

```

import UIKit
import SceneKit
import ARKit
import CoreImage
import Vision
import PerspectiveTransform

class StraightViewController: UIViewController {
    var image: UIImage!
    var imageView = UIImageView()

    var UITopLeftPoint, UITopRightPoint, UIbottomRightPoint,
    UIbottomLeftPoint: CGPoint!
    override func viewDidLoad() {
        super.viewDidLoad()
        UIImageWriteToSavedPhotosAlbum(image!, self, nil, nil)
        self.view.backgroundColor = .white
        imageView.image = image
        view.addSubview(imageView)
        let height = UIScreen.main.bounds.size.height
        let width = UIScreen.main.bounds.size.width
        imageView.frame = CGRect(x: 0, y: 0, width: width, height:
height)
        drawCircles(points: [UITopLeftPoint, UITopRightPoint,
UIbottomRightPoint, UIbottomLeftPoint])
        imageView.layer.transform = reverseTranformation()
        // Do any additional setup after loading the view.
    }

    lazy var fiducialPerspective: Perspective = {
        let corners = [UITopLeftPoint!, UITopRightPoint!,
UIbottomLeftPoint!, UIbottomRightPoint!]// points of actual qr code
        return Perspective(corners)
    }()

    func reverseTranformation() -> CATransform3D {
        // the destination should be a square somewhere for desired qr
code
        var bound = view.frame
        let destination = Perspective(bound)
        return fiducialPerspective.projectiveTransform(destination:
destination)
    }
}

extension StraightViewController {
    func drawCircles(points: [CGPoint]) {
        for point in points {
            drawCircle(point: point)
        }
    }
}

```

```
    }  
  }  
  private func drawCircle(point: CGPoint) {  
    let circlePath = UIBezierPath(arcCenter: point, radius: 3.0,  
startAngle: CGFloat(0), endAngle: CGFloat(Double.pi * 2.0), clockwise:  
true)  
    let shapeLayer = CAShapeLayer()  
    shapeLayer.path = circlePath.cgPath  
    shapeLayer.fillColor = UIColor.green.cgColor  
    shapeLayer.strokeColor = UIColor.black.cgColor  
    view.layer.addSublayer(shapeLayer)  
  }  
}
```