```swift
import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var decreaseButton: UIButton!
    @IBOutlet weak var increaseButton: UIButton!
    @IBOutlet weak var vertexCountLabel: UILabel!
    @IBOutlet weak var polygonView: PolygonView!

    lazy var polygonModel: PolygonShape = {
        let polygon = PolygonShape()
        polygon.numberOfSides = 8
        return polygon
    }()

    @IBAction func increaseSides(_ sender: UIButton) {
        polygonModel.numberOfSides += 1
        updateUI()
    }

    @IBAction func decreaseSides(_ sender: UIButton) {
        polygonModel.numberOfSides -= 1
        updateUI()
    }

    private func updateUI() {
        let defaults: UserDefaults = UserDefaults.standard
        defaults.set(polygonModel.numberOfSides, forKey: "numerOfSides")
        defaults.synchronize()

        polygonView.setNeedsDisplay()
        vertexCountLabel.text = "\(polygonModel.numberOfSides)"
        decreaseButton.isEnabled = polygonModel.numberOfSides == 3 ? false : true
        increaseButton.isEnabled = polygonModel.numberOfSides == 12 ? false : true
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        let defaults: UserDefaults = UserDefaults.standard
        if let numberOfSides = defaults.object(forKey: "numerOfSides") as? Int {
            polygonModel.numberOfSides = numberOfSides
        }
        polygonView.delegate = polygonModel
        updateUI()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

```swift
//
//  PolygonView.swift
//  HelloPoly
//
//  Created by Me on 10/01/2019.
//  Copyright © 2019 UCD. All rights reserved.
//

import UIKit

class PolygonView: UIView {
    var delegate: PolygonProtocol? = nil
    var lineWidth: Float = 2.0
    var strokeColor: UIColor = UIColor.blue
    var fillColor: UIColor = UIColor.green.withAlphaComponent(0.5)

    // Only override drawRect: if you perform custom drawing.
    // An empty implementation adversely affects performance during animation.
    override func draw(_ rect: CGRect) {
        // Drawing code

        let insetRect = rect.insetBy(dx: CGFloat(lineWidth / 2.0), dy: CGFloat(lineWidth / 2.0))

        if let vertices = delegate?.pointsInRect(insetRect) {
            fillColor.setFill()
            strokeColor.setStroke()
            let path = UIBezierPath()
            path.move(to: vertices[0])
            for vertex in vertices[1..<vertices.count] {
                path.addLine(to: vertex)
            }
            path.close()
            path.stroke()
            path.fill()

        }
    }
}

```

```swift
import UIKit

protocol PolygonProtocol {
    func pointsInRect(_ rect: CGRect) -> [CGPoint]
}

class PolygonShape: NSObject, PolygonProtocol {
    private let names = ["Triangle", "Square", "Pentagon", "Hexagon", "Heptagon", "Octagon", "Nonagon", "Decagon", "Hendecagon", "Dodecagon"]

    var numberOfSides: Int = 8 {
        didSet {
            if !(3 ... 12).contains(numberOfSides) {
                if oldValue <= 3 {
                    numberOfSides = 3
                } else {
                    numberOfSides = 12
                }
            }
        }
    }

    var name: String { return names[numberOfSides - 3] }

    override var description: String { return name }

    func pointsInRect(_ rect: CGRect) -> [CGPoint] {
        let center = rect.center
        let radius = min(rect.size.width, rect.size.height) / 2.0
        let arc = 2 * CGFloat.pi / CGFloat(numberOfSides)

        var vertexArray = [CGPoint]()
        for i in 0 ..< numberOfSides {
            var vertex = center
            vertex.x += cos(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
            vertex.y += sin(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
            vertexArray.append(vertex)
        }
      return vertexArray
    }
}

var polygonPoints = { (rect: CGRect, numberOfSides: Int) -> [CGPoint] in
    let center = rect.center
    let radius = min(rect.size.width, rect.size.height) / 2.0
    let arc = 2 * CGFloat.pi / CGFloat(numberOfSides)

    var vertexArray = [CGPoint]()
    for i in 0 ..< numberOfSides {
        var vertex = center
        vertex.x += cos(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
        vertex.y += sin(arc * CGFloat(i) - 2 * CGFloat.pi) * radius
        vertexArray.append(vertex)
    }
    return vertexArray
}

extension CGRect {
    var center: CGPoint {
        return CGPoint(x: size.width / 2.0 + origin.x, y: size.height / 2.0 + origin.y)
    }
}
```

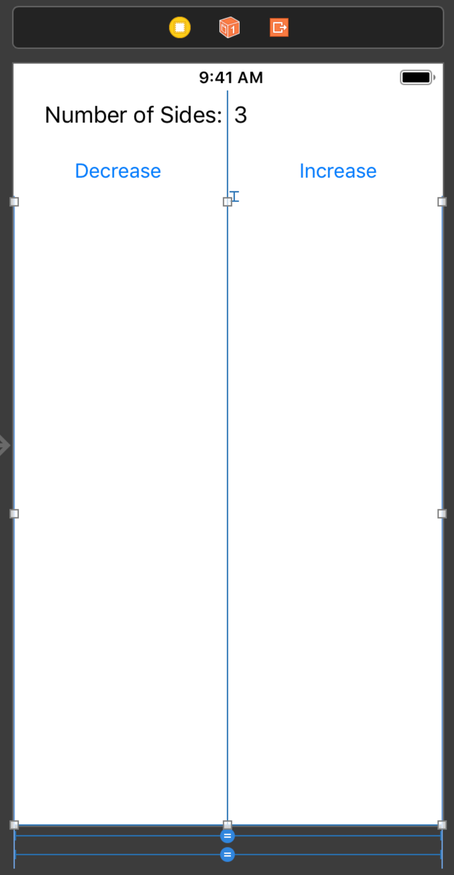
HelloPoly 〉 iPhone XS Max

HelloPoly: **Ready** | Today at 11:11 AM

HelloPoly 〉 HelloPoly 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Scene 〉 View Controller 〉 View 〉 Polygon View

- HelloPoly
  - HelloPoly
    - AppDelegate.swift
    - ViewController.swift
    - PolygonView.swift
    - PolygonShape.swift
    - Main.storyboard
    - Assets.xcassets
    - LaunchScreen.storyboard
    - Info.plist
  - Products

- View Controller Scene
  - View Controller
    - View
      - Safe Area
      - Stack View
        - Stack View
          - Number of Sides:
          - Vertex Count Label
        - Stack View
          - Decrease
          - Increase Button
      - Polygon View
      - Constraints
        - Safe Area.bottom = Poly...
        - Stack View.top = Safe Ar...
        - Stack View.centerX = Saf...
        - Stack View.width = width
        - Polygon View.centerX =...
        - Polygon View.width = wi...
        - Polygon View.top = Stac...
  - First Responder
  - Exit
  - Storyboard Entry Point



View as: iPhone SE (wC hR) — 100%

Custom Class
Class: PolygonView
Module: HelloPoly
Inherit Module From Target

Identity
Restoration ID

User Defined Runtime Attributes

Key Path	Type	Value
lineWidth	Number	2
strokeColor	Color	Custom
fillColor	Color	Light Gray...

Document
Label: Xcode Specific Label
Object ID: ZrE-uC-xtu
Lock: Inherited - (Nothing)
Localizer Hint: Comment For Localizer

Accessibility
Accessibility: Enabled
Label: Label
Hint: Hint
Identifier: Identifier
Traits: Button, Image, Static Text, Search Field, Plays Sound, Keyboard Key, Summary Element, User Interaction Enabled, Updates Frequently, Starts Media Session, Adjustable, Allows Direct Interaction, Causes Page Turn, Header, Link, Selected