 4
            shapeLayer.lineDashPattern = pattern
            let path = CGMutablePath()
            let y = 50 + (idx * 20)
            path.addLines(between: [CGPoint(x: 20, y: y),CGPoint(x: 450, y: y)])
            shapeLayer.path = path
            rootLayer.addSublayer(shapeLayer)
        }
    }
```



Different Line Dash Pattern in CAShapeLayer

Line Cap Styles

```
func differentLineCaps(){
            let shapeLayer1 = CAShapeLayer()
            shapeLayer1.strokeColor = NSColor.black.cgColor
            shapeLayer1.lineWidth = 20
            shapeLayer1.lineCap = CAShapeLayerLineCap.butt
            let path1 = CGMutablePath()
            path1.addLines(between: [CGPoint(x: 20, y: 50),CGPoint(x: 250, y: 50)])
            shapeLayer1.path = path1
            rootLayer.addSublayer(shapeLayer1)

            let shapeLayer2 = CAShapeLayer()
            shapeLayer2.strokeColor = NSColor.black.cgColor
            shapeLayer2.lineWidth = 20
            shapeLayer2.lineCap = CAShapeLayerLineCap.round
            let path2 = CGMutablePath()
            path2.addLines(between: [CGPoint(x: 20, y: 100),CGPoint(x: 250, y: 100)])
            shapeLayer2.path = path2
            rootLayer.addSublayer(shapeLayer2)

            let shapeLayer3 = CAShapeLayer()
            shapeLayer3.strokeColor = NSColor.black.cgColor
            shapeLayer3.lineWidth = 20
            shapeLayer3.lineCap = CAShapeLayerLineCap.square
            let path3 = CGMutablePath()
            path3.addLines(between: [CGPoint(x: 20, y: 150),CGPoint(x: 250, y: 150)])
            shapeLayer3.path = path3
            rootLayer.addSublayer(shapeLayer3)

       }
```

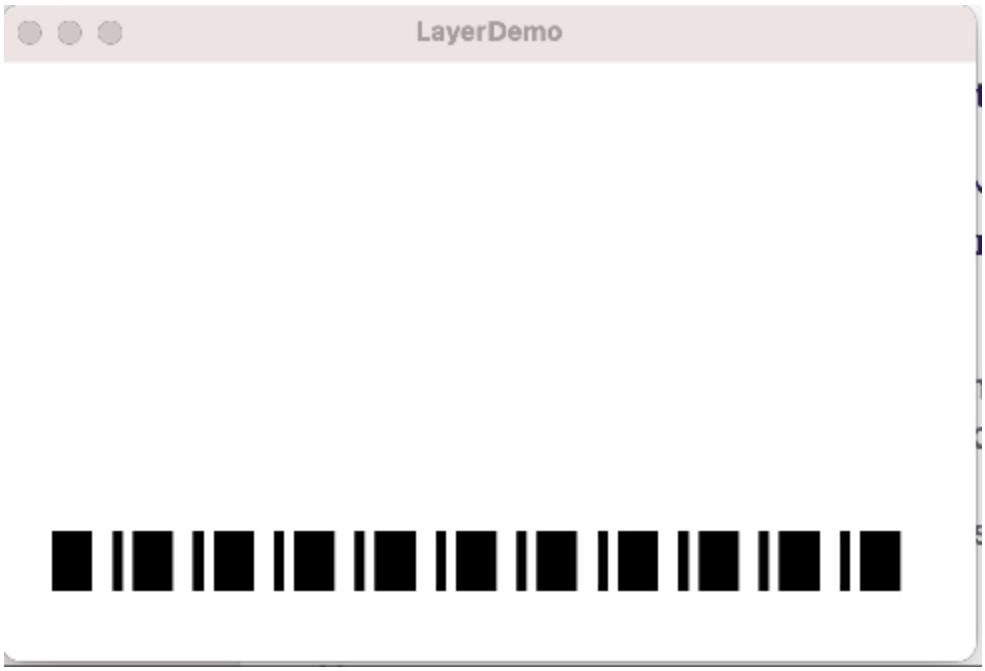
Line Cap Styles (butt, square, round)

Marching Ant Animation using CAShapeLayer

```
func marchingAntAnimationUsingCAShapeLayer(){
        let shapeLayer = CAShapeLayer()
        shapeLayer.strokeColor = NSColor.black.cgColor
        shapeLayer.lineWidth = 30
        shapeLayer.lineDashPattern = [20,10,5,5]
        let path = CGMutablePath()
        path.addLines(between: [CGPoint(x: 20, y: 50),CGPoint(x: 450, y: 50)])
        shapeLayer.path = path

        let lineDashAnimation = CABasicAnimation(keyPath: "lineDashPhase")
        lineDashAnimation.fromValue = 0
        lineDashAnimation.toValue = shapeLayer.lineDashPattern?.reduce(0) { $0 +
$1.intValue }
        lineDashAnimation.duration = 1.0
        lineDashAnimation.repeatCount = Float.greatestFiniteMagnitude

        shapeLayer.add(lineDashAnimation, forKey: nil)
        rootLayer.addSublayer(shapeLayer)
        }
```

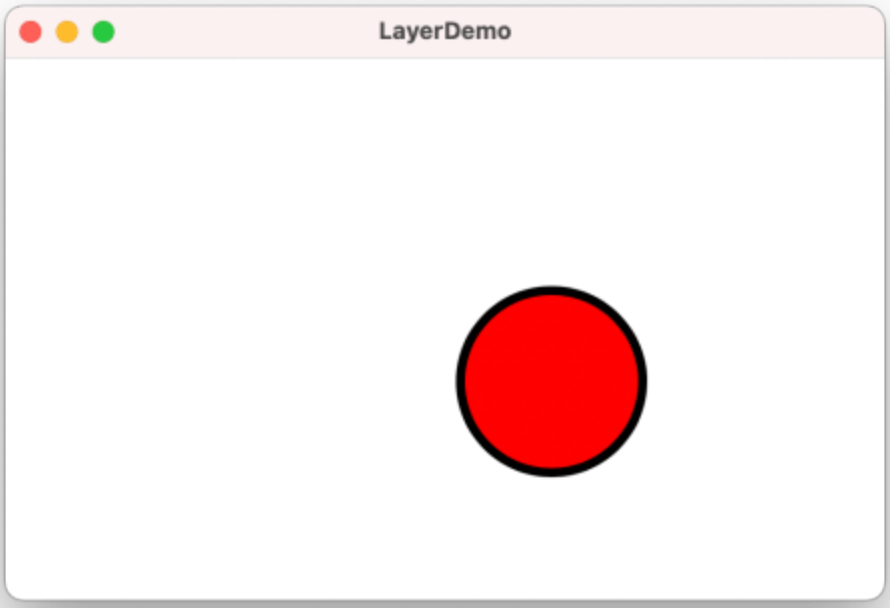


Marching Ant Animation using CAShapeLayer (lineDashAnimation)

Basic Shapes using CAShapeLayer (Circle, Rectangle, Triangle)

Circle CAShapeLayer

```
func circleShapeLayer(){
            let layer = CAShapeLayer()
            let path = CGMutablePath()
            let centerPoint = CGPoint(x: 300, y: 120)
            path.addArc(center: centerPoint, radius: 50, startAngle: CGFloat(0.0),
endAngle: CGFloat(Double.pi) * 2, clockwise: true)
            layer.path = path
            layer.strokeColor = NSColor.black.cgColor
            layer.lineWidth = 5.0
            layer.fillColor = NSColor.red.cgColor
            rootLayer.addSublayer(layer)
        }
```

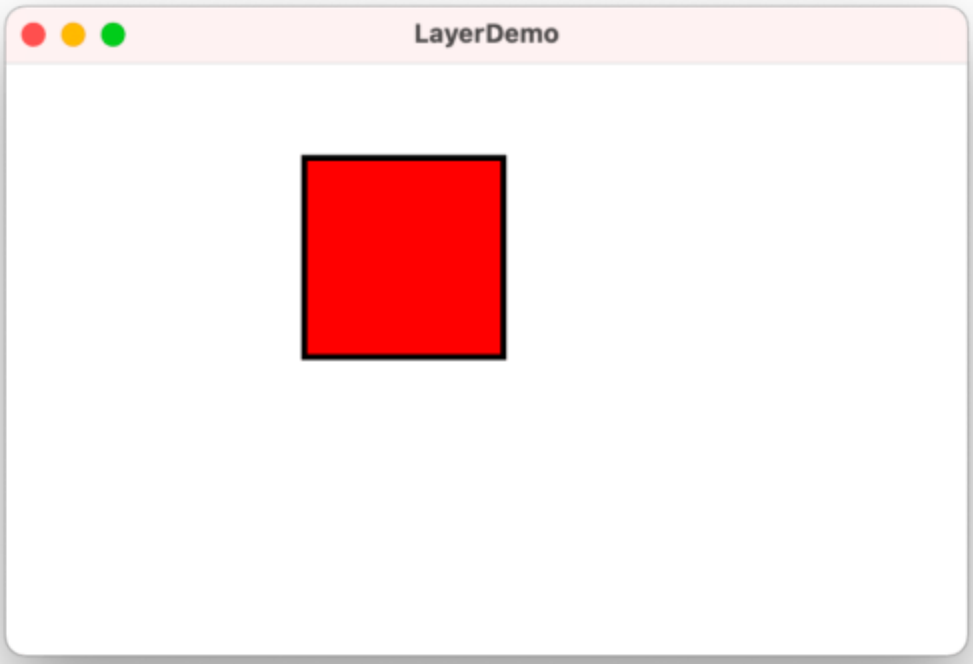


Circle CAShapeLayer

Rectangle CAShapeLayer

```
func rectangeCAShapeLayer(){
        let layer = CAShapeLayer()
        let path = CGMutablePath()
        path.addRect(CGRect(x: 150, y: 150, width: 100.0, height: 100.0))
        layer.path = path
        layer.strokeColor = NSColor.black.cgColor
        layer.lineWidth = 3.0
        layer.fillColor = NSColor.red.cgColor
        rootLayer.addSublayer(layer)

    }
```

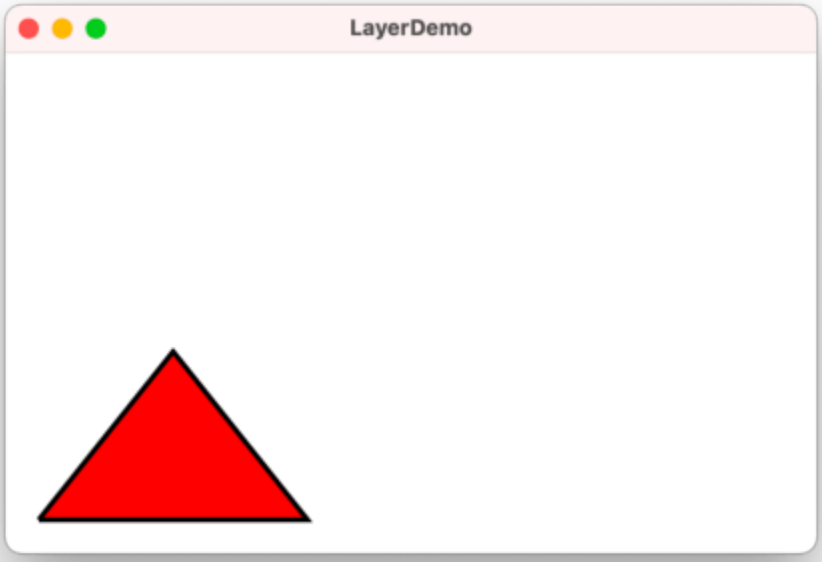


Swift – Rectangle CAShapeLayer

Triangle CAShapeLayer

```
func triangleCAShapeLayer(){
        let layer = CAShapeLayer()
        let path = CGMutablePath()
        path.move(to: CGPoint(x: 20.0, y: 20.0))
        path.addLine(to: CGPoint(x: 100.0, y: 120.0))
        path.addLine(to: CGPoint(x: 180.0, y: 20.0))
        path.addLine(to: CGPoint(x: 20.0, y: 20.0))
        layer.strokeColor = NSColor.black.cgColor
        layer.lineWidth = 3.0
        layer.fillColor = NSColor.red.cgColor
        layer.path = path
        rootLayer.addSublayer(layer)

    }
```



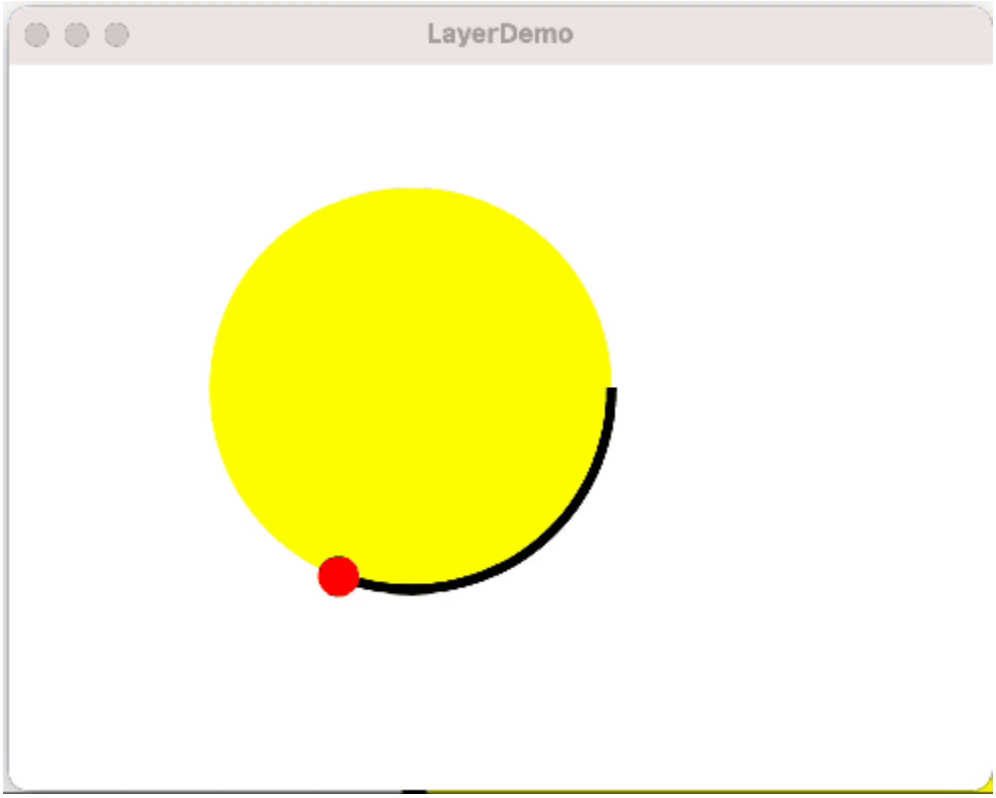
Swift CSShapeLayer – Triangle

CAShapeLayer Drawing Animation

```
func animateCAShapeLayerDrawing(){
            let layer = CAShapeLayer()
            let path = CGMutablePath()
            let centerPoint = CGPoint(x: 200, y: 200)
            path.addArc(center: centerPoint, radius: 100, startAngle: CGFloat(0.0),
endAngle: CGFloat(Double.pi) * 2, clockwise: true)
            layer.path = path
            layer.strokeColor = NSColor.black.cgColor
            layer.lineWidth = 5.0
            layer.fillColor = NSColor.yellow.cgColor
            rootLayer.addSublayer(layer)

            let pathAnimation = CABasicAnimation(keyPath: "strokeEnd")
            pathAnimation.duration = 5.0
            pathAnimation.fromValue = 0.0
            pathAnimation.toValue = 1.0
            pathAnimation.repeatCount = Float.greatestFiniteMagnitude
            layer.add(pathAnimation, forKey: "strokeEndAnimation")

            let indicatorLayer = CALayer()
            indicatorLayer.bounds = CGRect(x: 0, y: 0, width: 20.0, height: 20.0)
            indicatorLayer.cornerRadius = 10.0
            indicatorLayer.backgroundColor = NSColor.red.cgColor
            rootLayer.addSublayer(indicatorLayer)
            let indicatorAnimation = CAKeyframeAnimation(keyPath: "position")
            indicatorAnimation.duration = 5.0
            indicatorAnimation.path = path
            indicatorAnimation.calculationMode = CAAnimationCalculationMode.paced
            indicatorAnimation.repeatCount = Float.greatestFiniteMagnitude
            indicatorLayer.add(indicatorAnimation, forKey: "position")
        }
```



Swift CAShapeLayer – Drawing Animation