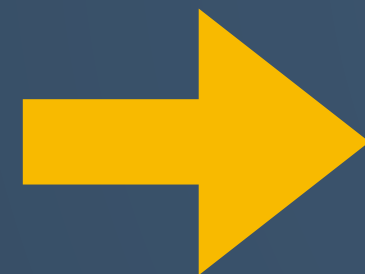
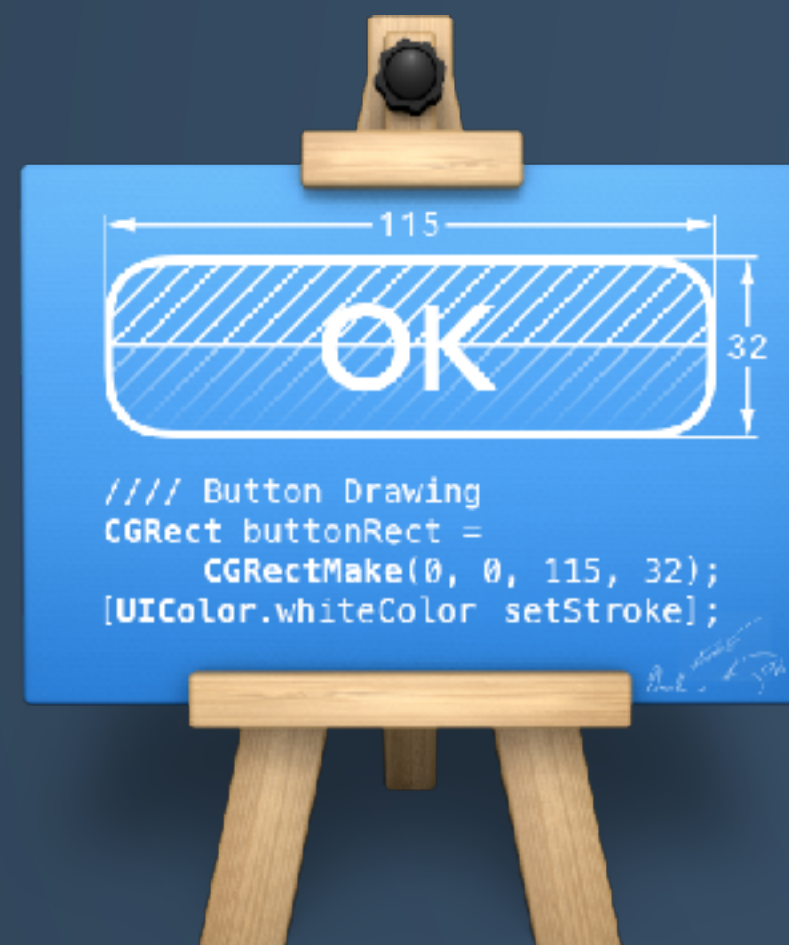


Resolution Independence with PaintCode



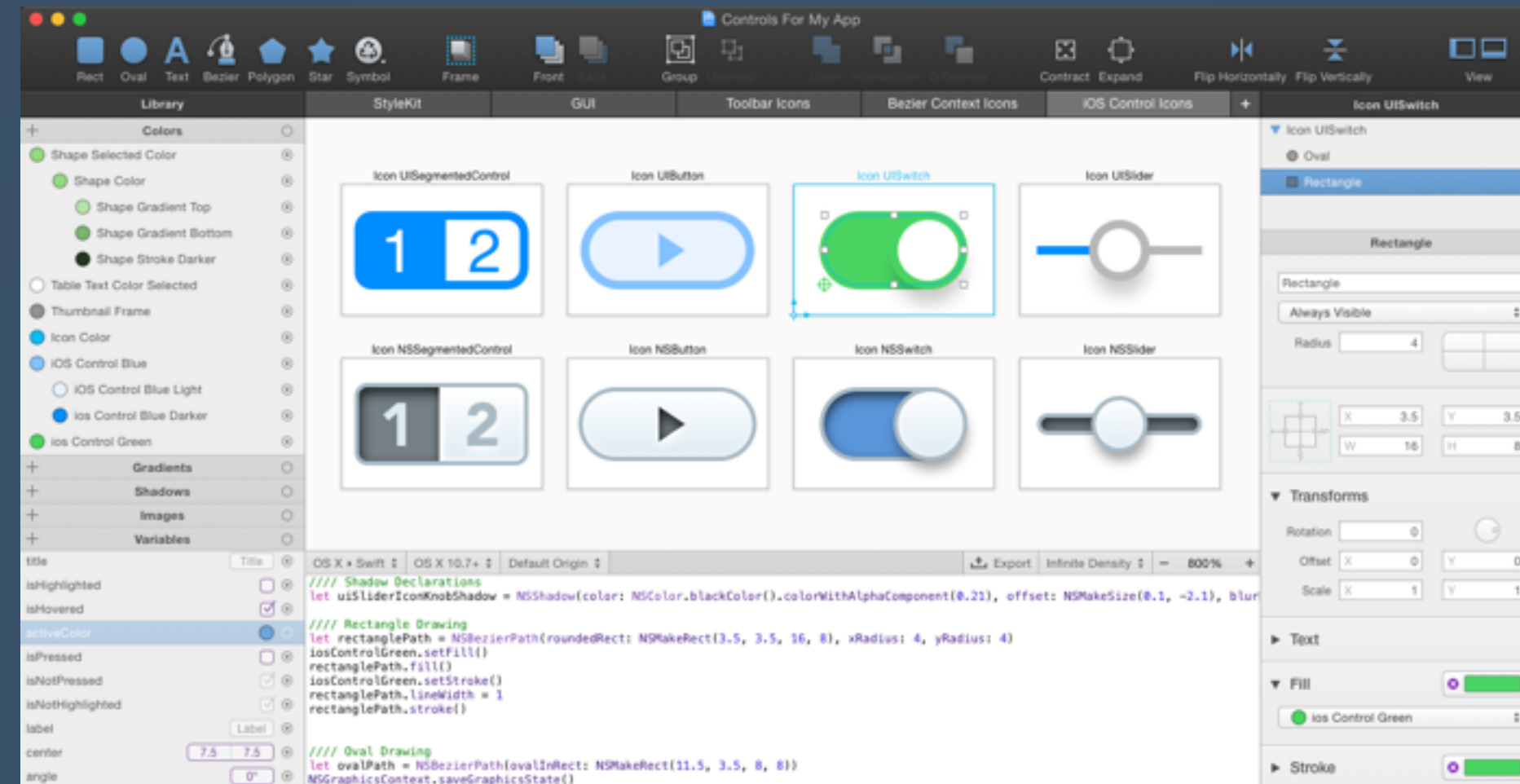
```
//// Oval Drawing
var ovalPath = UIBezierPath(ovalInRect: CGRectMake(44, 21, 81, 81))
UIColor.grayColor().setFill()
ovalPath.fill()

//// Rectangle Drawing
let rectanglePath = UIBezierPath(rect: CGRectMake(62, 39, 45, 45))
UIColor.greenColor().setFill()
rectanglePath.fill()

//// Bezier Drawing
var bezierPath = UIBezierPath()
bezierPath.moveToPoint(CGPointMake(165.5, 29.5))
bezierPath.addCurveToPoint(CGPointMake(193.5, 47.5))
bezierPath.addCurveToPoint(CGPointMake(158.5, 47.5))
UIColor.blackColor().setStroke()
bezierPath.lineWidth = 1
bezierPath.stroke()
```

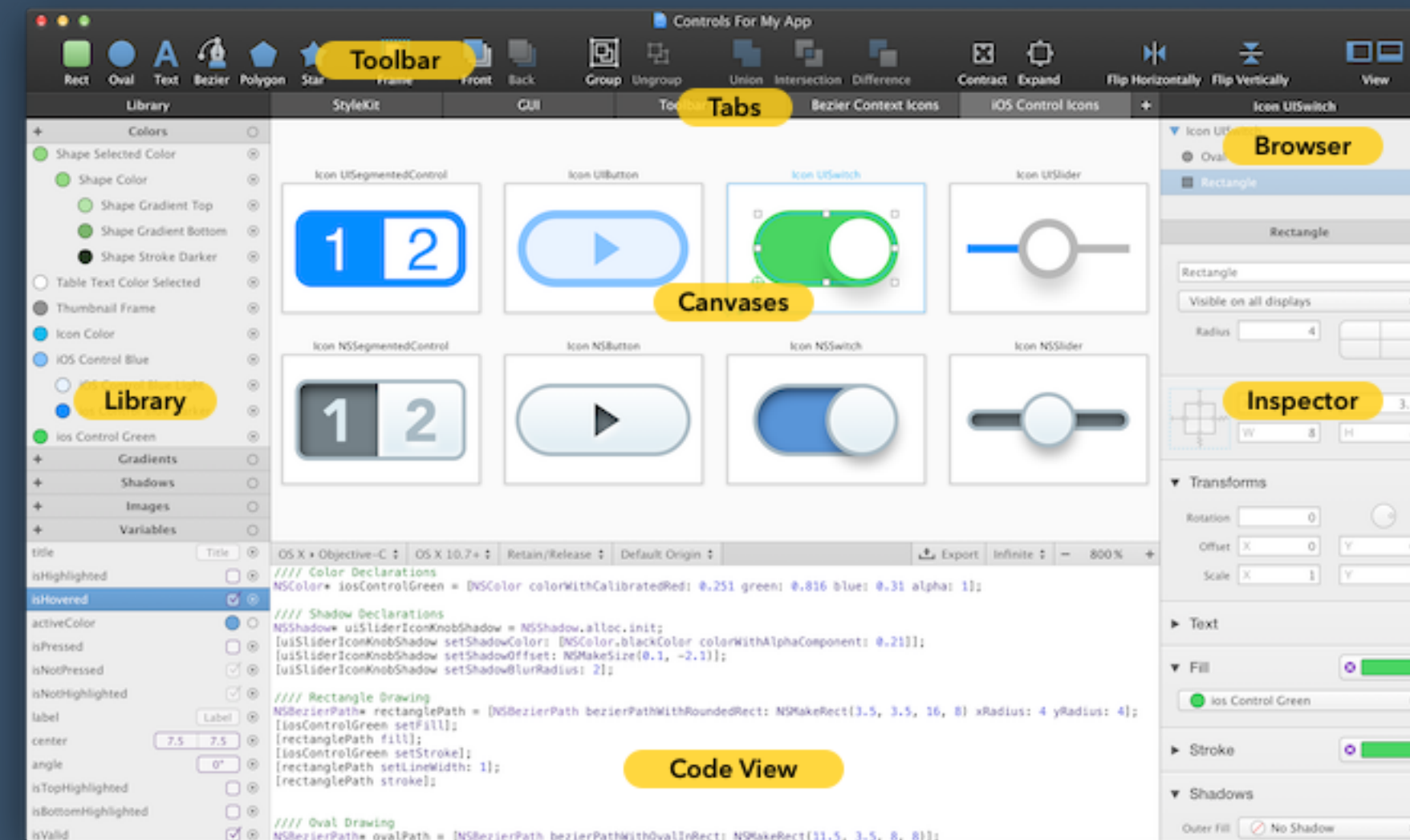
PaintCode

www.paintcodeapp.com



- Vector drawing tools for creating UI graphics
- Instantly generate Objective-C, Swift, or C# code from your drawings
- Export a single StyleKit class that contains all the drawing code as class methods

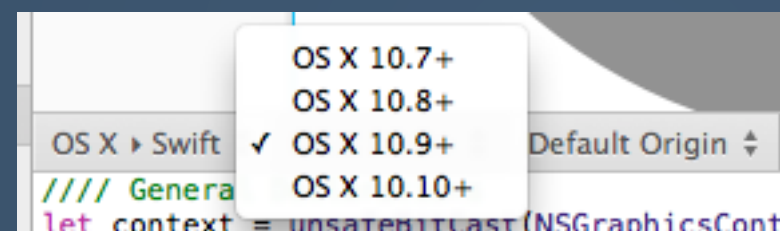
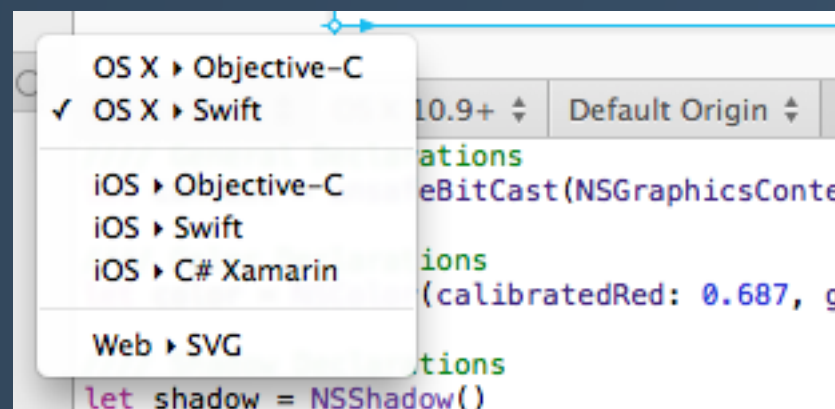
Main window



- Library of colors, gradients, shadows, and other attributes to define a shape
- Realtime code generation for the selected Canvas displayed in Code View

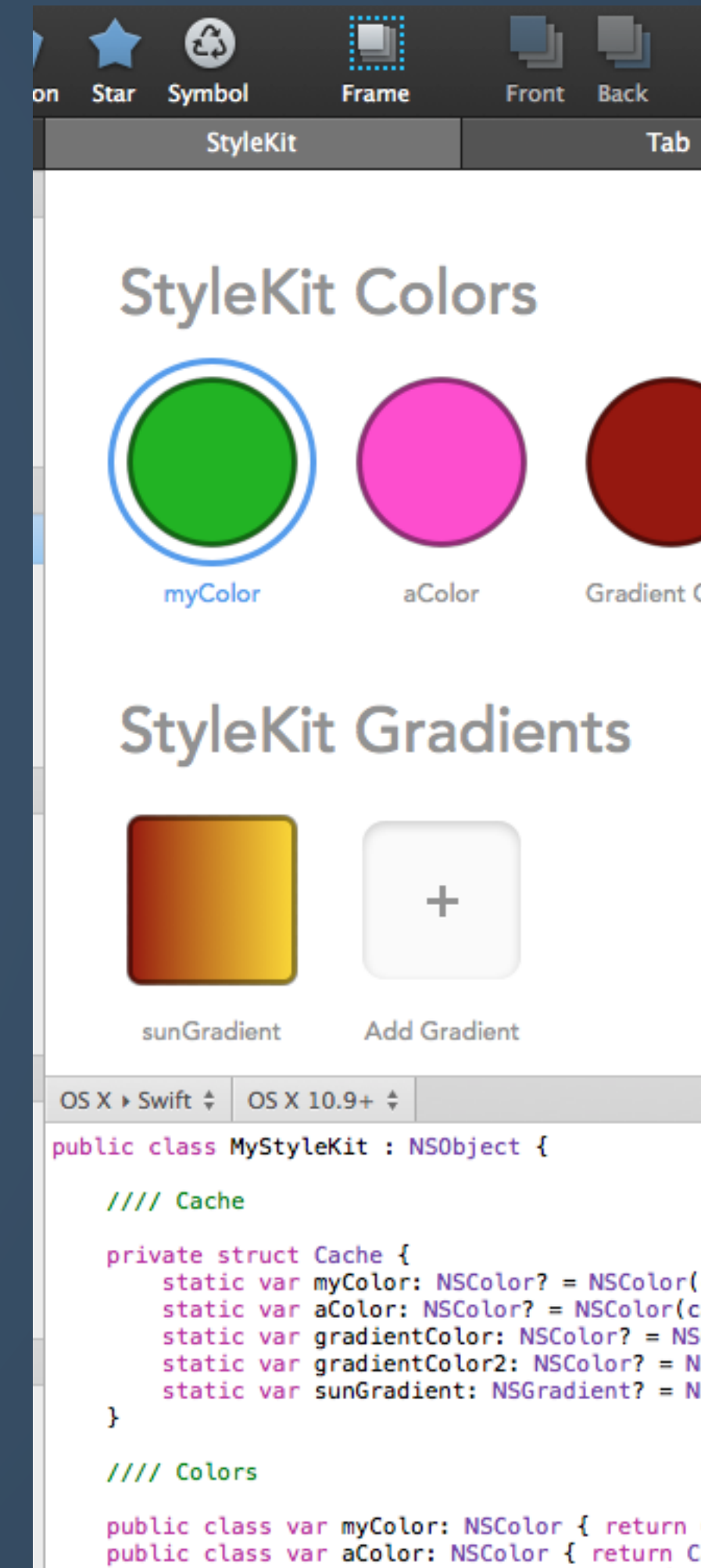
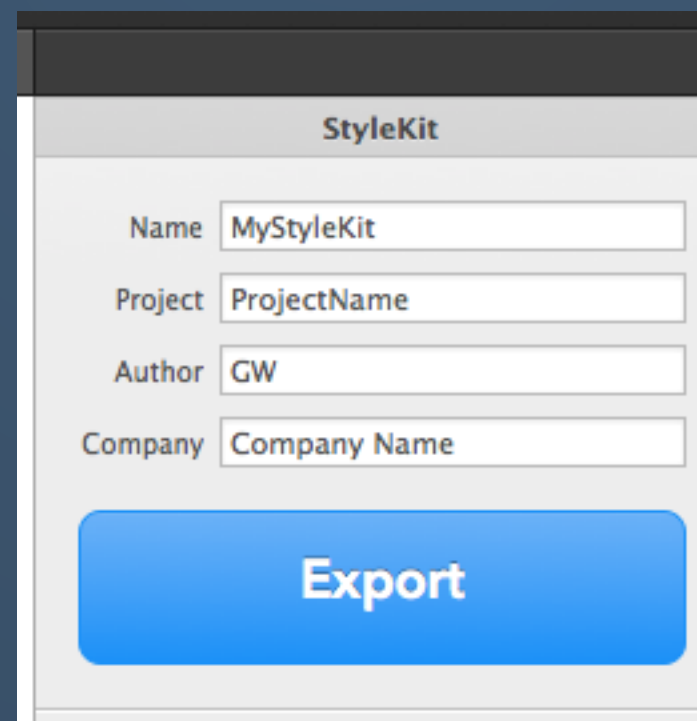
Generate code

- Automatically create code from drawing on the canvas
- Specify each drawing origin by dragging the origin in the canvas



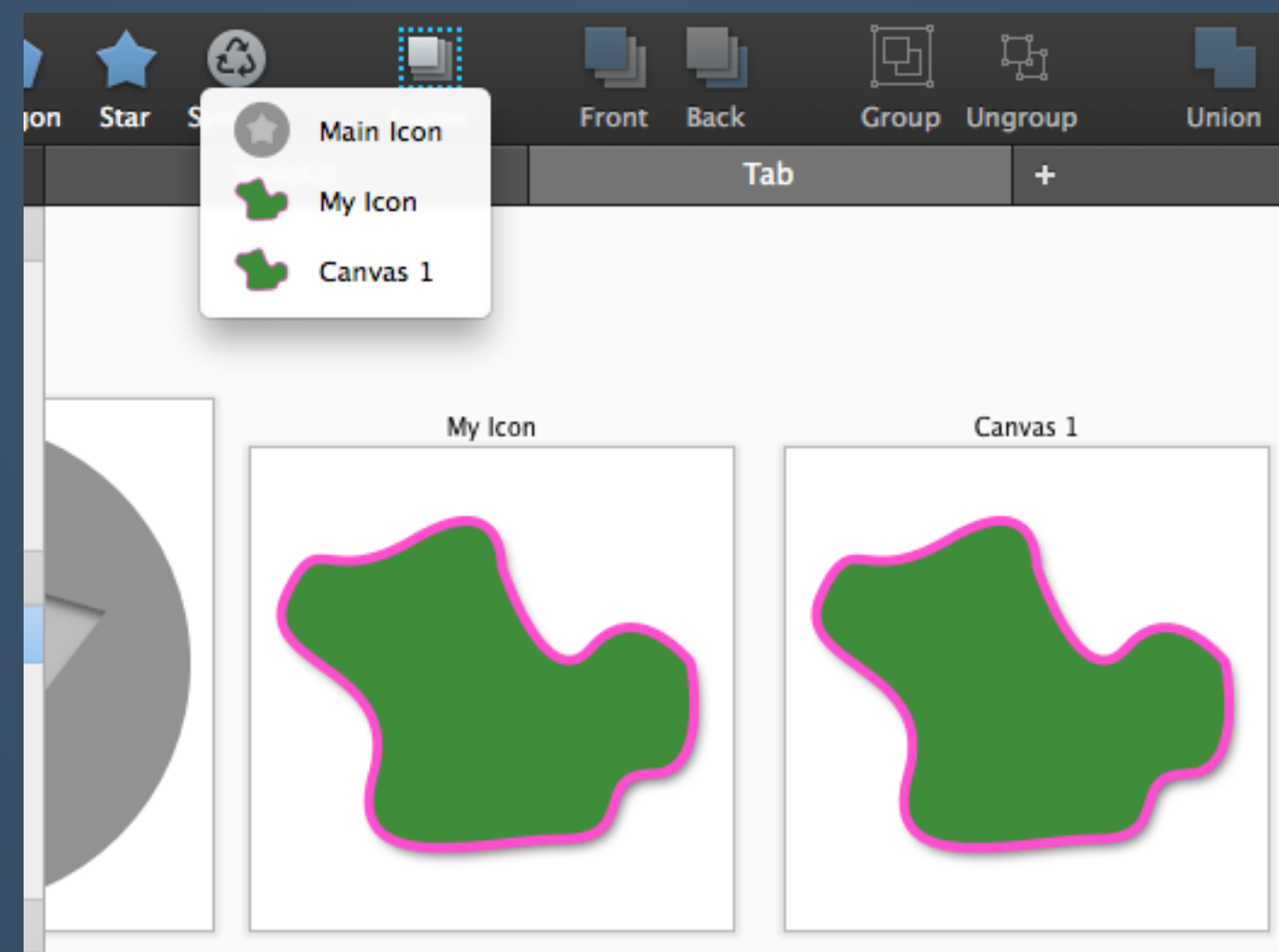
StyleKit

- Export code of all the drawings, colors, gradients, shadows, icons, and images as a single class
- Command-R in PaintCode to re-export drawing code to Xcode
- Great for fast design prototyping



Symbols

- Easily reuse drawings from other canvases
- Create a new canvas then select the "Symbol" tool
- Symbols behave like other shapes such as rectangles, ovals, etc.
- Changes in the original drawing are reflected in the symbol



Demo 