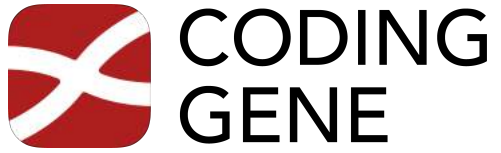


Macoun



Pixel

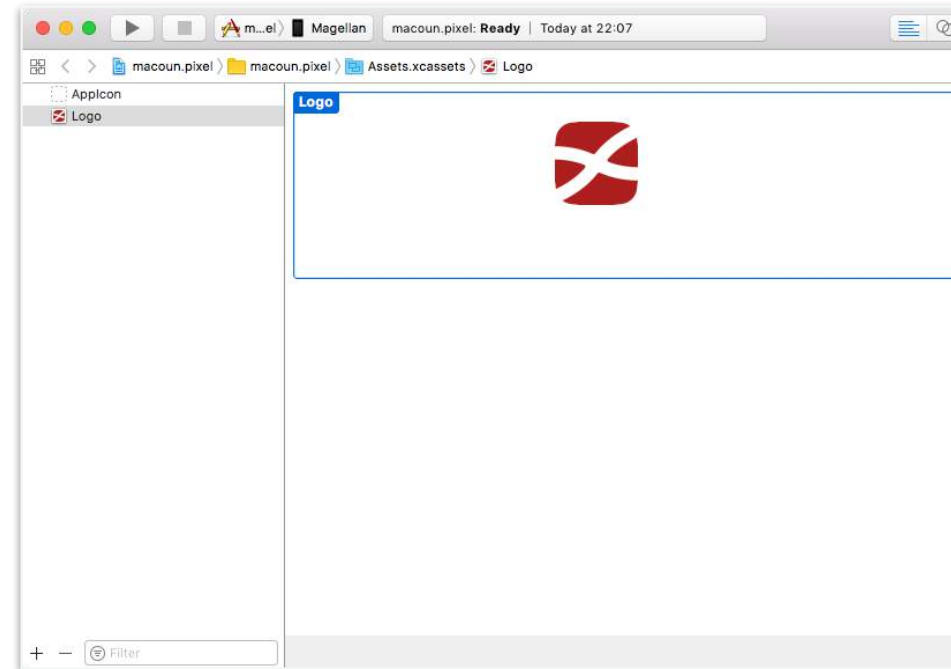
Wolfgang Muhsal
@Funkybit



PIXELSCHUBSER

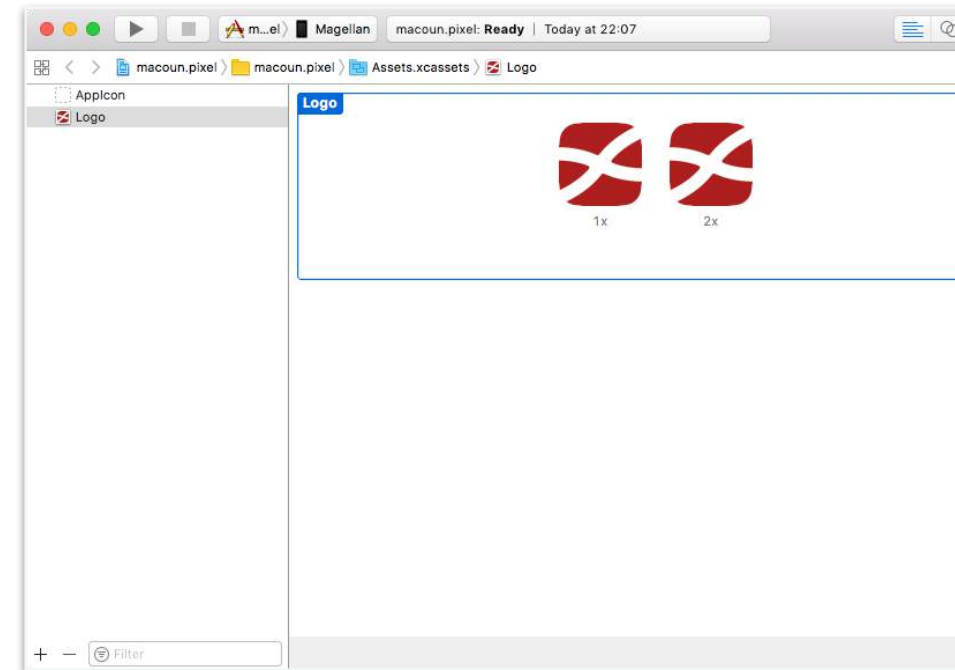
@1x - Non-Retina Welt

- Bis 2010 war alles einfach
- schöne dicke fette Pixel



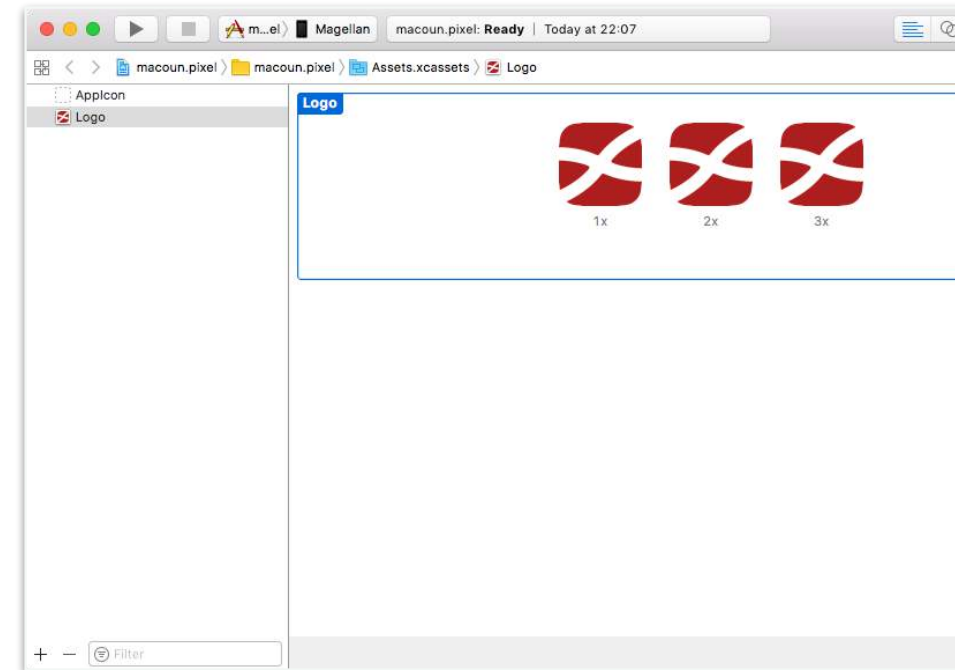
@2x - Schöne neue Retina Welt

- 4 mal so viele Pixel
- plus die kleineren Assets

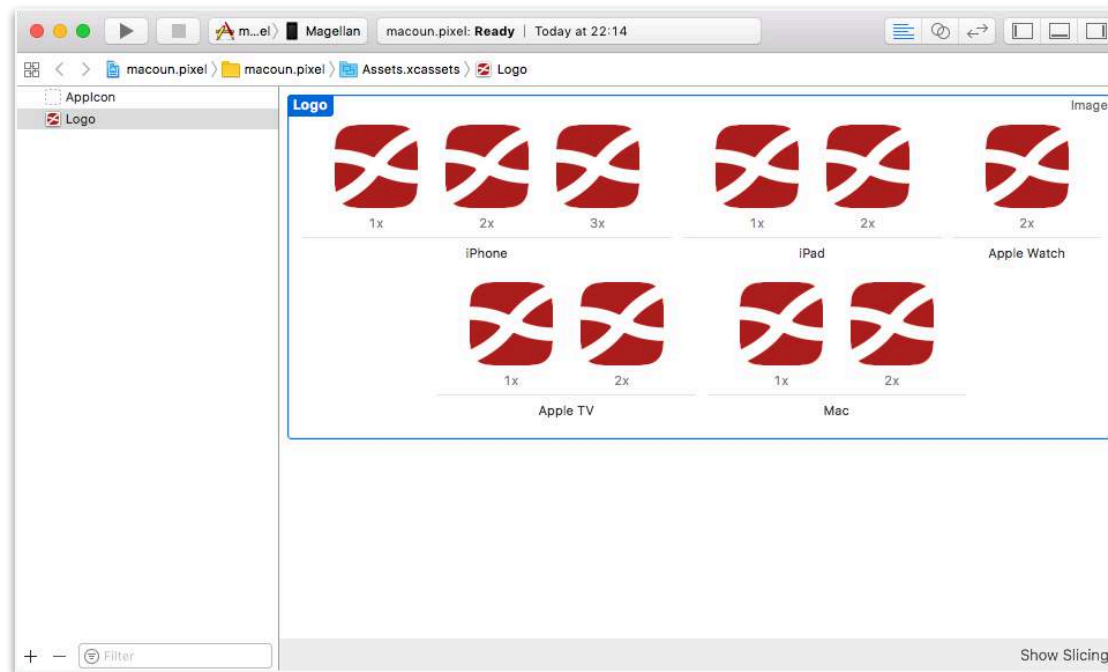


@3x - Super-Retina Welt

- 9 mal so viele Pixel
- plus die kleineren Assets



Varianten-Explosion





Was kann schon schief gehen...

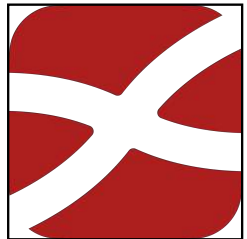
Grafiker...



- @1x : 99x98



- @2x : 200x201



- @3x : 402x403



Konsistenz

Mal mit Rand, mal ohne...

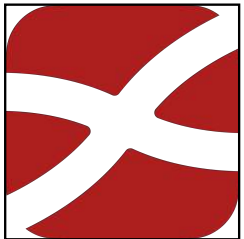


- @1x : 99x98

- einheitlicher Umlauf/Rand



- @2x : 200x201



- @3x : 402x403

Konsistenz



- @1x : 99x98

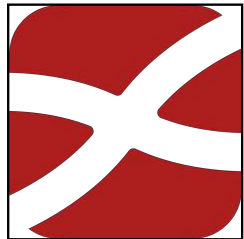
- einheitlicher Umlauf/Rand

- saubere Vielfache



- @2x : 200x201

98 x 2 ist nicht 201...



- @3x : 402x403

1,2,3



- @1x : 99x98

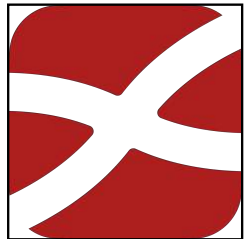
- einheitlicher Umlauf/Rand

- saubere Vielfache



- @2x : 200x201

- 1,2,3 nicht 1,2,4

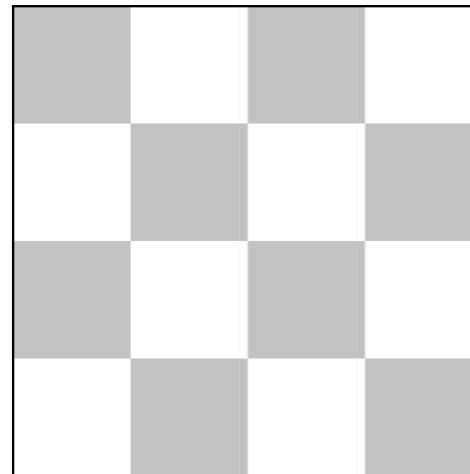
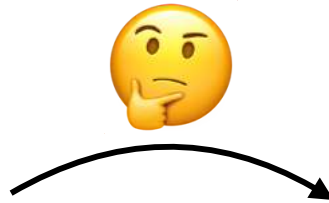
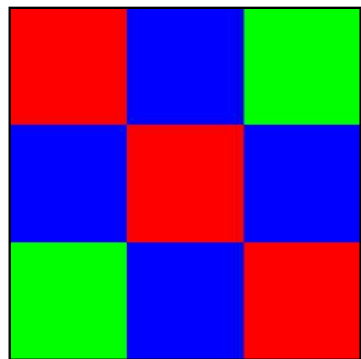


- @3x : 402x403

Was soll dann mal @4x sein? 800?

Die Mitte

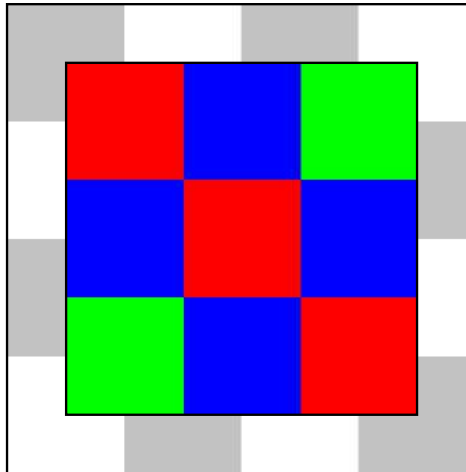
- Das Eckige muss mittig ins Eckige



•

Die Mitte

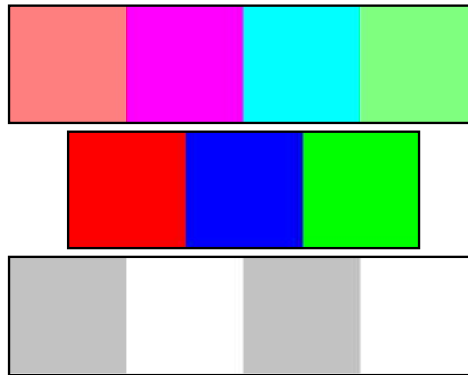
- Das Eckige muss mittig ins Eckige



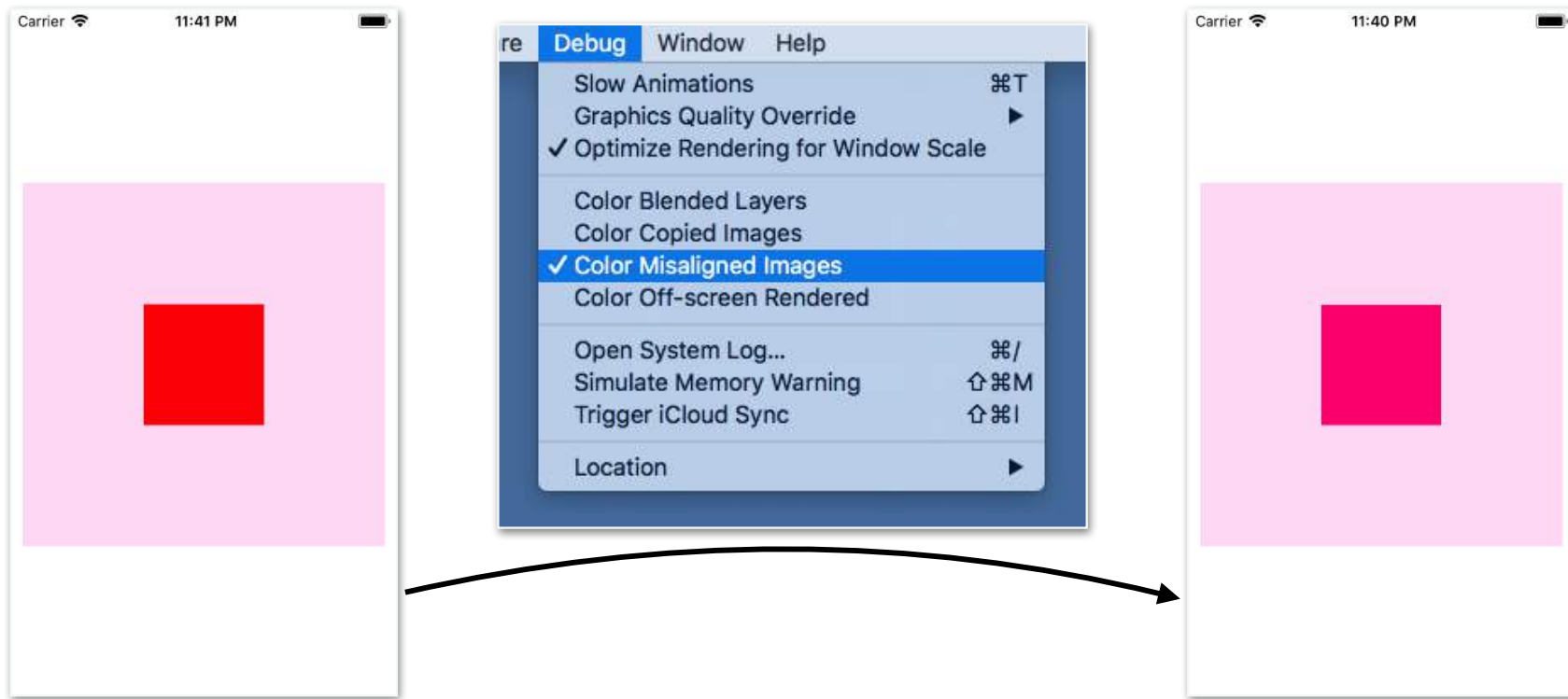
•

Die Mitte

- Bildschirm
- Bild
- Canvas



Misaligned Images



Misaligned Text

Lorem Ip

Lorem Ip

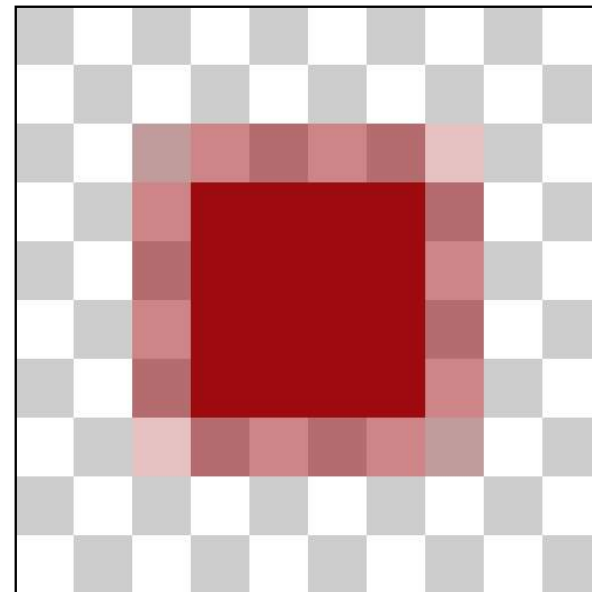
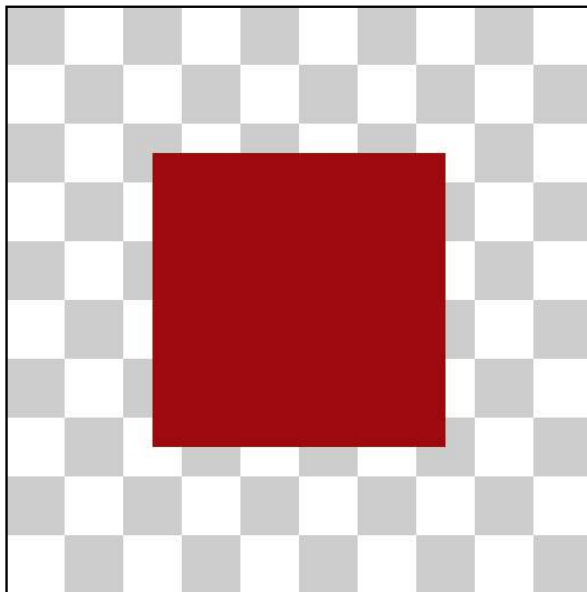
Kleine Goldene Regel

In non-retina alle Maße
in geraden Zahlen.

- 8, 16, 24
- 50, 100, 150
- passt immer in z.B. 100x100

Aber die Quelle muss stimmen

Vektor-Programme wie z.B. Illustrator orientieren sich nicht an Pixeln.

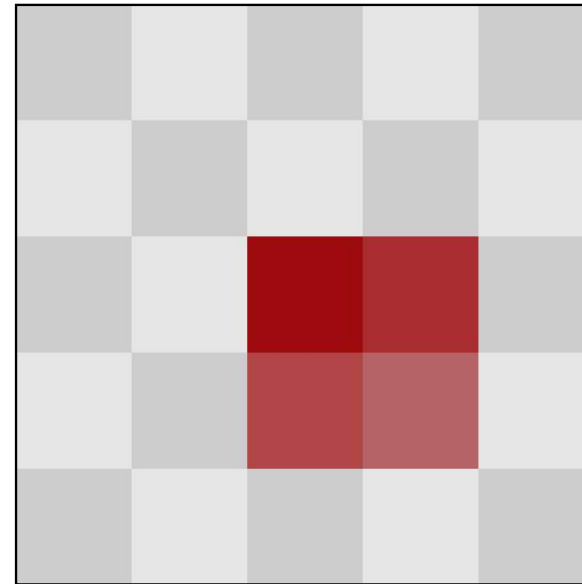
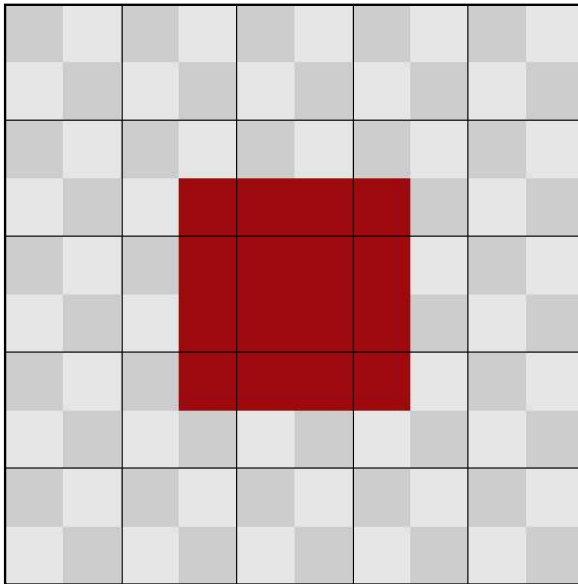


Retina als Default

- Grafiker entwerfen gerne direkt in Retina
 - Exports sehen einfach besser aus
 - zugegeben: Non-Retina ist nahezu ausgestorben

Retina als Default

In Retina entworfen und runter skaliert...



Zu viel Pixel-Perfektion

... kann auch komisch aussehen.



Retina als Default

Setze bitte die Schriftgröße
für den SubTitle auf 30.

Alles klar, wird gemacht!

Product Title

New Subtitle

Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua

Vermaßung

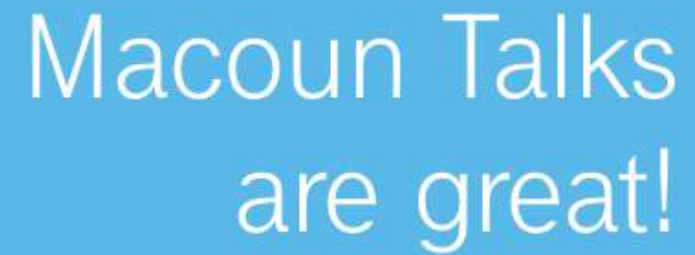
- Style/Design Spec sagt
 - Schrift: 30pt
 - Margins: 18px



Macoun Talks are great!

Vermaßung

- Style/Design Spec sagt
 - Schrift: 30pt
 - Margins: 18px



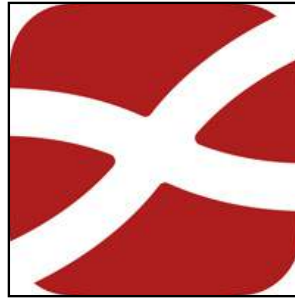
Macoun Talks
are great!

Punkte sind die neuen Pixel

- Pixel (px)
 - Antrag: Aus dem Wörterbuch streichen
- Punkte (pt)
 - wenn's dann mal alle verstanden haben...

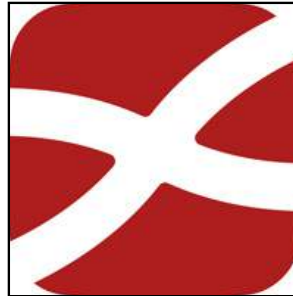
Bildformat

- geliefert:
- JPG, weißer Hintergrund



Bildformat

- geliefert:
 - JPG, weißer Hintergrund
- Wie hat sich der Grafiker die UI gewünscht?



Bildformat

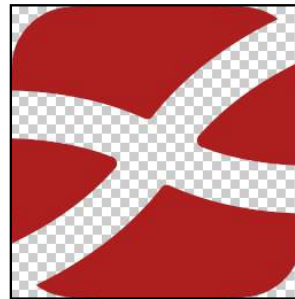
- JPG, weißer Hintergrund

- 12 kb



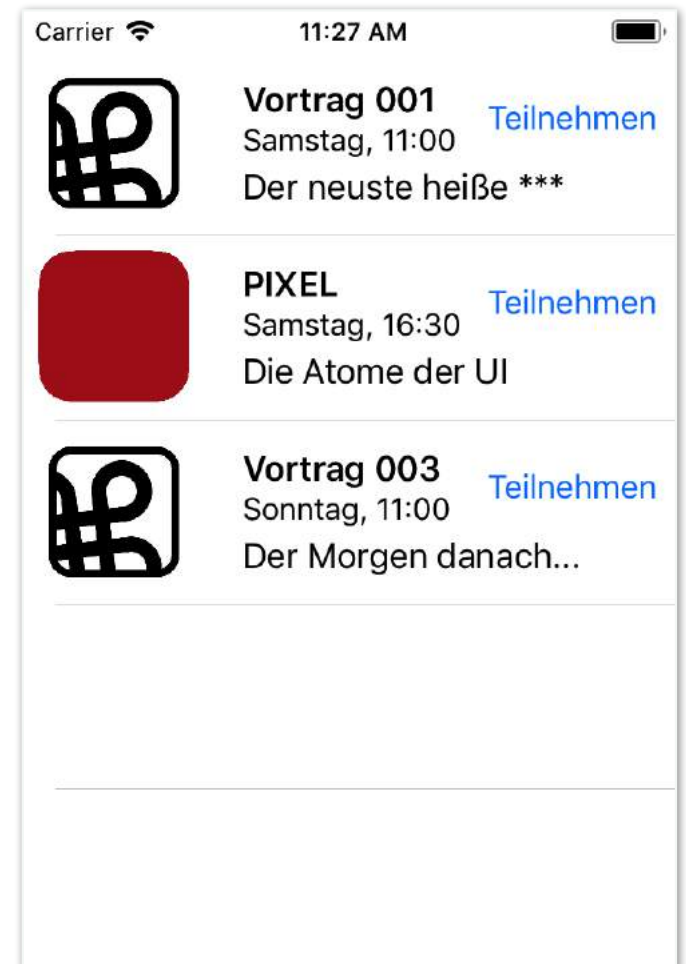
- PNG mit Transparenz

- 8kb



Transparenz kostet

- Irgendeine App mit einer TableView
- Eigentlich alles gut
 - aber beim Scrollen ruckelt es...
 - keine 60 Frames



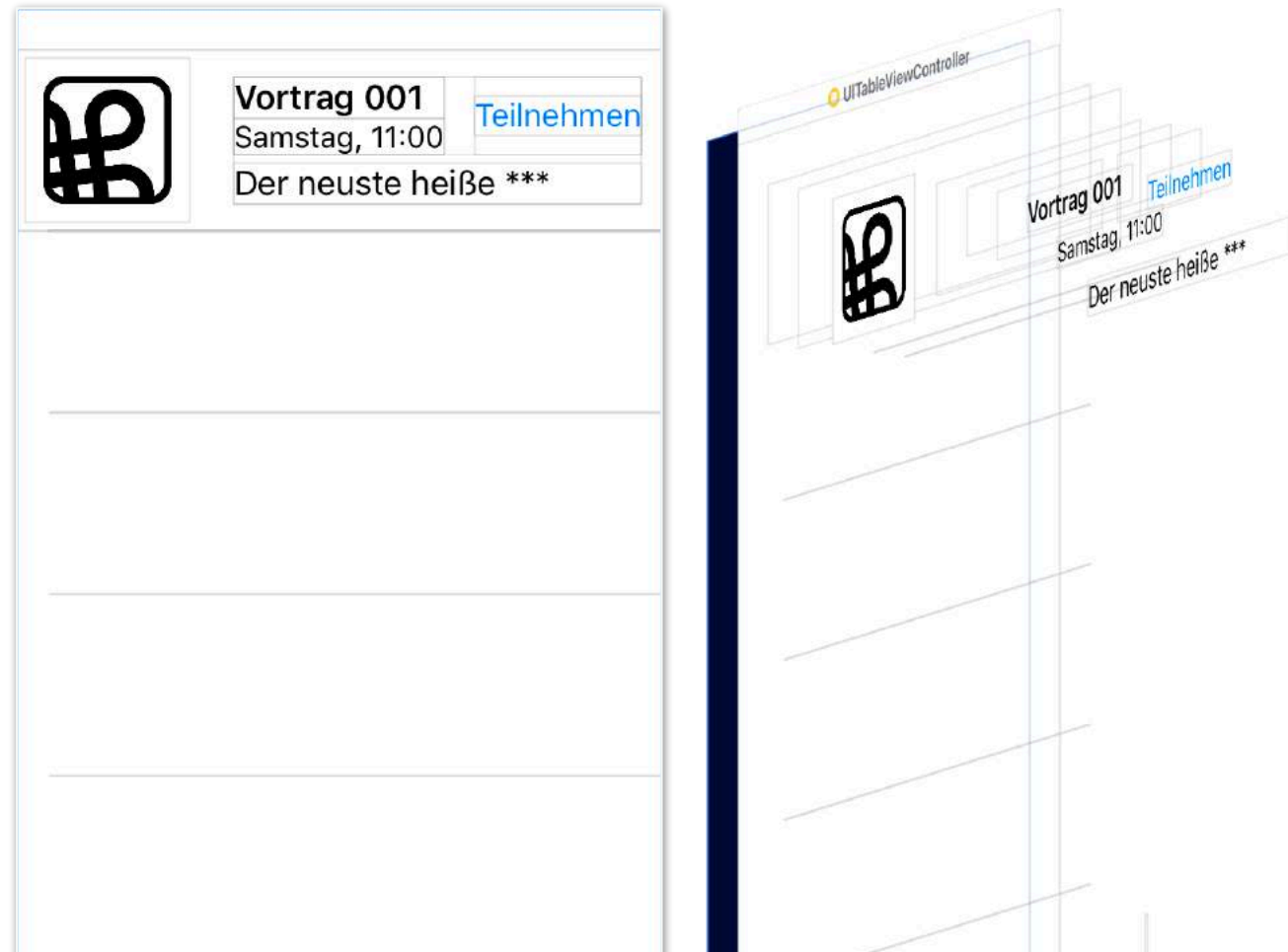
Transparenz kostet

- Die View-Hierarchie
 - ok-ish...



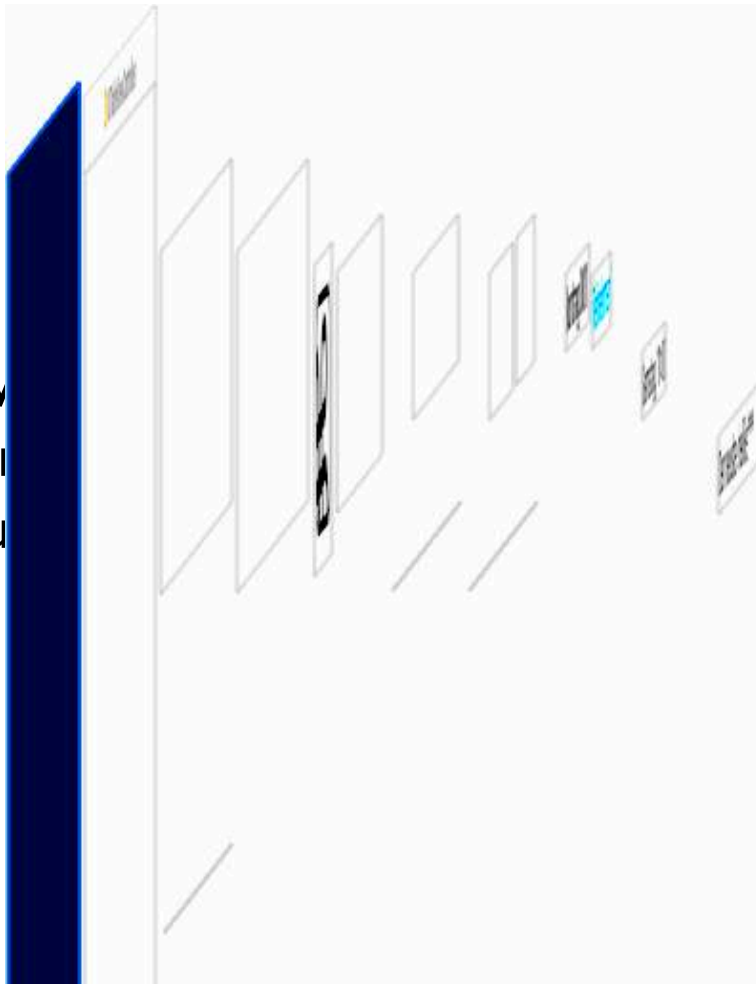
Transparenz kostet

- Die View-Hierarchie
 - ok-ish...
- Alle Backgrounds sind
 - .clear



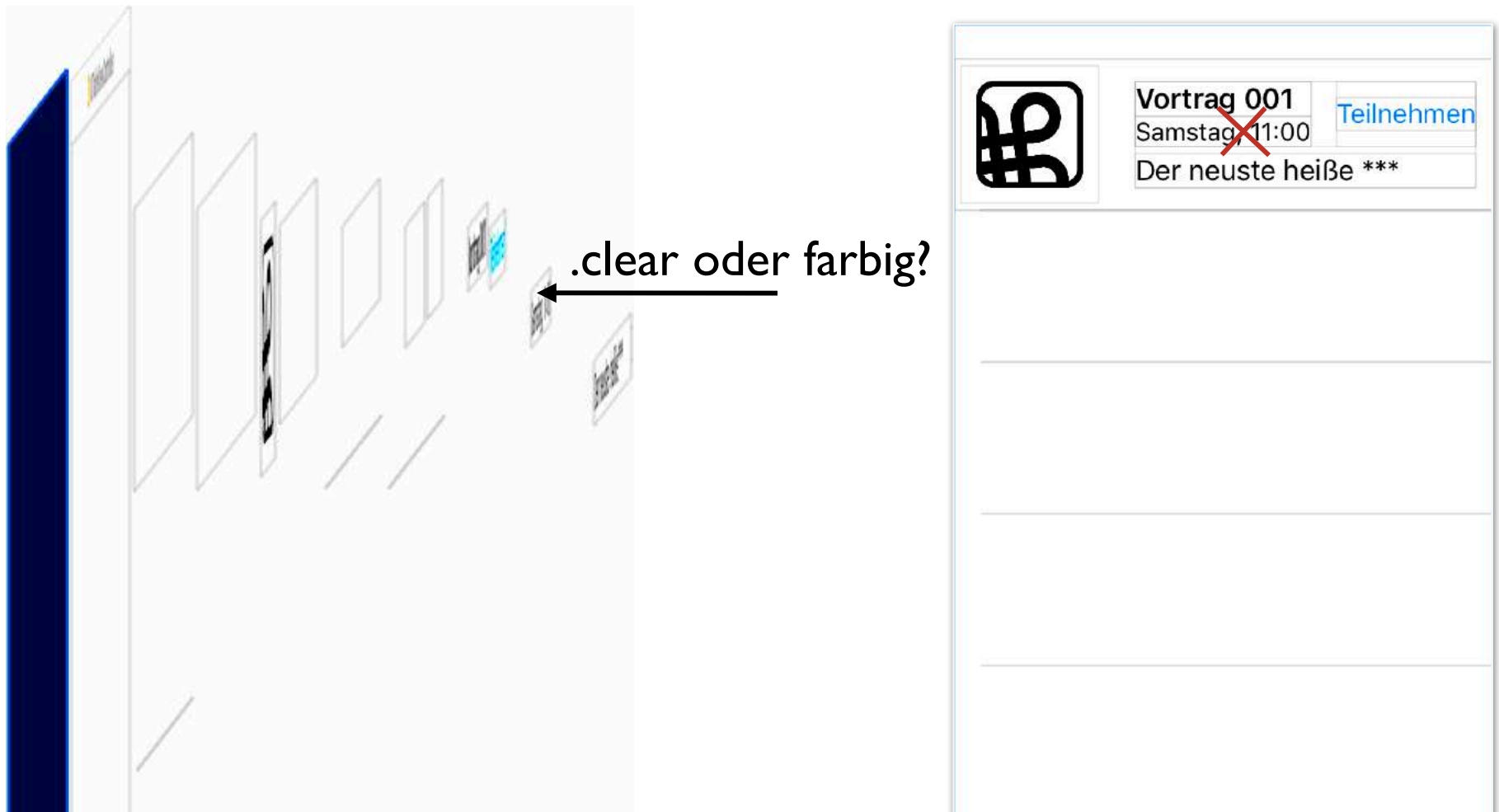
Transparenz kostet

- Wenn ein zu

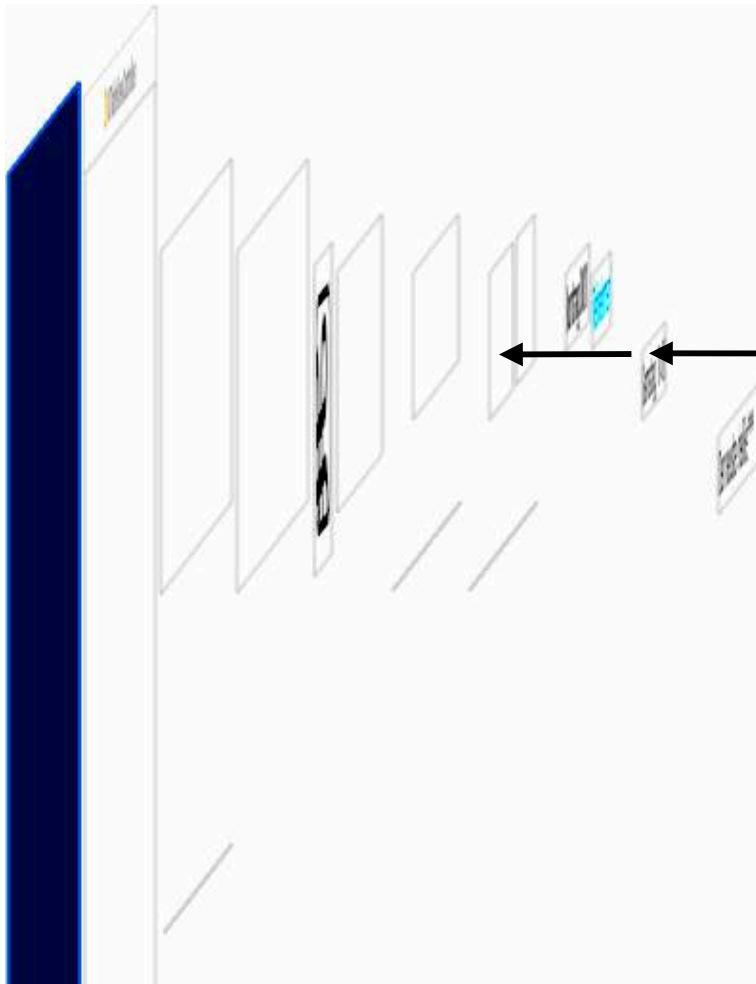


	Vortrag 001	Teilnehmen
	Samstag 11:00	
	Der neuste heiße ***	
<hr/>		
<hr/>		
<hr/>		
<hr/>		

Transparenz kostet

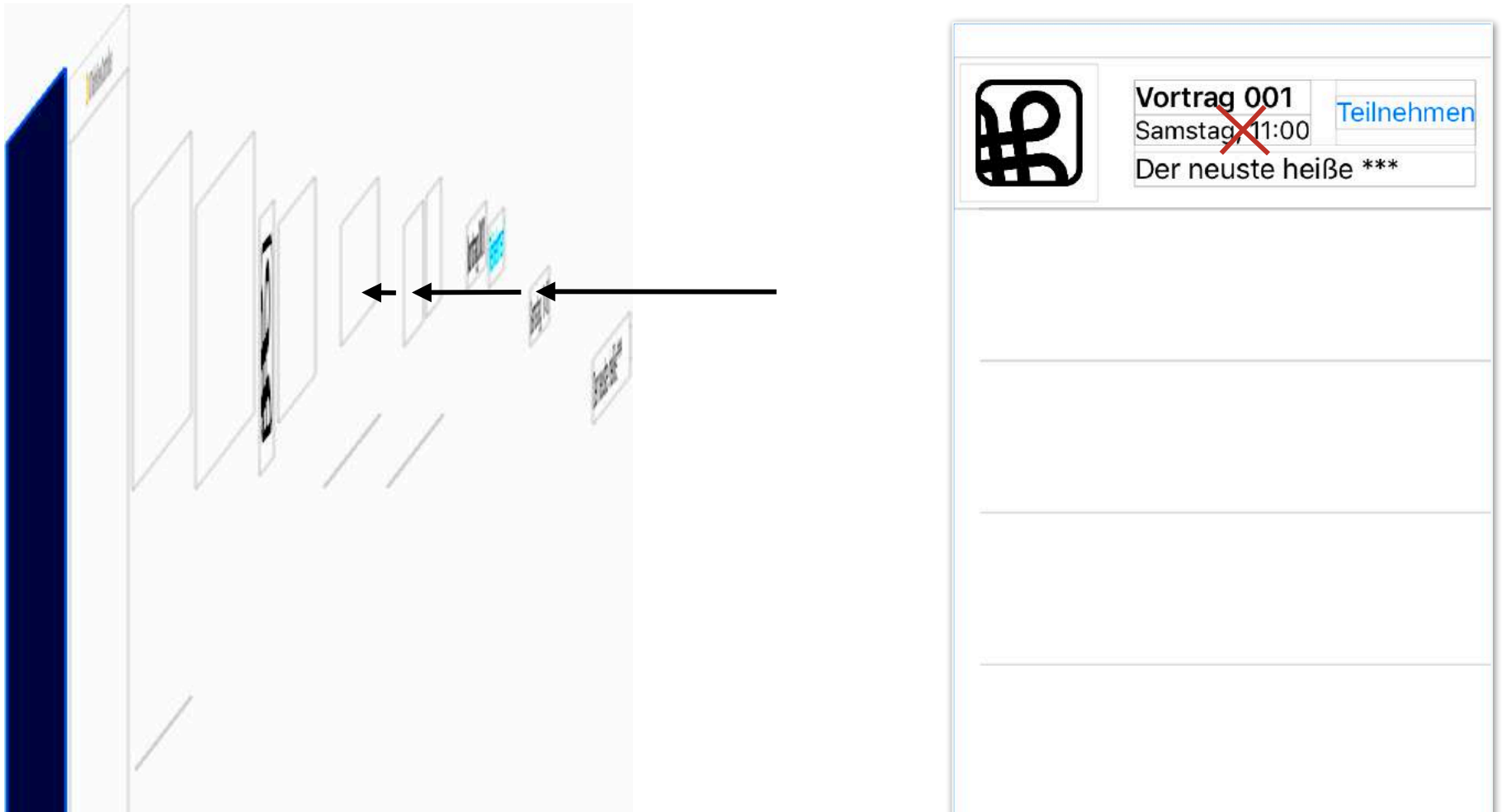


Transparenz kostet

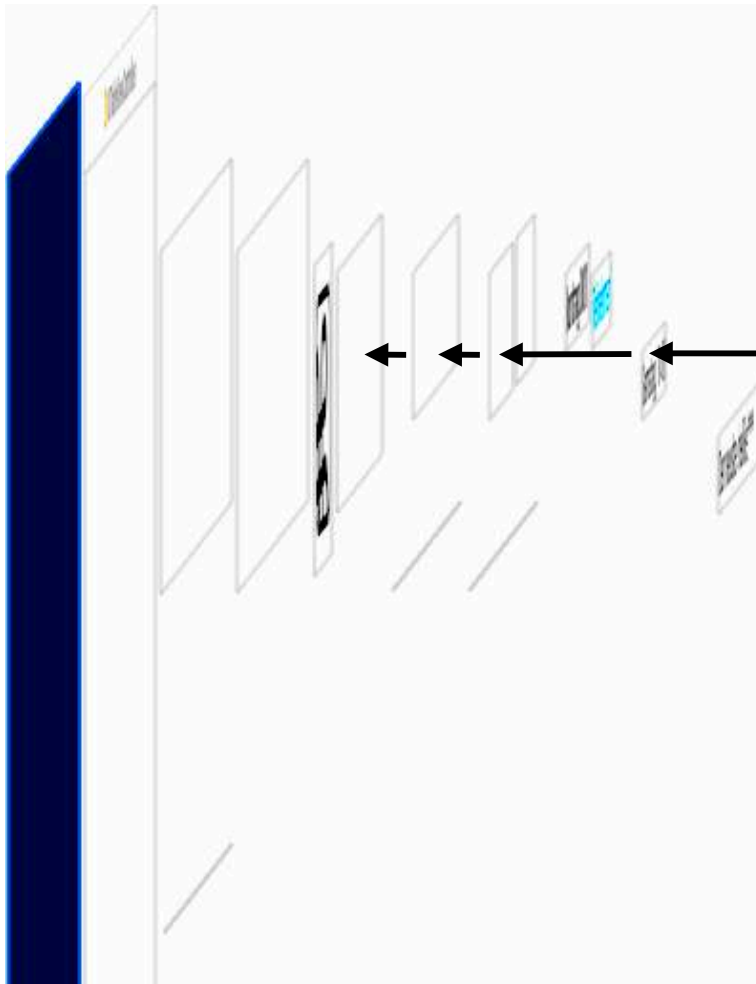


	Vortrag 001	Teilnehmen
	Samstag 11:00	
	Der neuste heiße ***	
<hr/>		
<hr/>		
<hr/>		
<hr/>		

Transparenz kostet

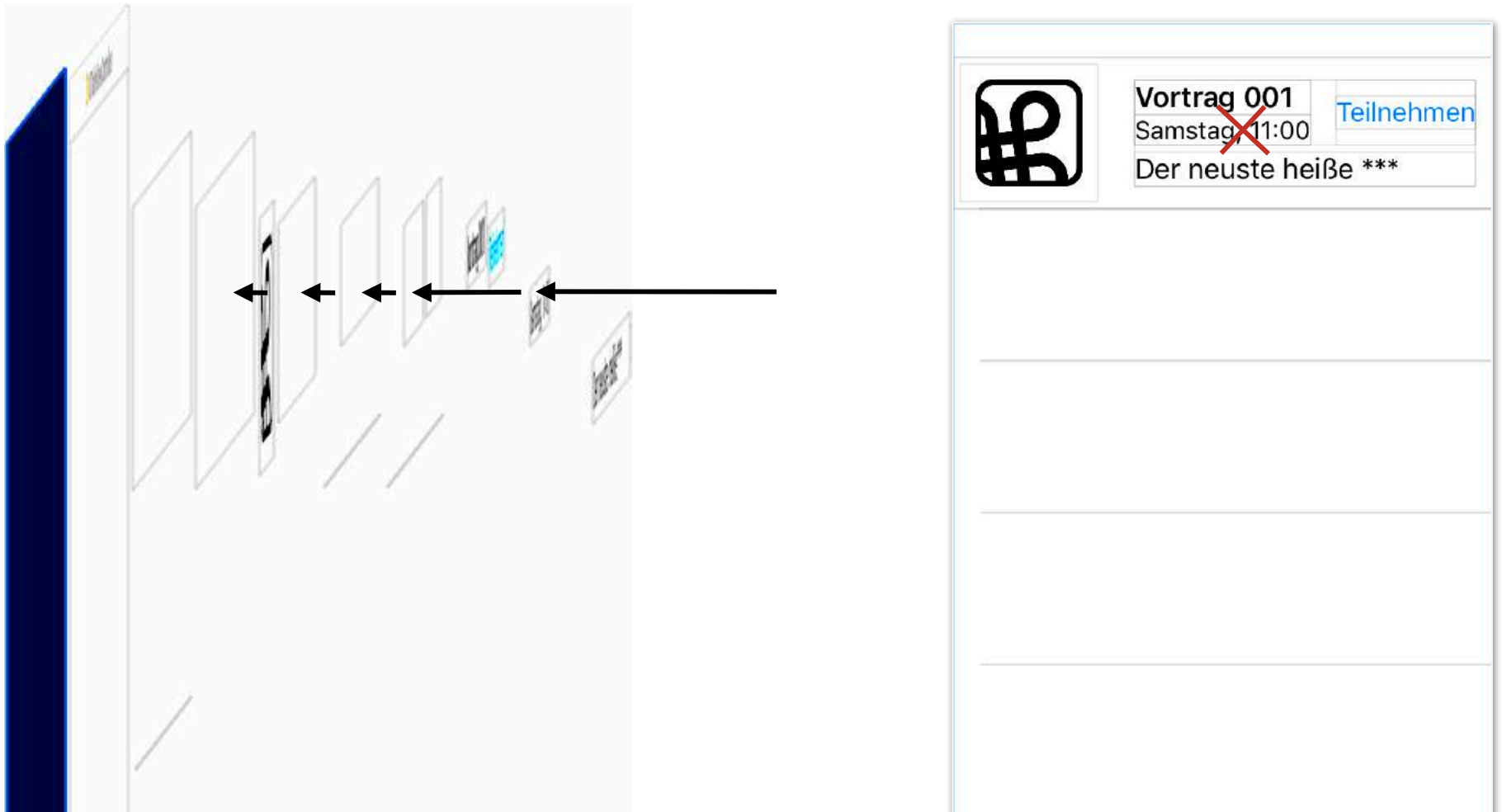


Transparenz kostet

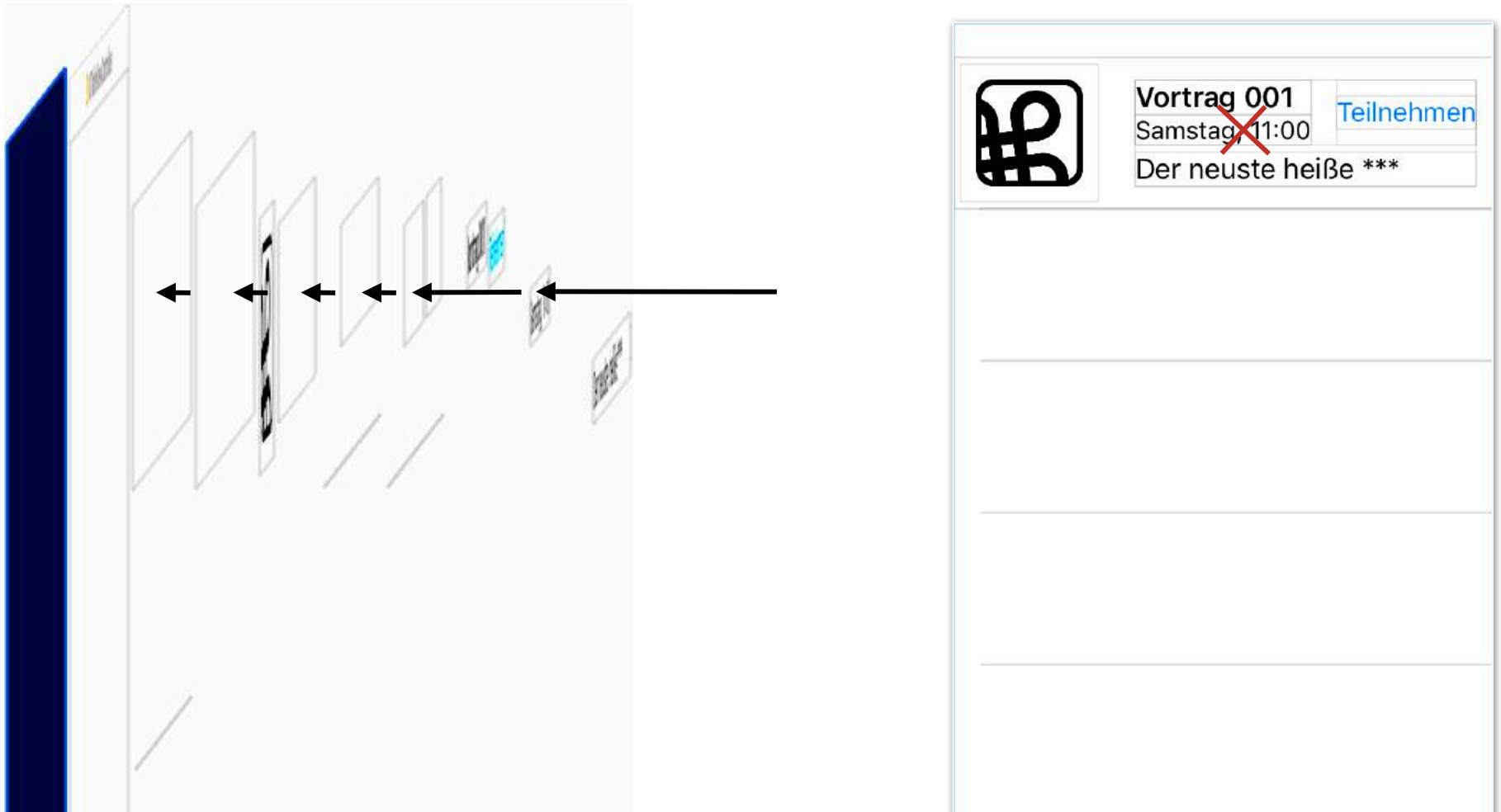


	Vortrag 001	Teilnehmen
	Samstag 11:00	
	Der neuste heiße ***	

Transparenz kostet

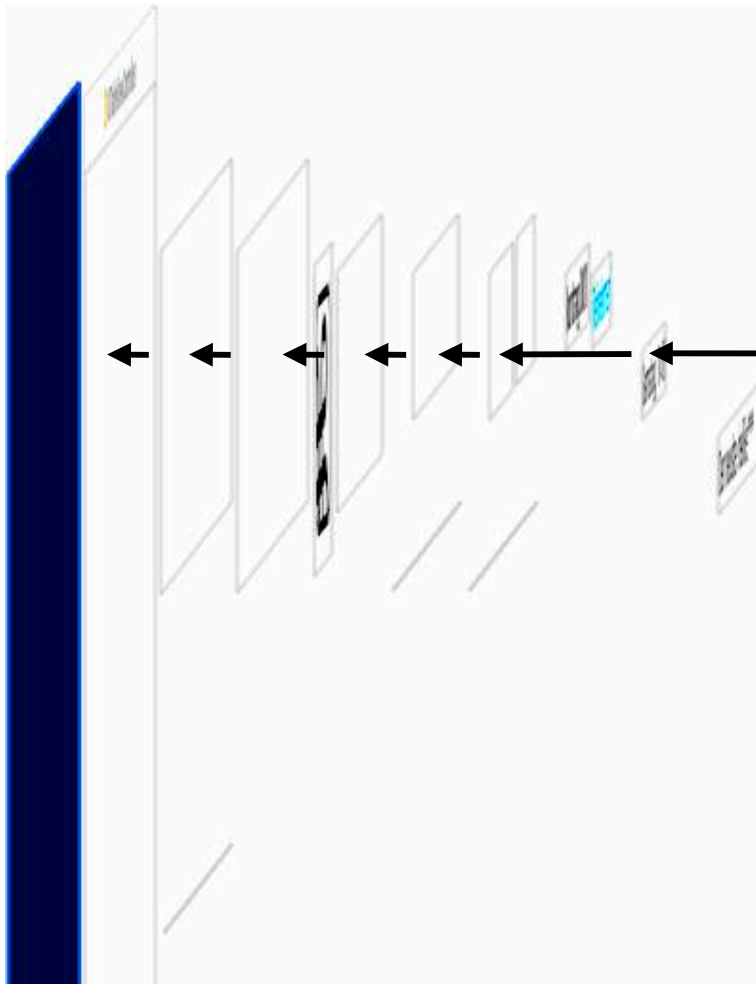


Transparenz kostet



Transparenz kostet

Endlich!

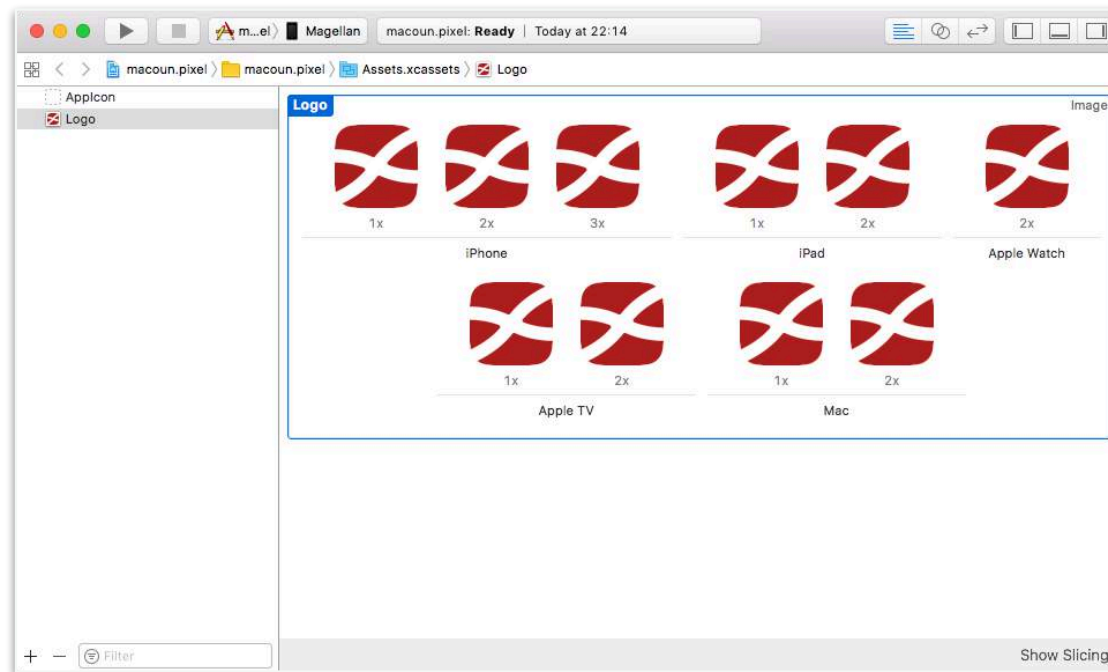


	Vortrag 001	Teilnehmen
	Samstag 11:00	
	Der neuste heiße ***	
<hr/>		
<hr/>		
<hr/>		
<hr/>		

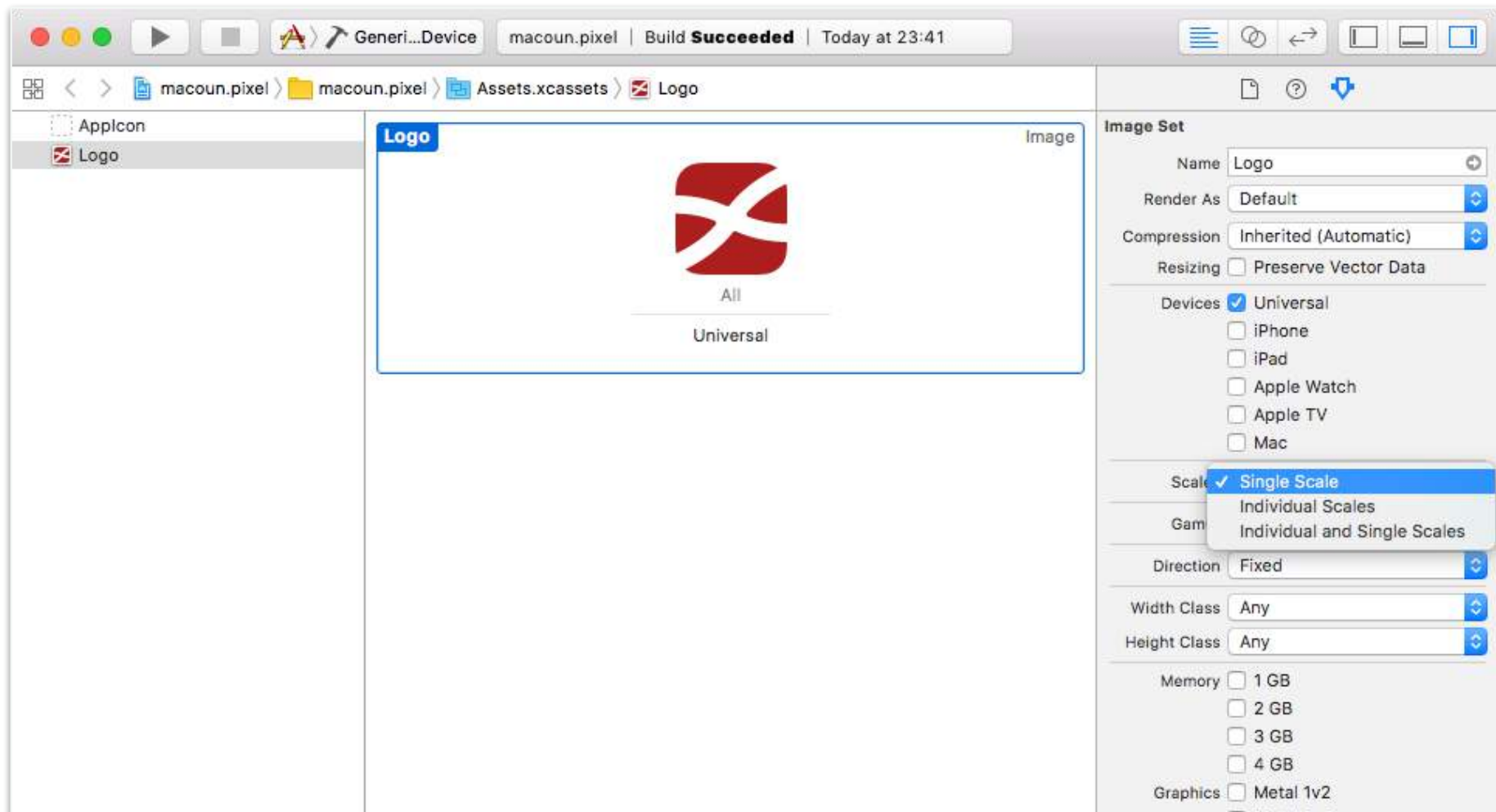
Gute Gründe gegen Transparenz

- Transparente Flächen sind komplexer zu rendern
 - kosten mehr CPU/GPU, besonders im Rudel
- sind manchmal auch gar nicht nötig
 - Label auf vollständig weißem Hintergrund
 - ➡ Hintergrundfarbe ebenfalls auf Weiß
- manchmal das fehlende i-Tüpfelchen zu 60 Frames

Varianten-Explosion



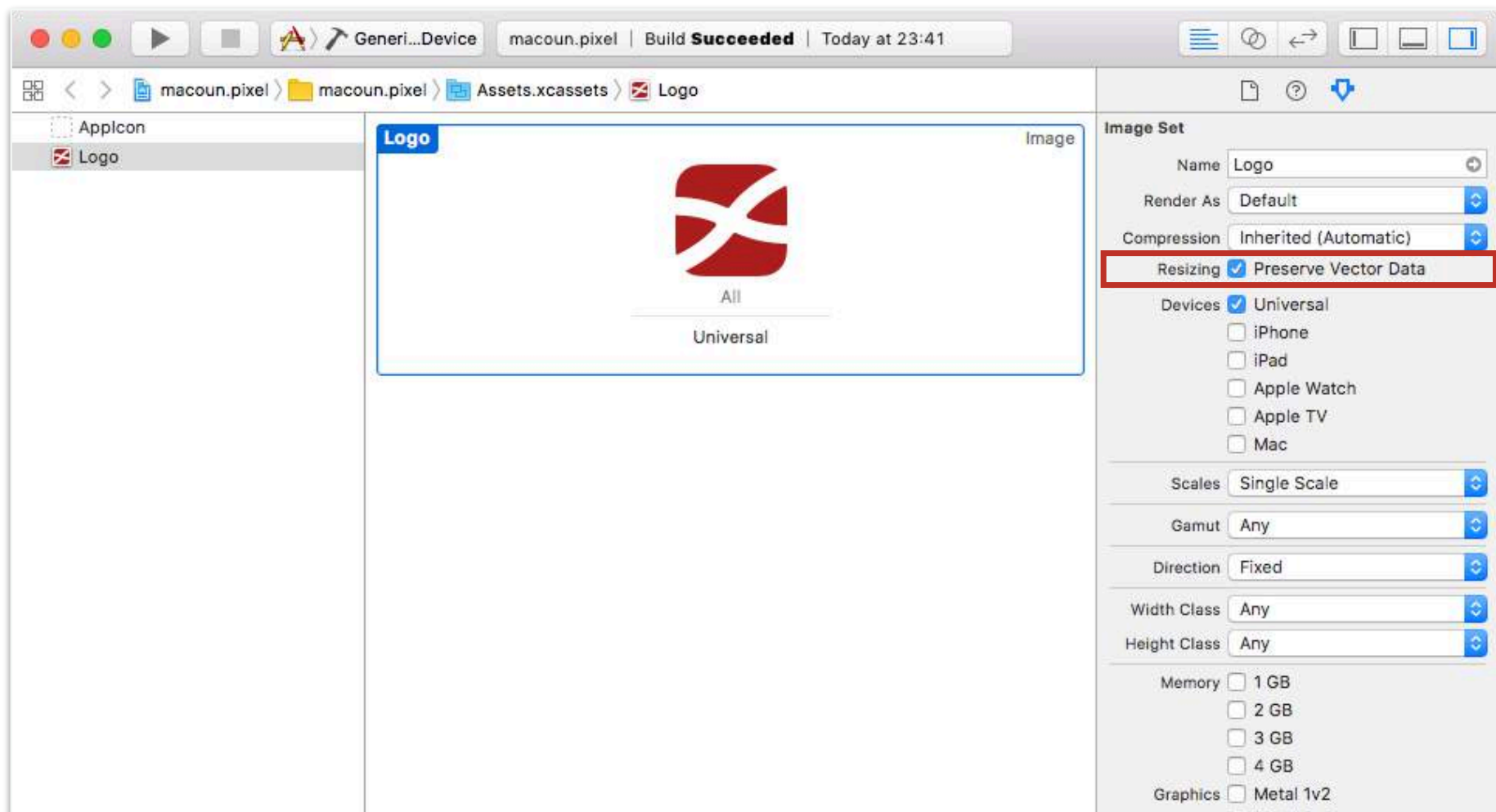
Das PDF-Zeitalter



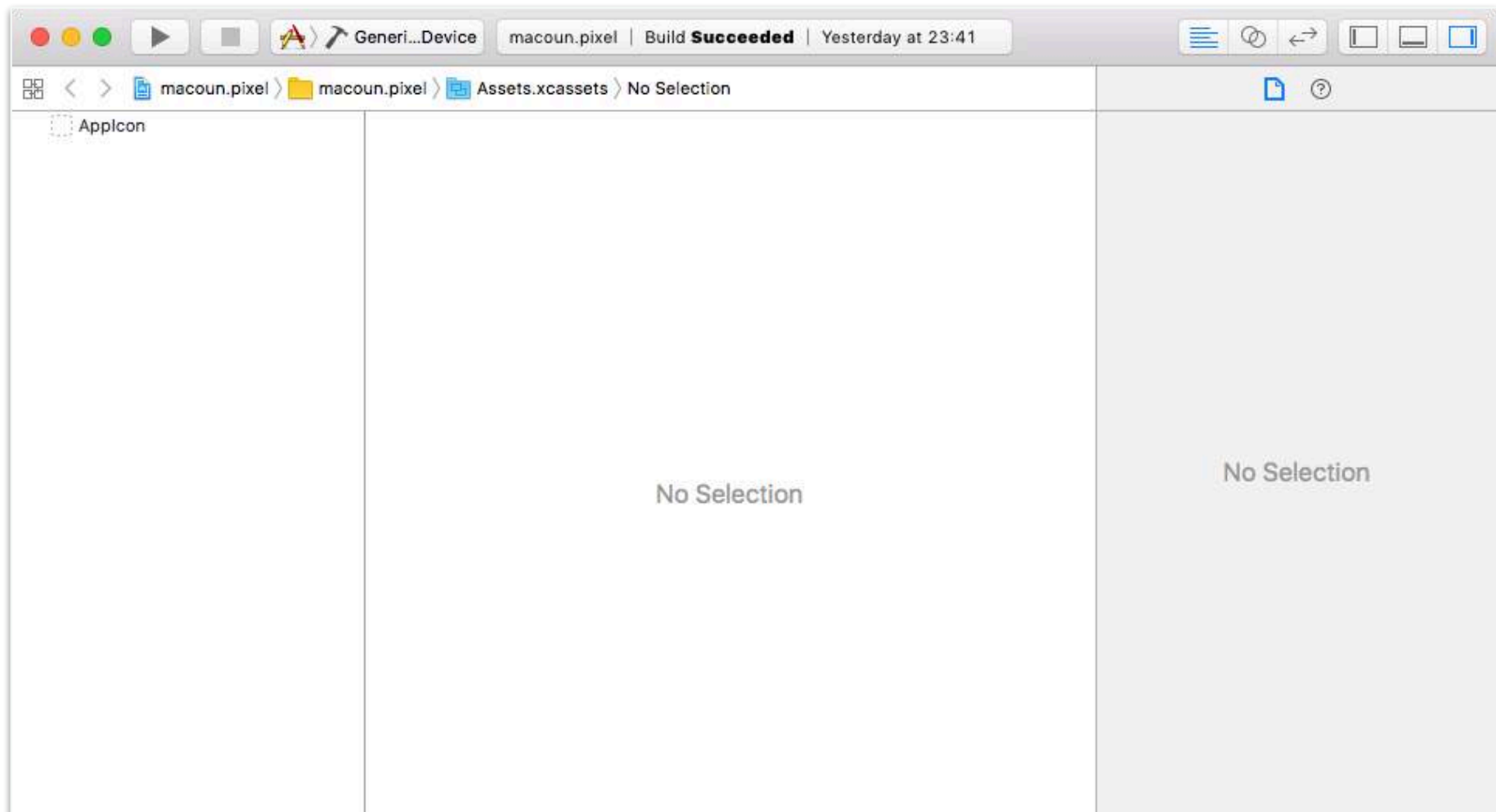
Das PDF-Zeitalter

- Eine PDF-Datei
 - meist kleiner als PNG
 - Design-Quelldatei und Asset in einem
 - automatisch versioniert
- Xcode macht den Rest

Xcode 9 - Neue Tricks



Das Post-PDF-Zeitalter

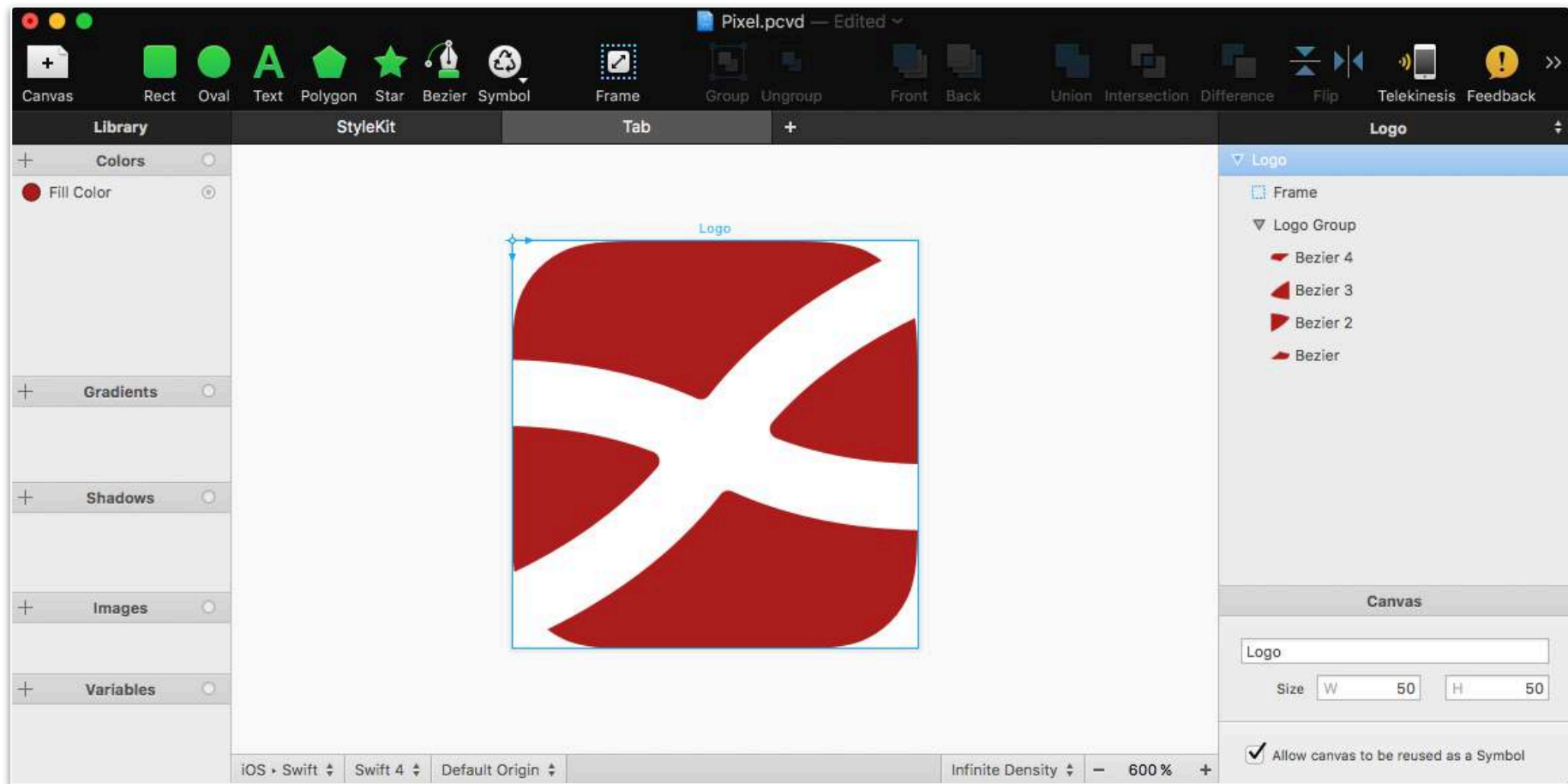


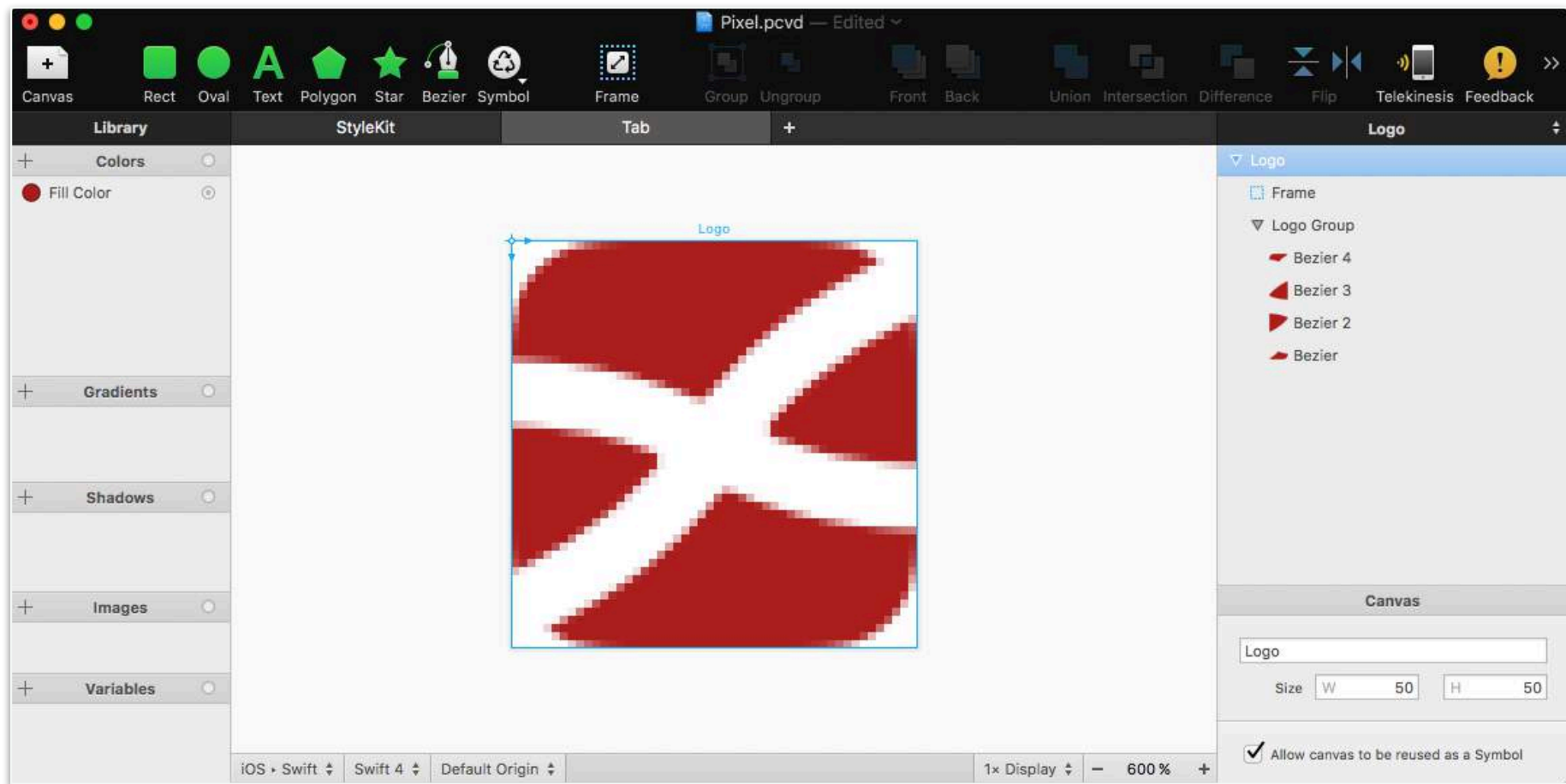
Das Post-PDF-Zeitalter

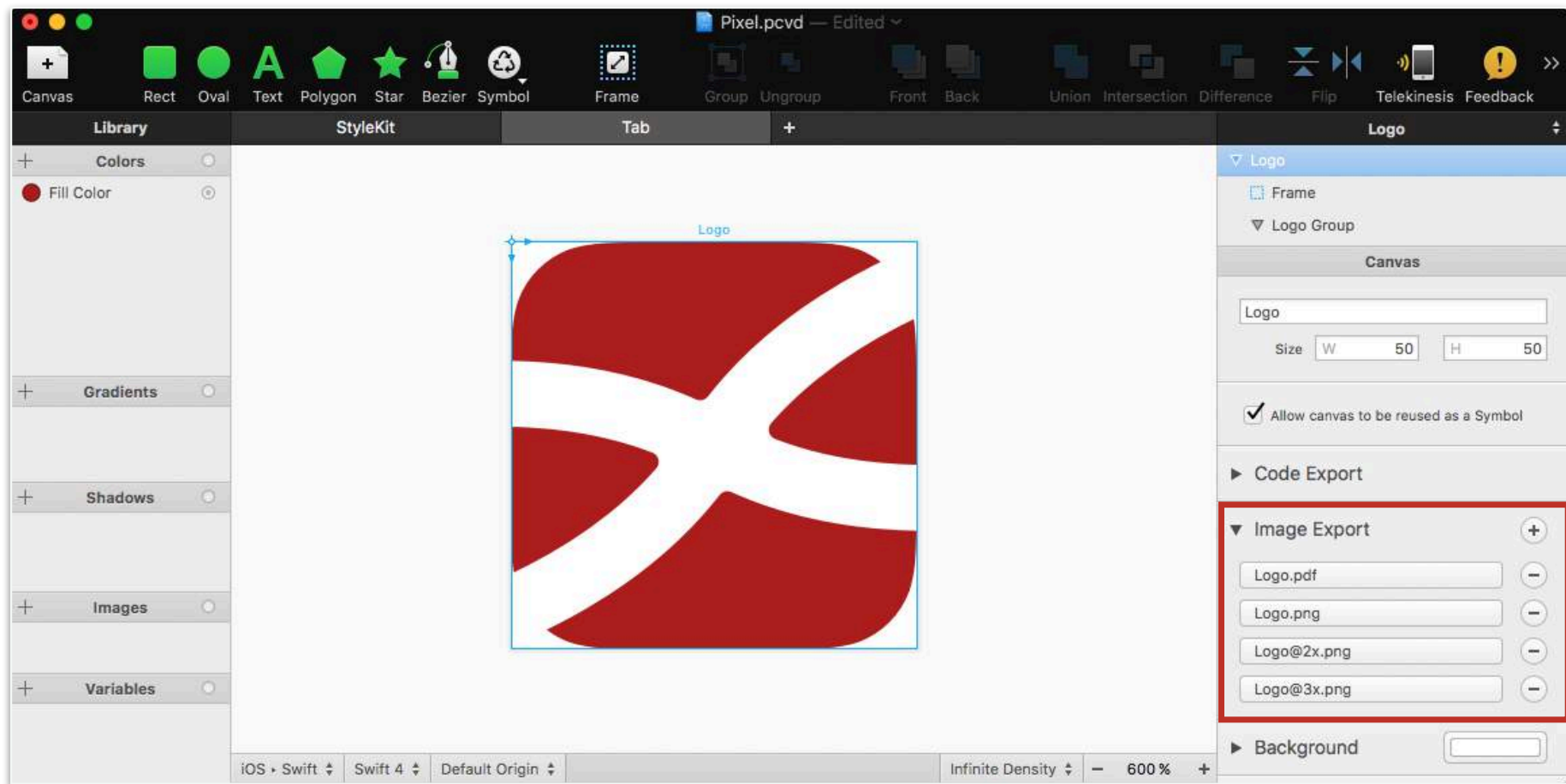
- Die meisten Grafiken sind sehr primitiv
 - Icons, Gradienten
- Man könnte die Formen auch einfach direkt zeichnen...

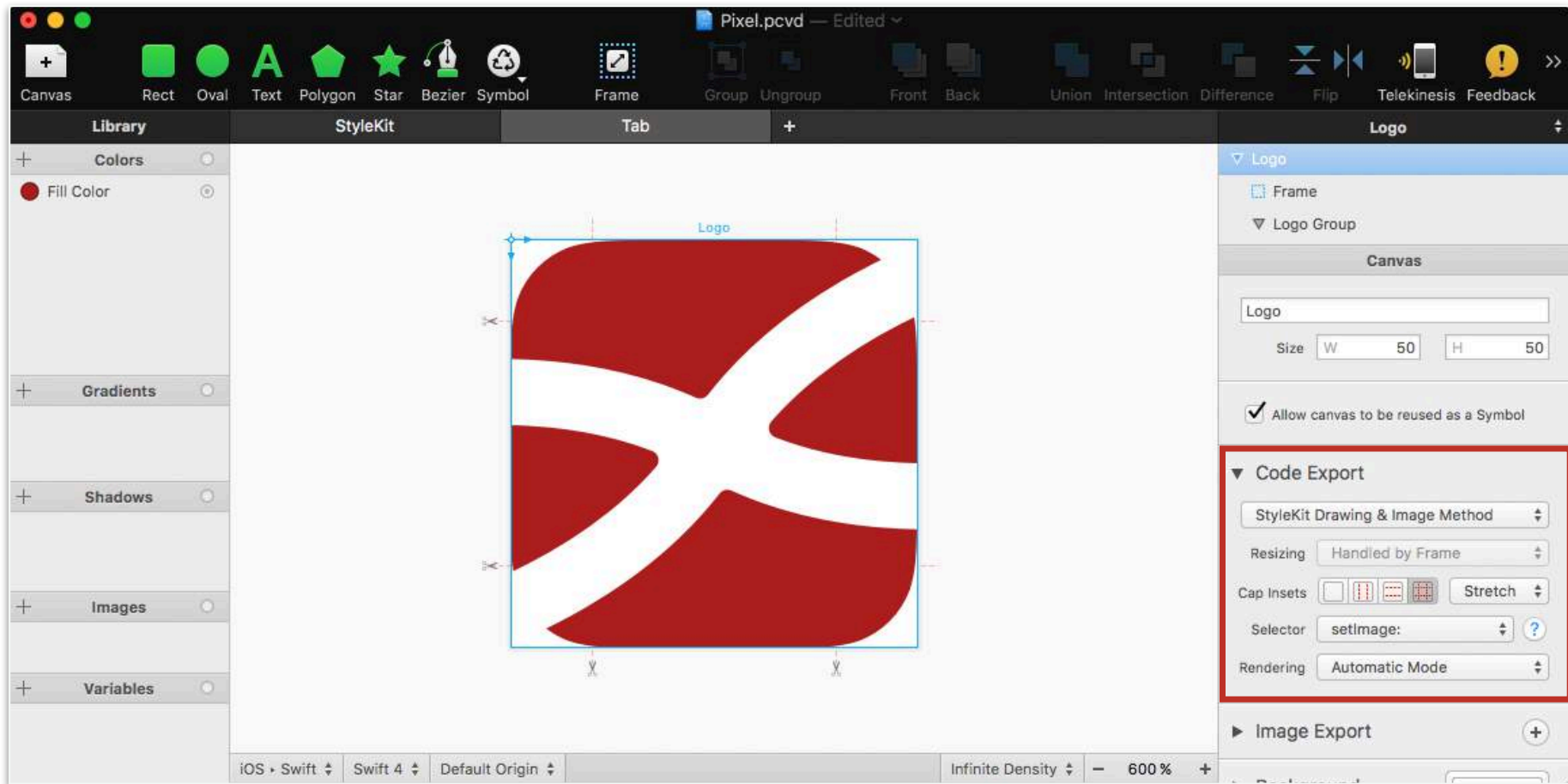
PaintCode

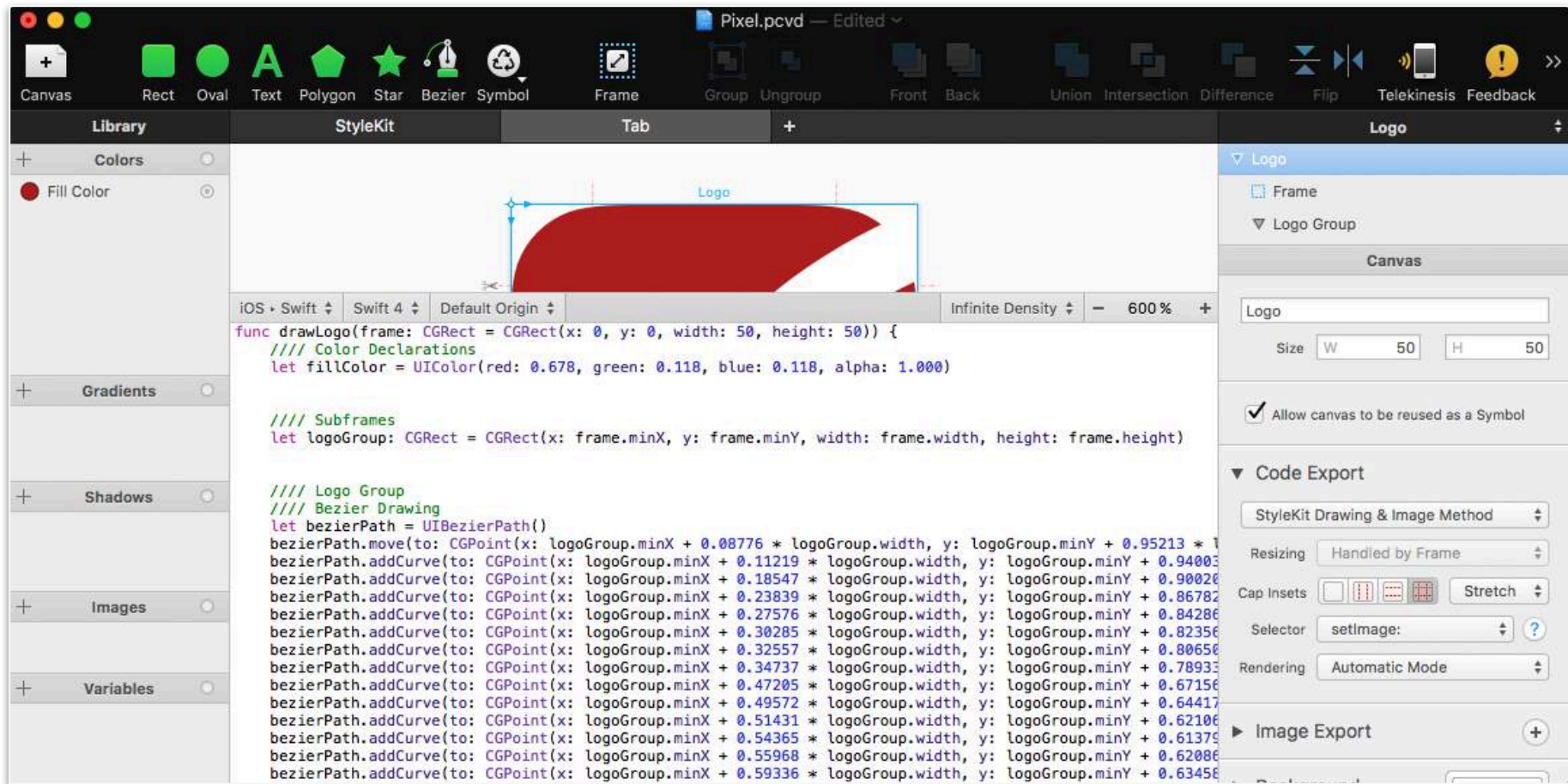
- So ne Art Sketch/Illustrator/Affinity Designer
 - nur eben für Entwickler gemacht
- Konvertiert Vektor-Grafiken in
 - Obj-C, Swift 3/4, C# Xamarin, Java, VectorDrawable, SVG, PNG, JPG, PDF, CSS
- Nebst einheitlichem StyleKit für Farben, Verläufe, ...



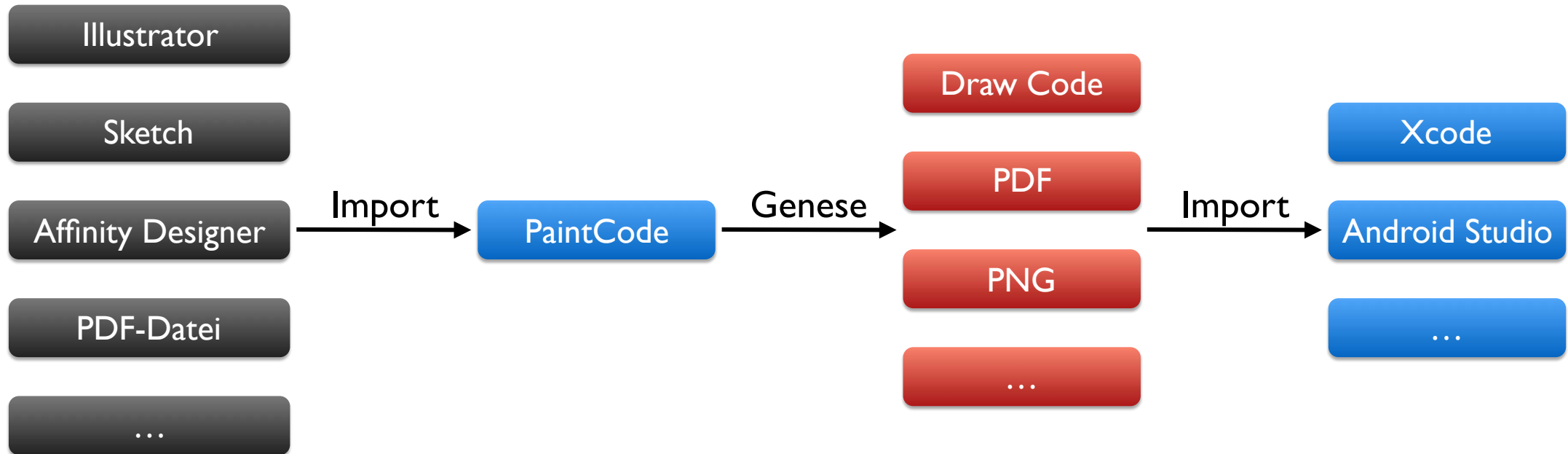


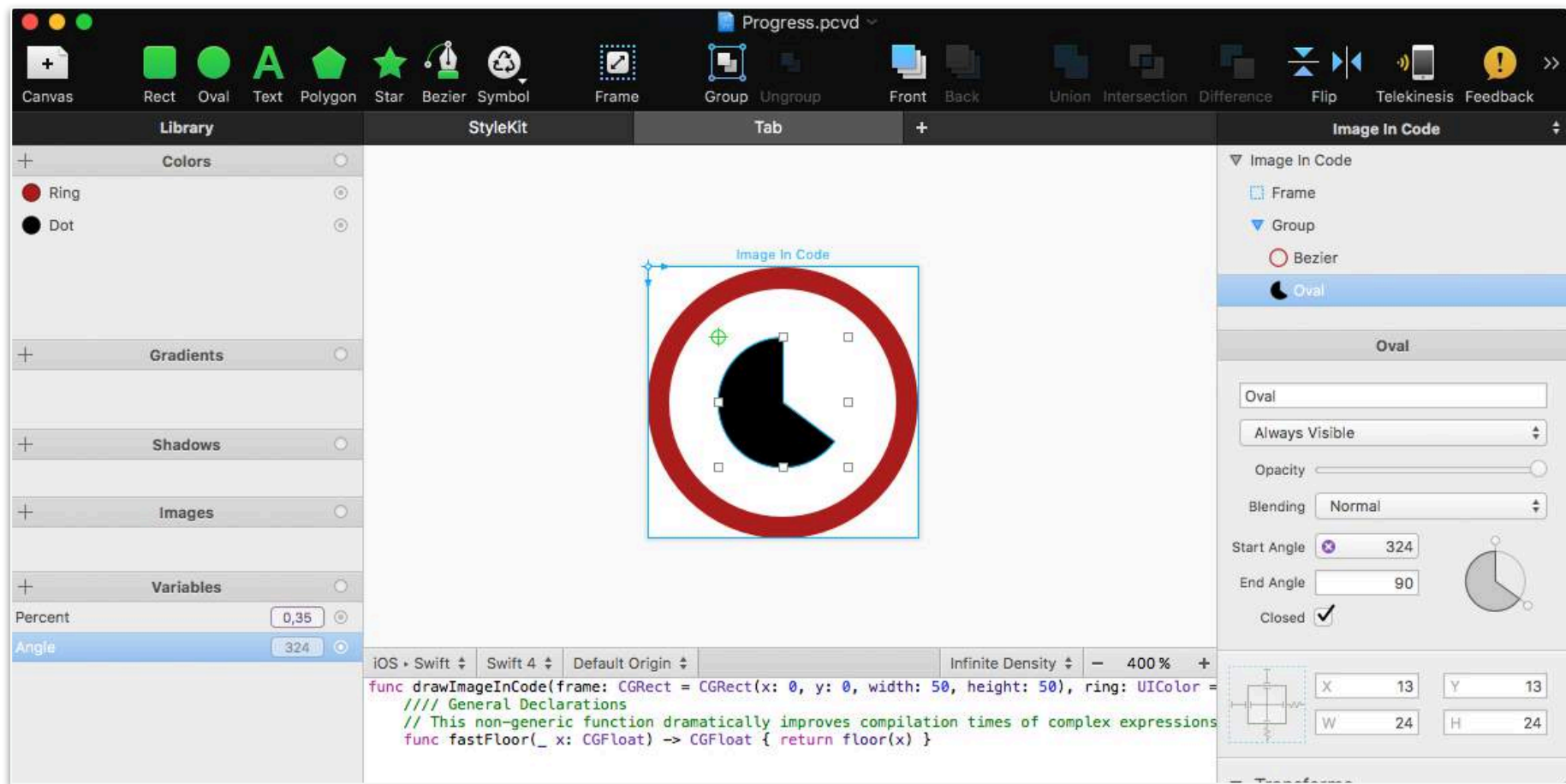






Daten Fluss



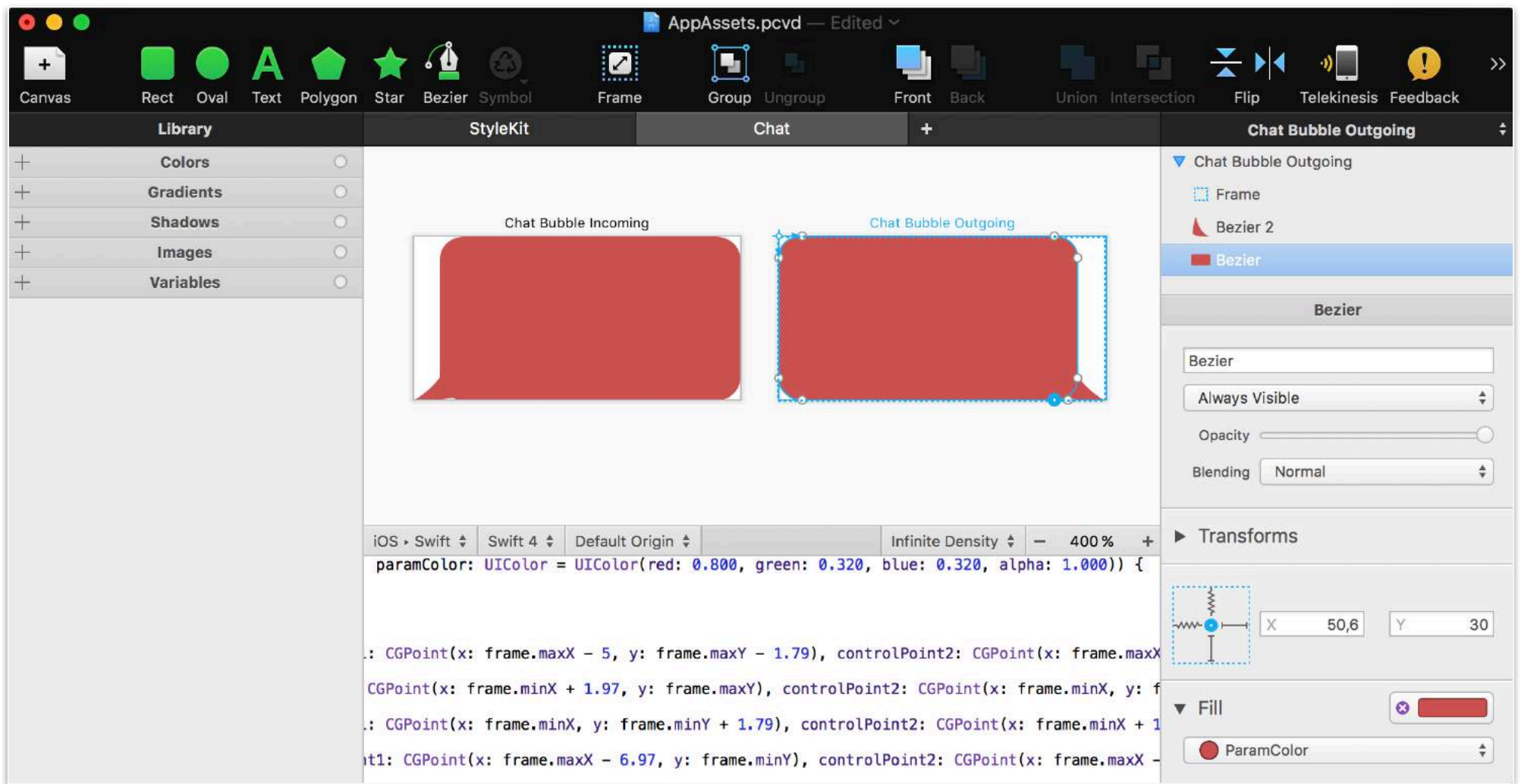


DrawCode

```
public class StyleKitName : NSObject {  
    @objc dynamic public class func  
        drawImageInCode(frame: CGRect = CGRect(...),  
                        ring: UIColor = UIColor(...),  
                        dot: UIColor = UIColor(...),  
                        percent: CGFloat = 0.35) {  
        ...  
    }  
}
```


DrawCode

```
extension UIImage {  
    static func codeImage(imageSize: CGSize, ...) -> UIImage? {  
        let imageRect = CGRect(x: 0, y: 0,  
                                width: imageSize.width,  
                                height: imageSize.height)  
        UIGraphicsBeginImageContextWithOptions(imageRect.size, false, 0)  
        drawImageInCode(frame: imageRect, ...)  
        let image = UIGraphicsGetImageFromCurrentImageContext()  
        UIGraphicsEndImageContext()  
        return image  
    }  
}
```



DrawCode

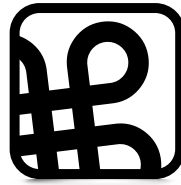
- Flexibel
 - Größe, Farbe, parametrisiertes Layout
- Unschlagbarer Platzverbrauch
- Kleine Helfer-Konstrukte nötig
- UIImage Erzeugung
 - oder auch direkt in `draw(_ rect: CGRect)`
- Merging von PaintCode Dateien allerdings die Hölle

Zusammenfassung

- Kleine Stolpersteine auf dem Weg zur sauberen UI
- Alternativen zu Pixel-Bilder-Stacks
- Oder ganz ohne Bilddateien

Fragen?

Vielen Dank



Macoun