



Автоматизация ревью ресурсных картинок в iOS: ImageLinter

Сергей Балалаев

Руководитель отдела разработки мобильного приложения “ПВЗ”

21 февраля 2023



Сергей Балалаев

Руководитель отдела разработки
мобильных приложений «ПВЗ»

Карьера:

- **14 лет** опыта в мобильной разработке
- **9 лет** руковожу людьми
- **8 лет** преподаю курсы iOS



Сергей Балалаев

Руководитель отдела разработки
мобильных приложений «ПВЗ»

Задачи в Ozon:

- Координация и планирование
- Формирование стека технологий
- Построение архитектуры
- Постановка технических задач
- iOS комитет
- Route 256
- Написание статей на Habr
- Выступления на митапах

План

Маршрут презентации

1. Проблемы с изображениями
2. Викторина
3. Причины проблем
4. Инструменты и решения
 - Figma: экспорт
 - Кодогенерация: R.swift, SwiftGen
 - Snapshot-тесты
5. Приложение ПВЗ
 - Imagerlinter
6. Заключение



1

Проблемы с изображениями в iOS



Проблемы на типичном проекте iOS

1. **Отсутствие изображения**
удалили/переименовали/описались
2. **Дубликаты названий**
Assets VS just file
3. **Ошибки разрешения**
путаницы 1x, 2x, 3x
4. **Растр вместо вектора**
контент PDF и SVG
5. **Мусор: не используемые и дубли**
удалили код/добавили идентичную картинку
6. **Ошибки размера на пиксели**
нечетные пиксели для четного разрешения



Image not found!

Викторина

Попробуем разобраться в том, как работают приложения, если разработчик косячит с картинками в ресурсах приложения

3

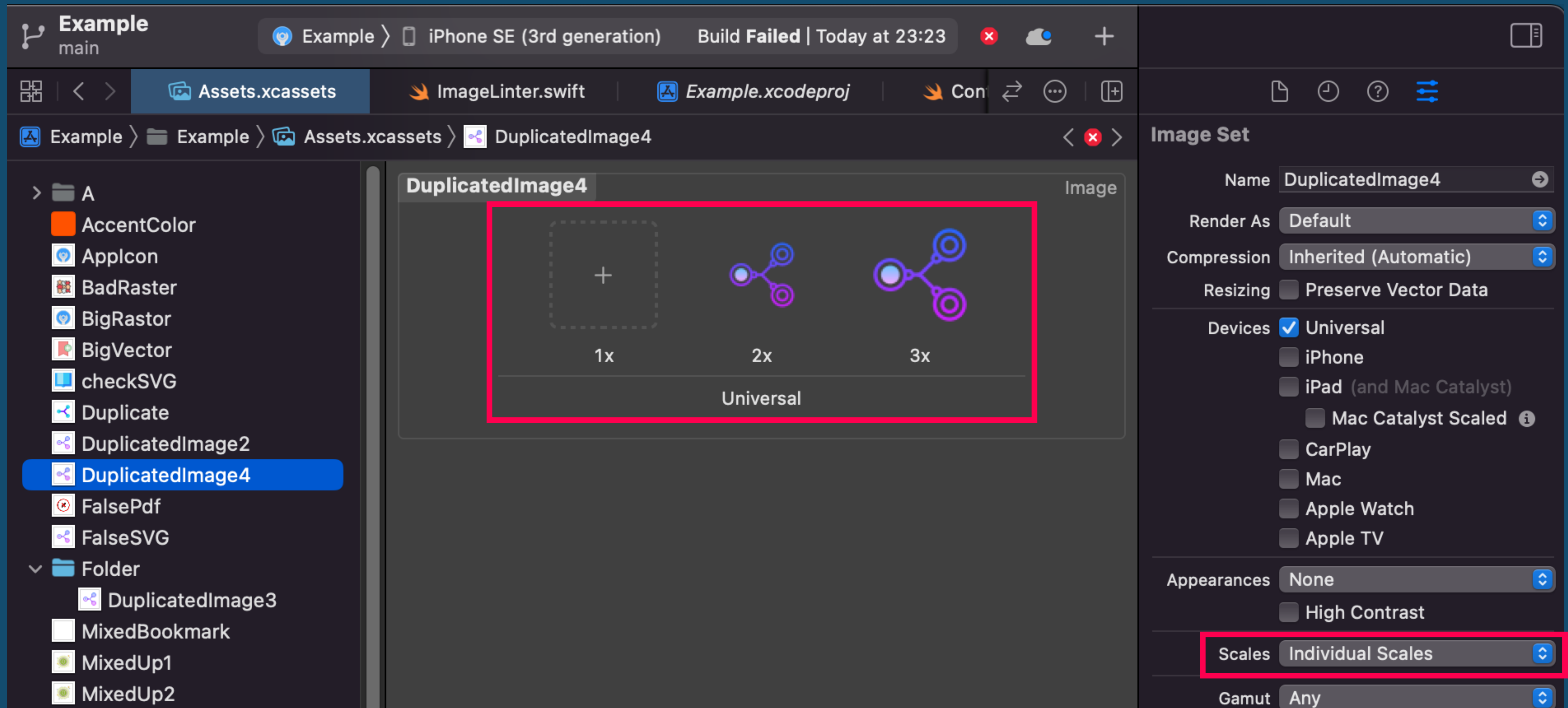
Причины проблем



Assets

Растровые изображения

Assets



Векторные изображения

Assets

The screenshot displays the Xcode interface for editing an image set. The top status bar shows the project name 'Example', the target device 'iPhone SE (3rd generation)', and a 'Build Failed' message. The breadcrumb navigation indicates the current location is 'Assets.xcassets > FalseSVG'. The left sidebar shows a list of assets, with 'FalseSVG' selected. The main canvas shows the 'FalseSVG' image set, which is a blue and purple logo consisting of three circles connected by lines. Below the image, the 'All' and 'Universal' scales are visible. The right sidebar shows the 'Image Set' settings, including 'Name' (FalseSVG), 'Render As' (Default), 'Compression' (Inherited (Automatic)), and 'Resizing' (Preserve Vector Data). The 'Devices' section is expanded, showing 'Universal' selected. The 'Scales' dropdown menu is highlighted with a red box, showing 'Single Scale' selected. The 'Gamut' is set to 'Any'.

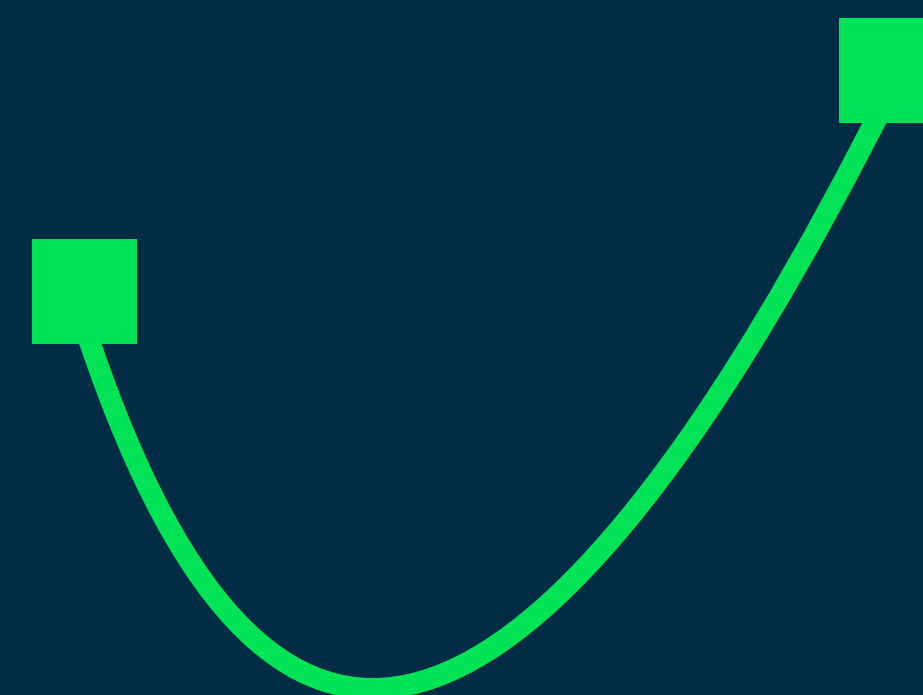
Поддержка форматов

Assets



Растр:

HEIF, HEIC, PNG, JPEG



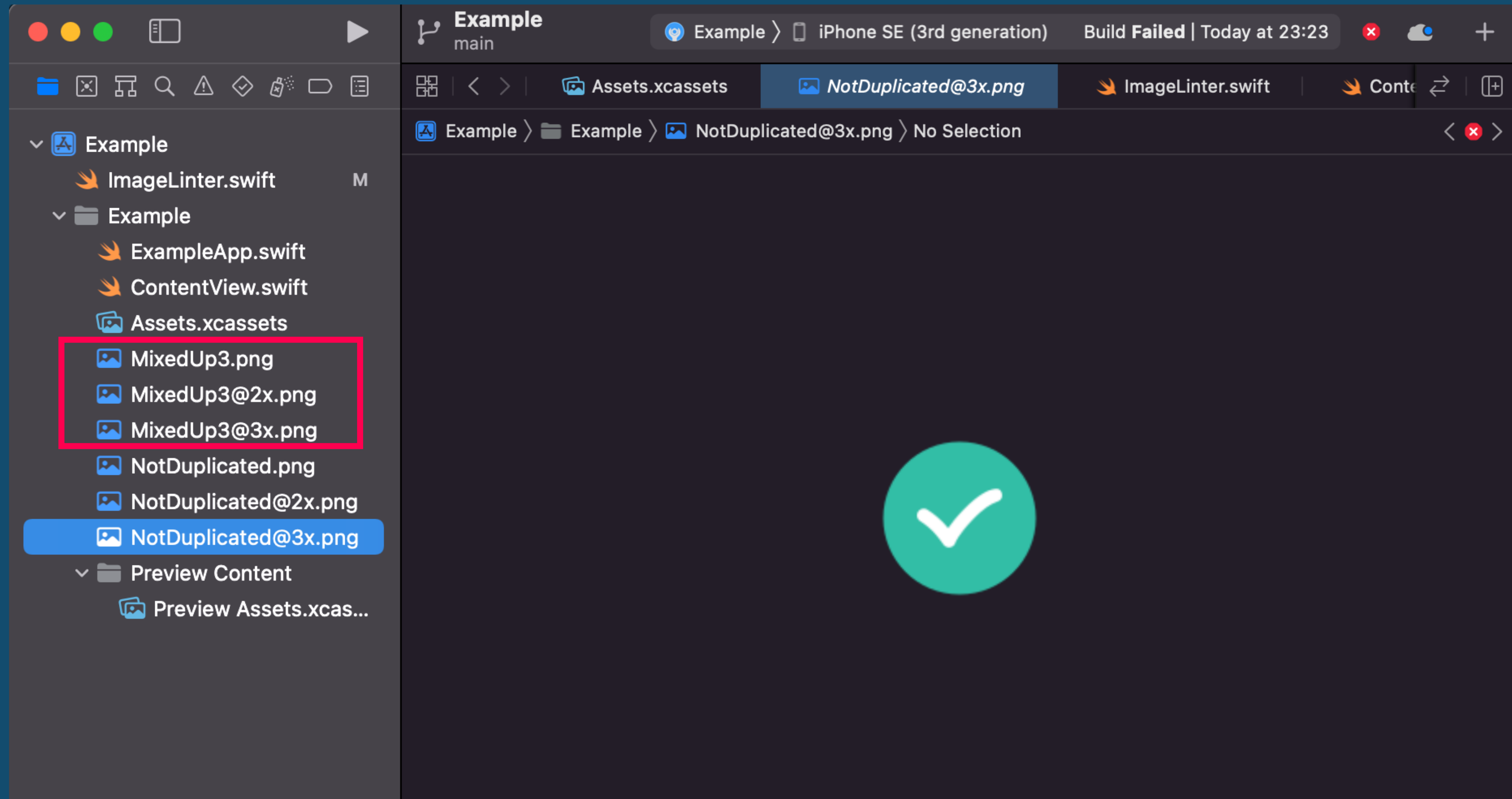
Вектор:

PDF, SVG

Just file

Хранение изображений в виде файлов

Just file

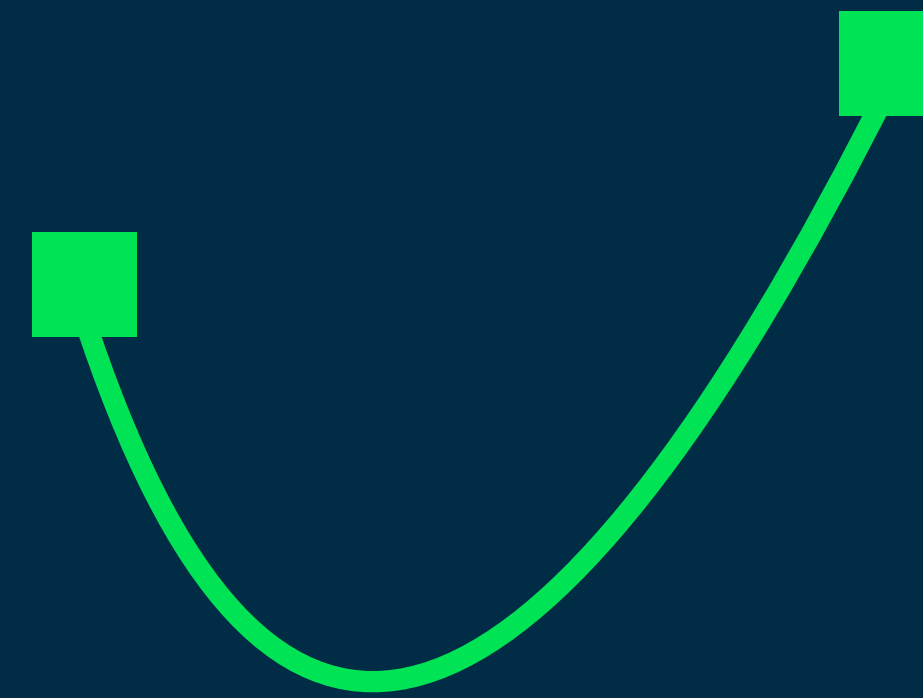


Поддержка форматов

Just file



Растр:
PNG, JPEG

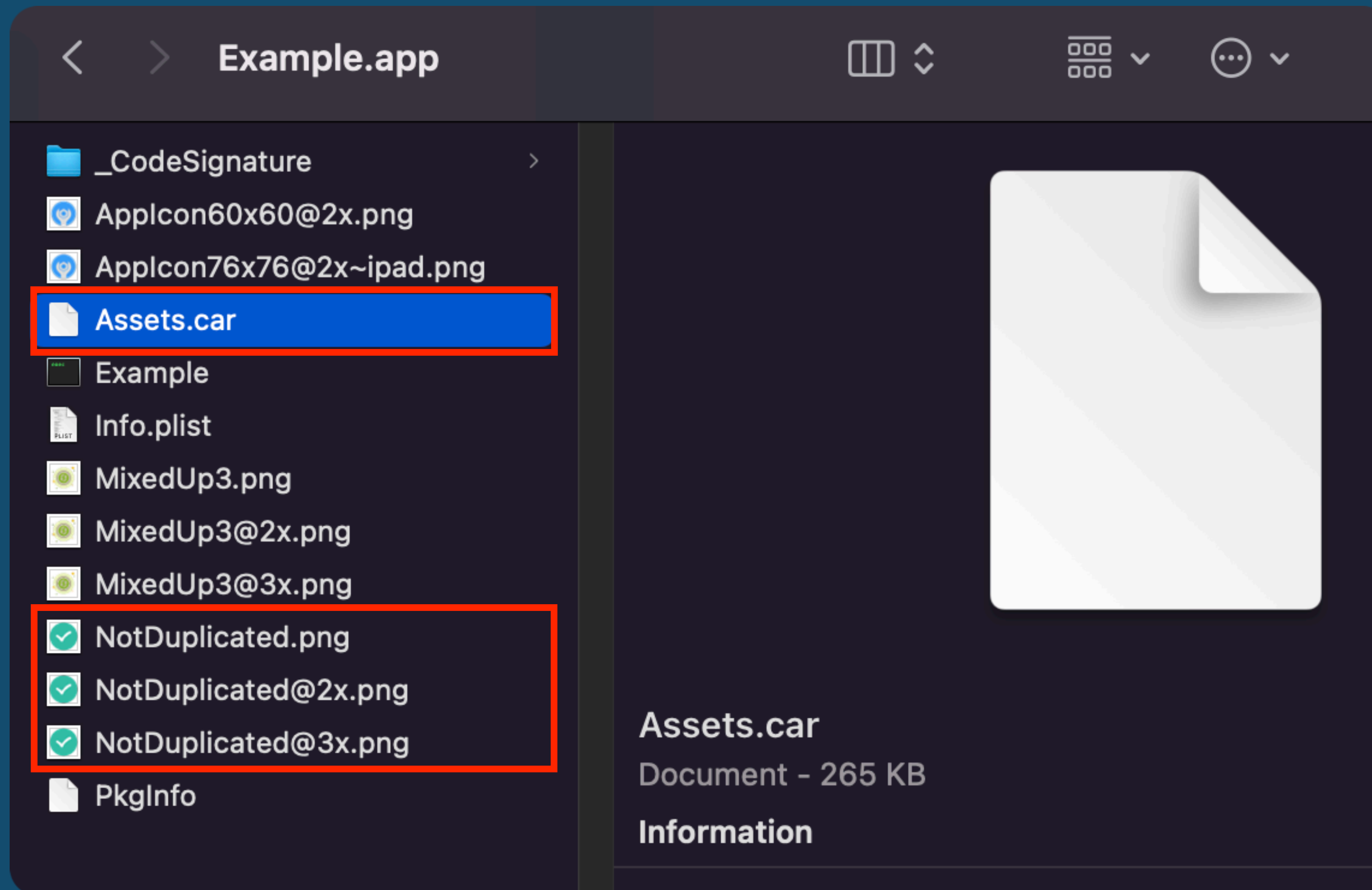


Вектор:
PDF

Компиляция

Добавляем картинку

Xcode



Архив Assets.car можно распаковать

Xcode



iOS Asset Extractor

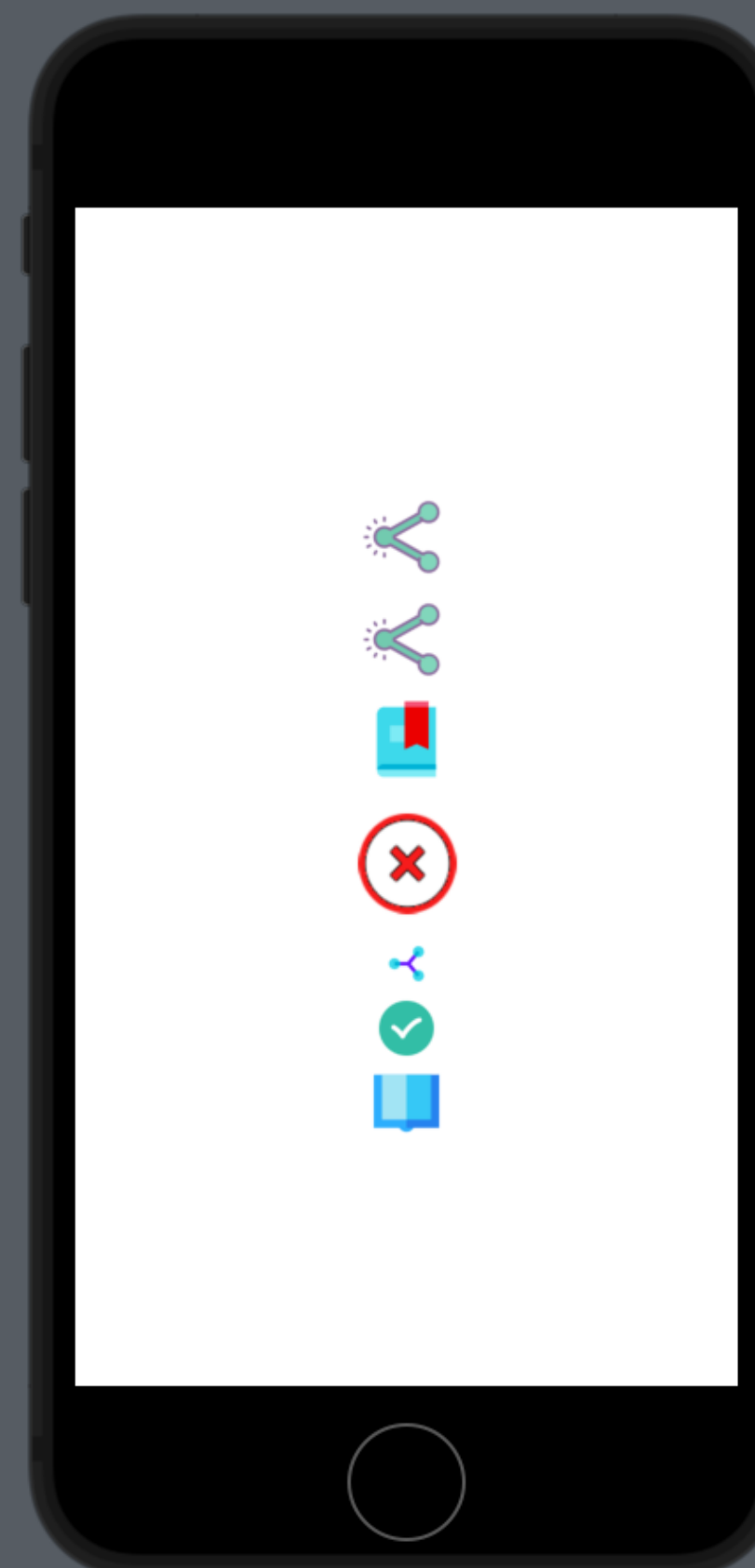
A tool to extract images from the iOS SDK.

Использование в коде

Отображаем изображения через ресурсные названия

XCode

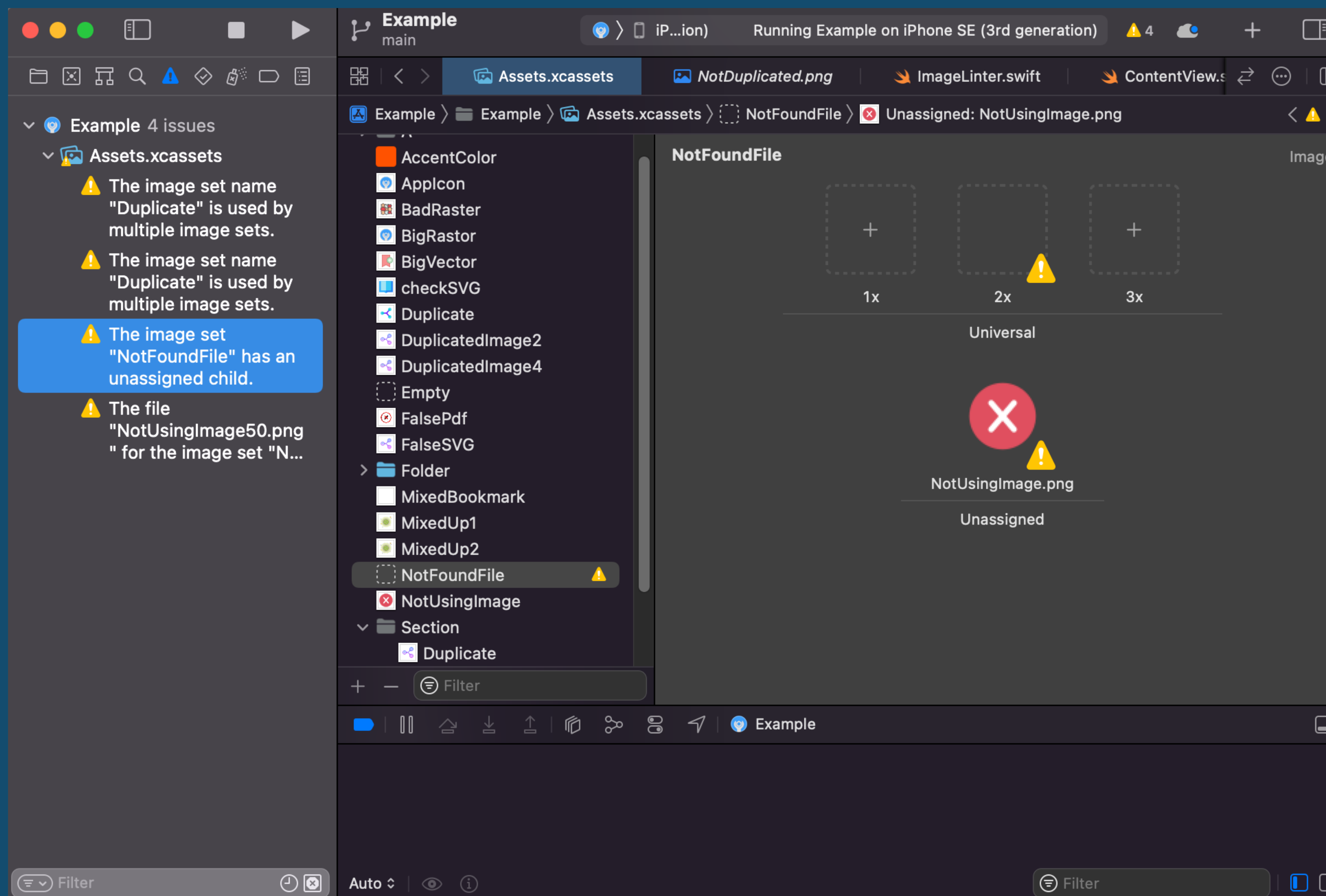
```
5 // Created by Sergey Balalaev on 27.09.2022.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         VStack {
13             Image("TruePng")
14                 .imageScale(.large)
15                 .foregroundColor(.accentColor)
16             Image("TruePng")
17             Image("TruePdf")
18             Image("FalsePdf")
19             Image("Duplicate")
20
21             if let image = UIImage(named:
22                 "NotDuplicated") {
23                 Image(uiImage: image)
24             }
25
26             Image("checkSVG")
27         }
28         .padding()
29     }
30
31 struct ContentView_Previews: PreviewProvider {
32     static var previews: some View {
33         ContentView()
34     }
35 }
36
```



Как Xcode заботится о нас

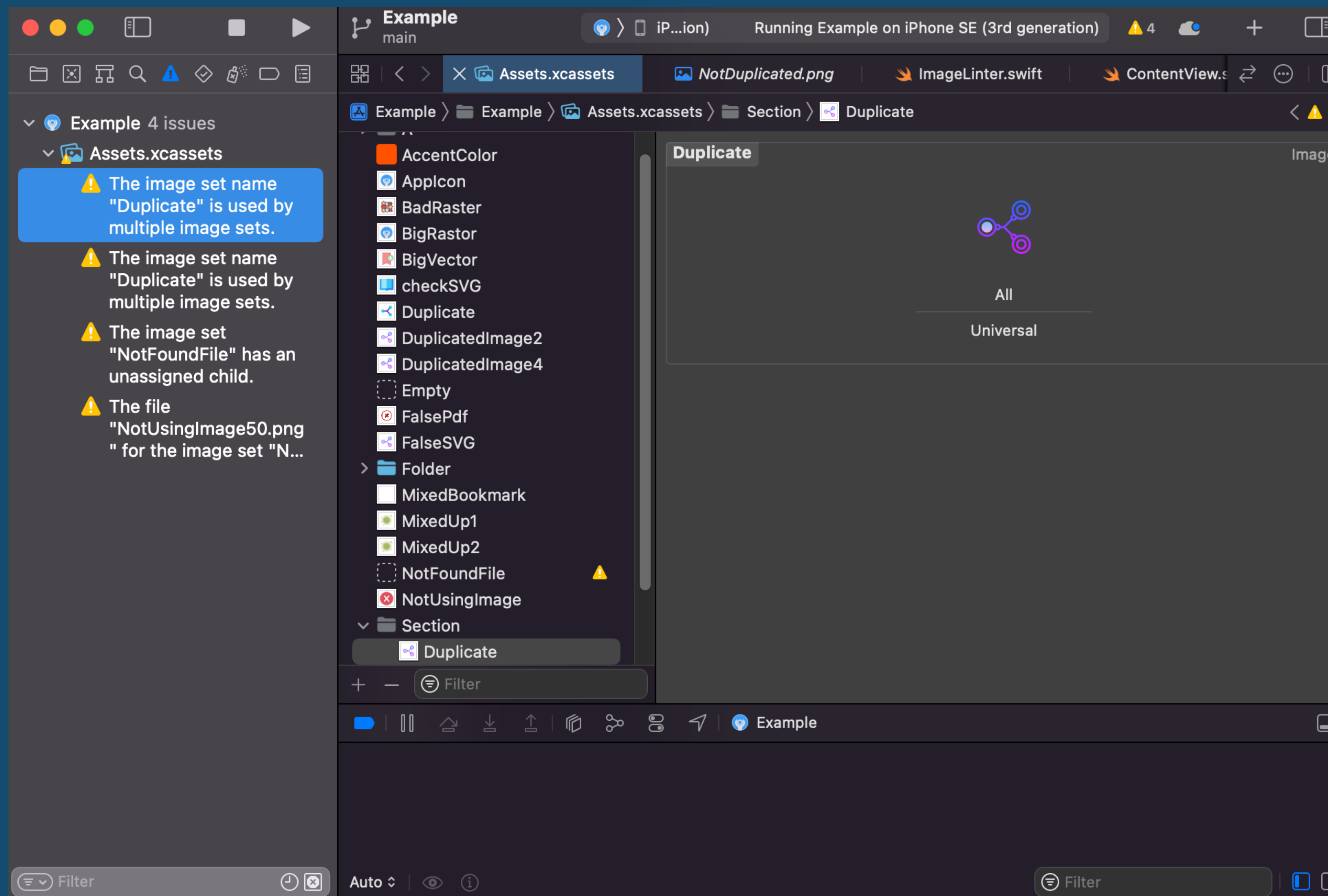
Проверка ресурсов

XCode 12 и более



Проверка ресурсов

XCode 14



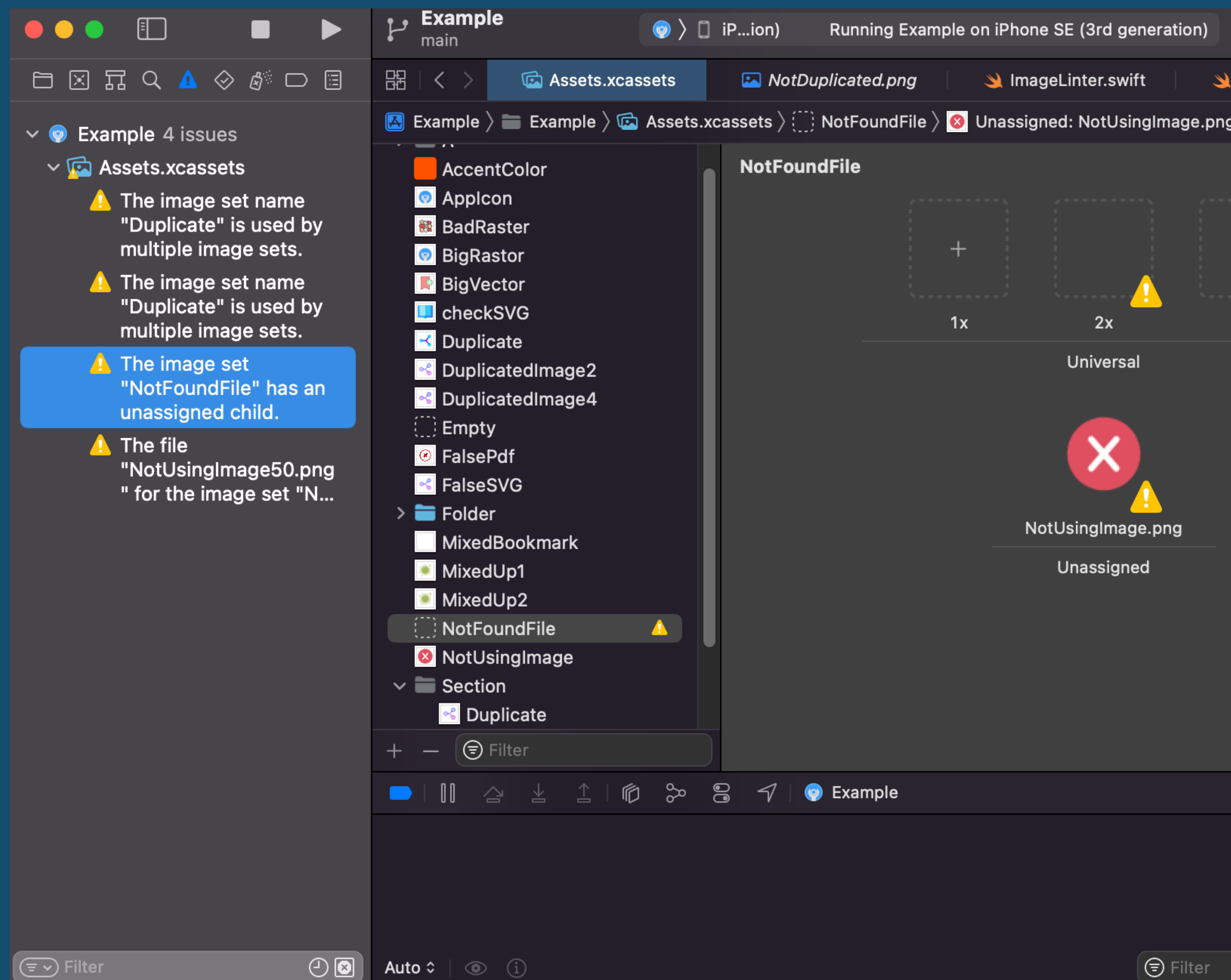
Что Xcode не может

1. Отсутствие изображения



Отсутствие изображения

Проблемы



```
ContentView: View {
  body: some View {
    VStack {
      Image("TruePng")
        .imageScale(.large)
        .foregroundColor(.accentColor)
      Image("TruePng")
      Image("TruePdf")
      Image("FalsePdf")
      Image("Duplicate")

      if let image = UIImage(named:
        "NotDuplicated") {
        Image(uiImage: image)
      }

      Image("checkSVG")
    }
  }
  .padding()
```

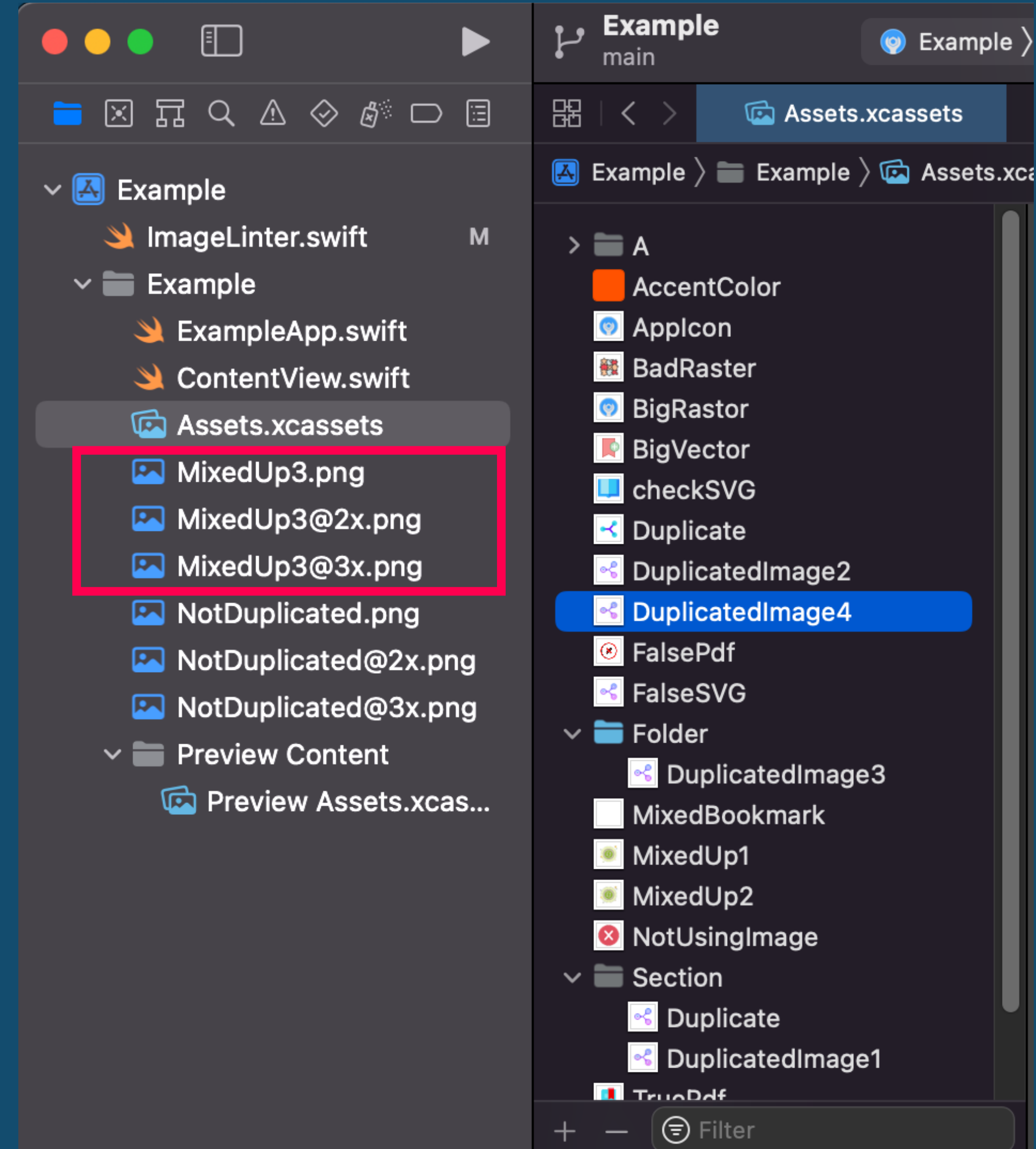
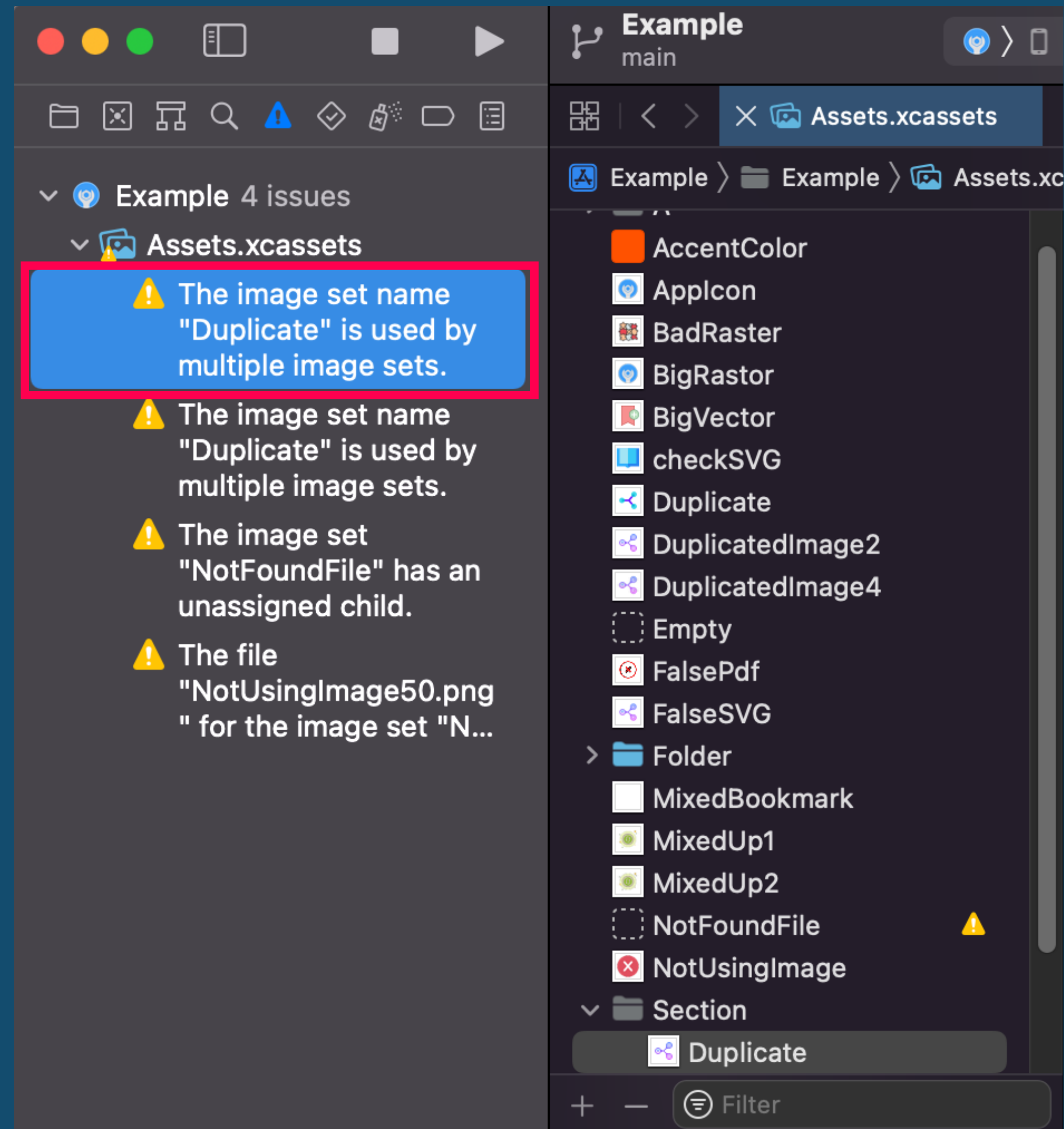
1. Отсутствие изображения

2. Дубликаты названий



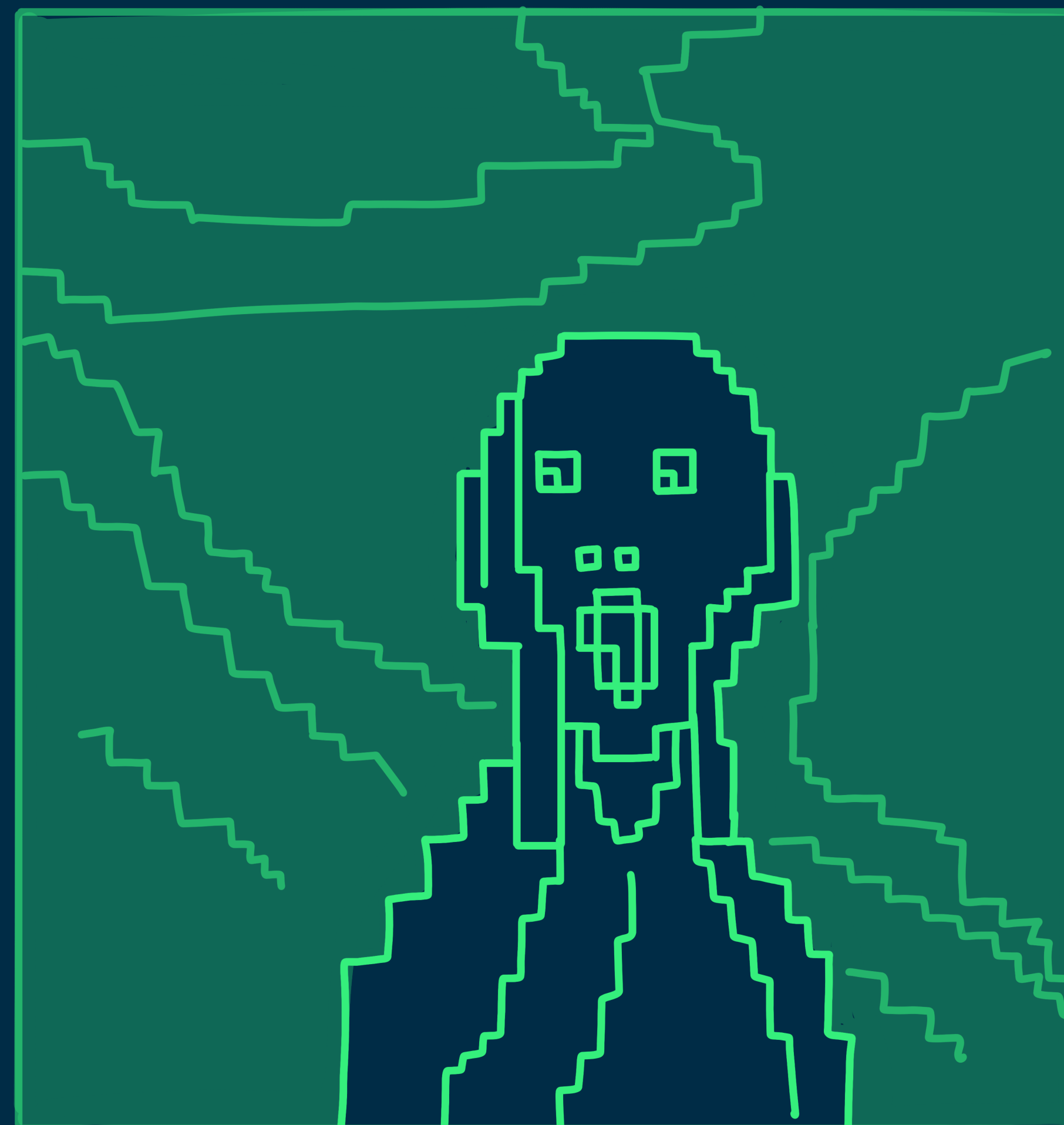
Дубликаты названий

Проблемы



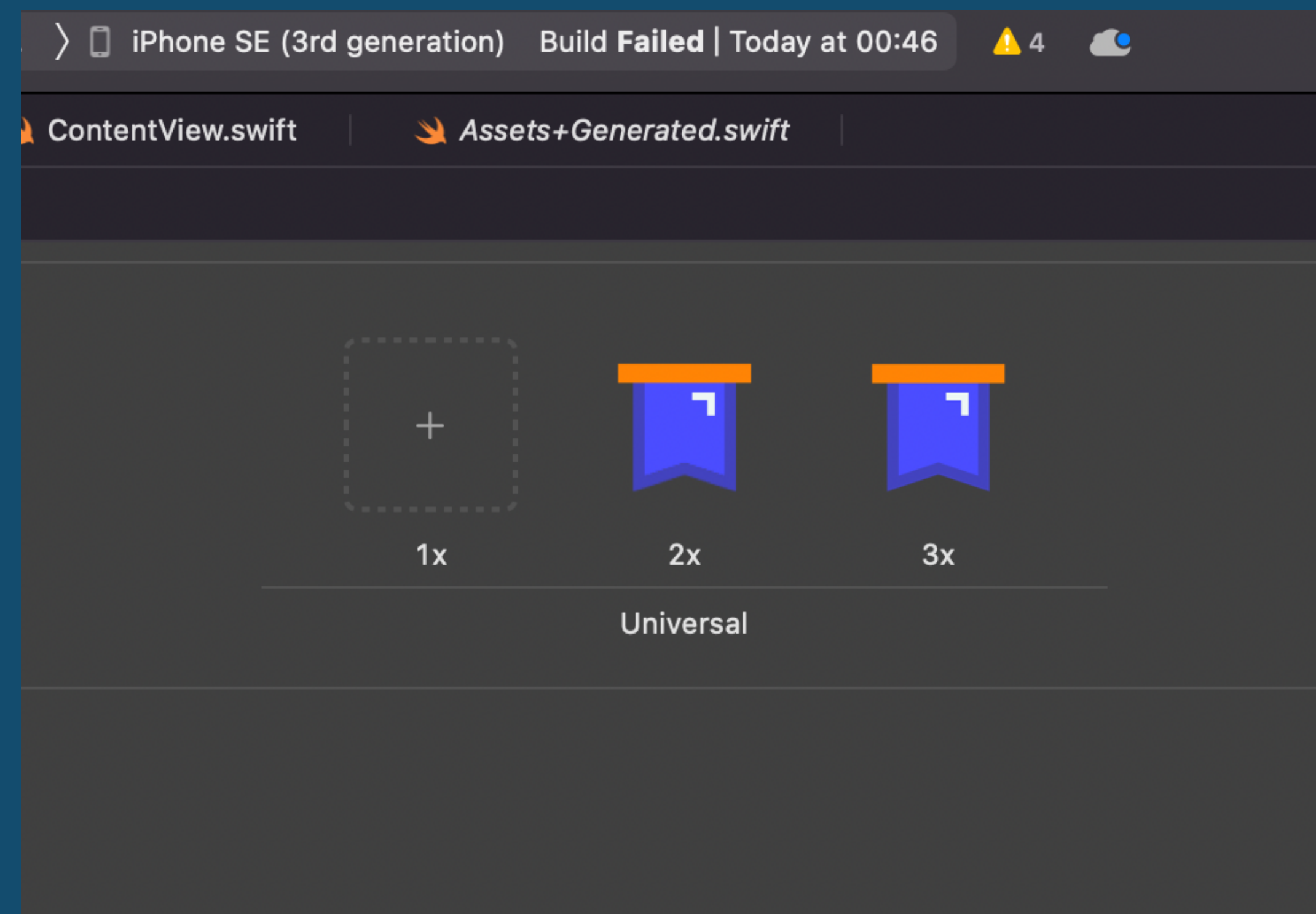
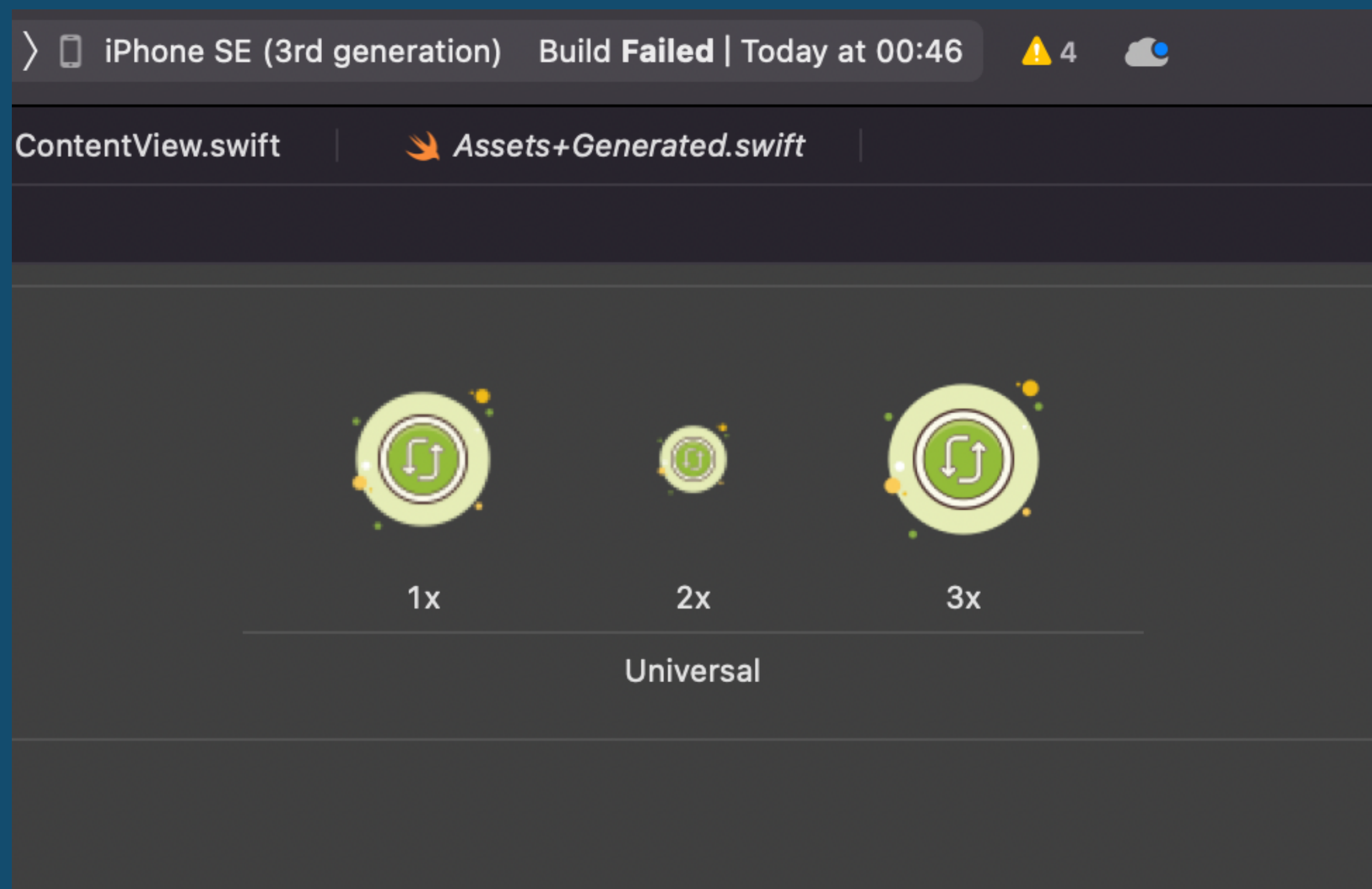
Проблемы на типичном проекте iOS

1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения



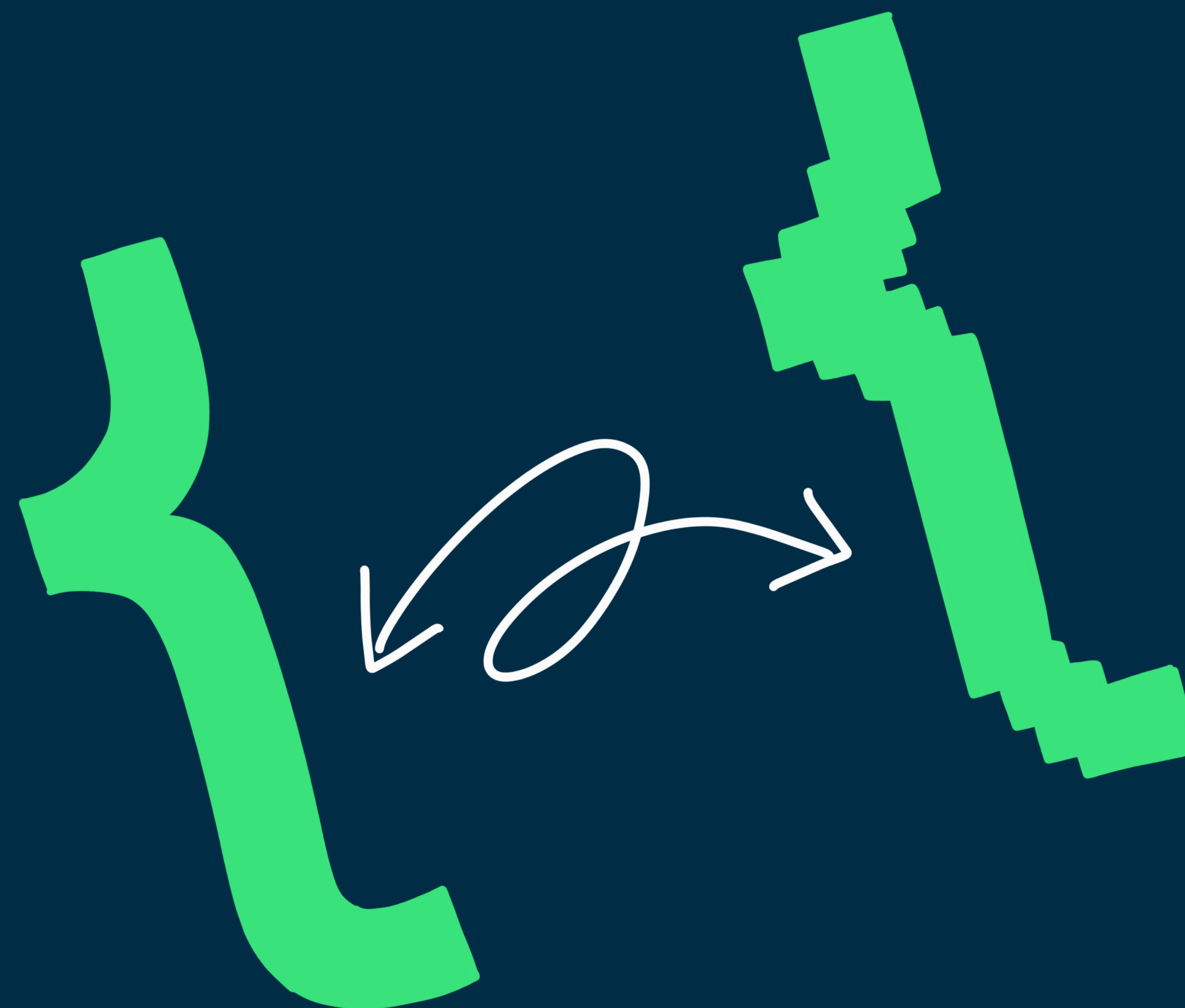
Ошибки разрешений

Проблемы



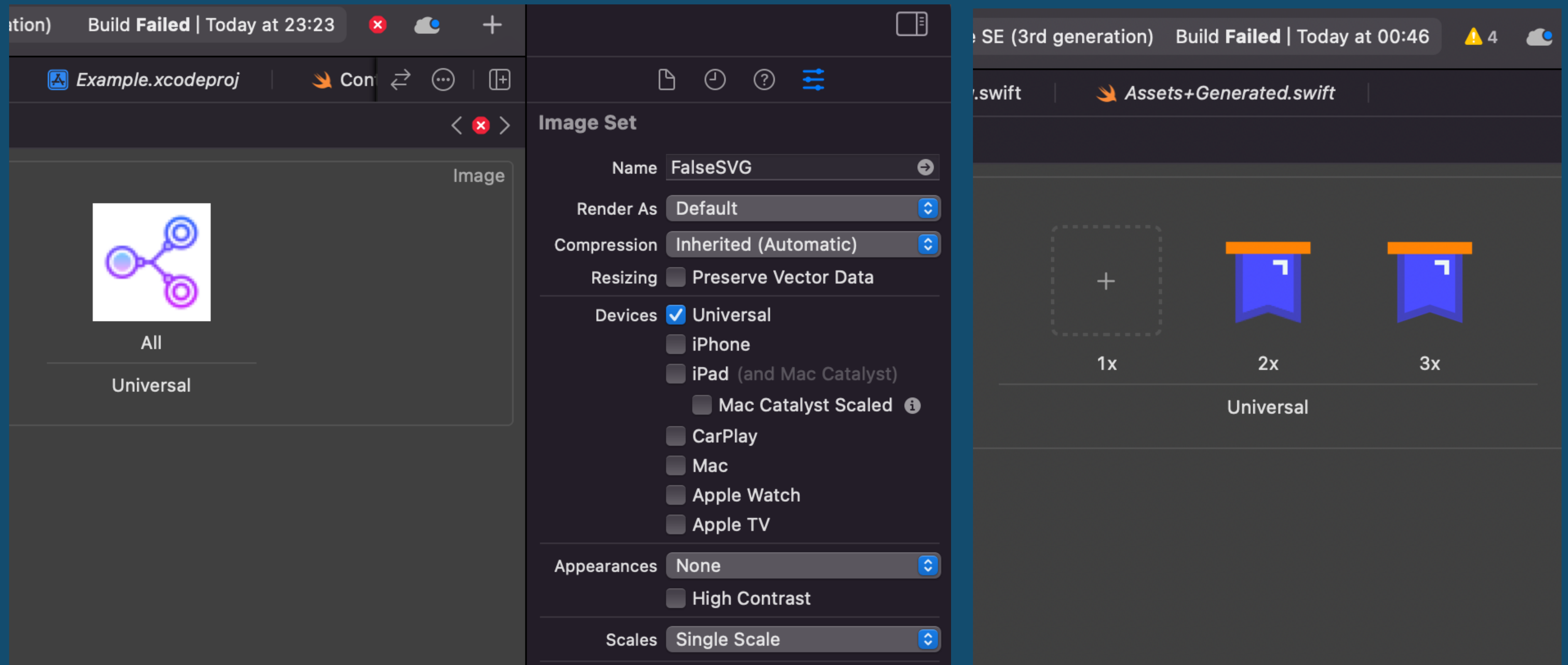
Проблемы на типичном проекте iOS

1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения
4. Растр вместо вектора



Растр вместо вектора

Проблемы



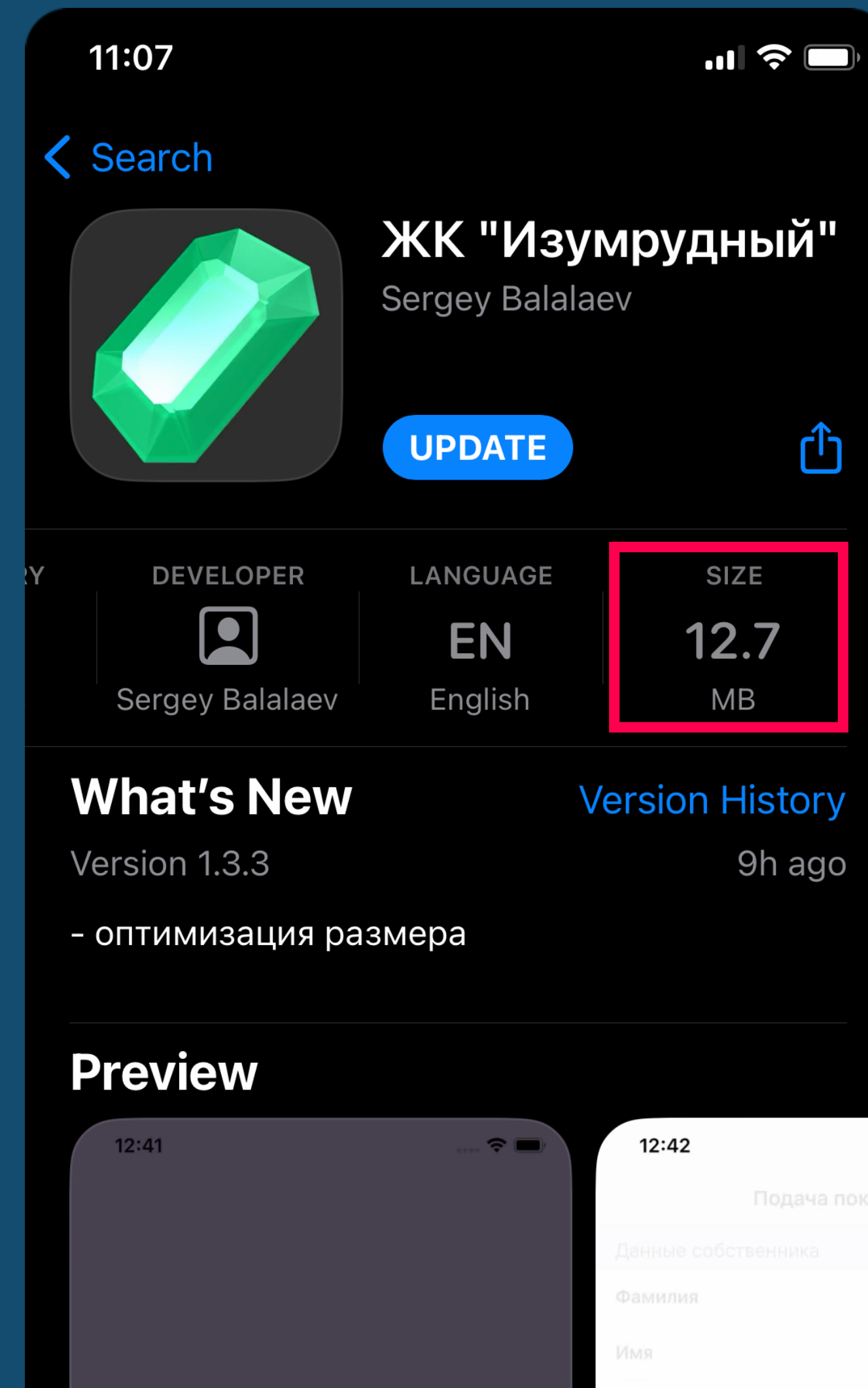
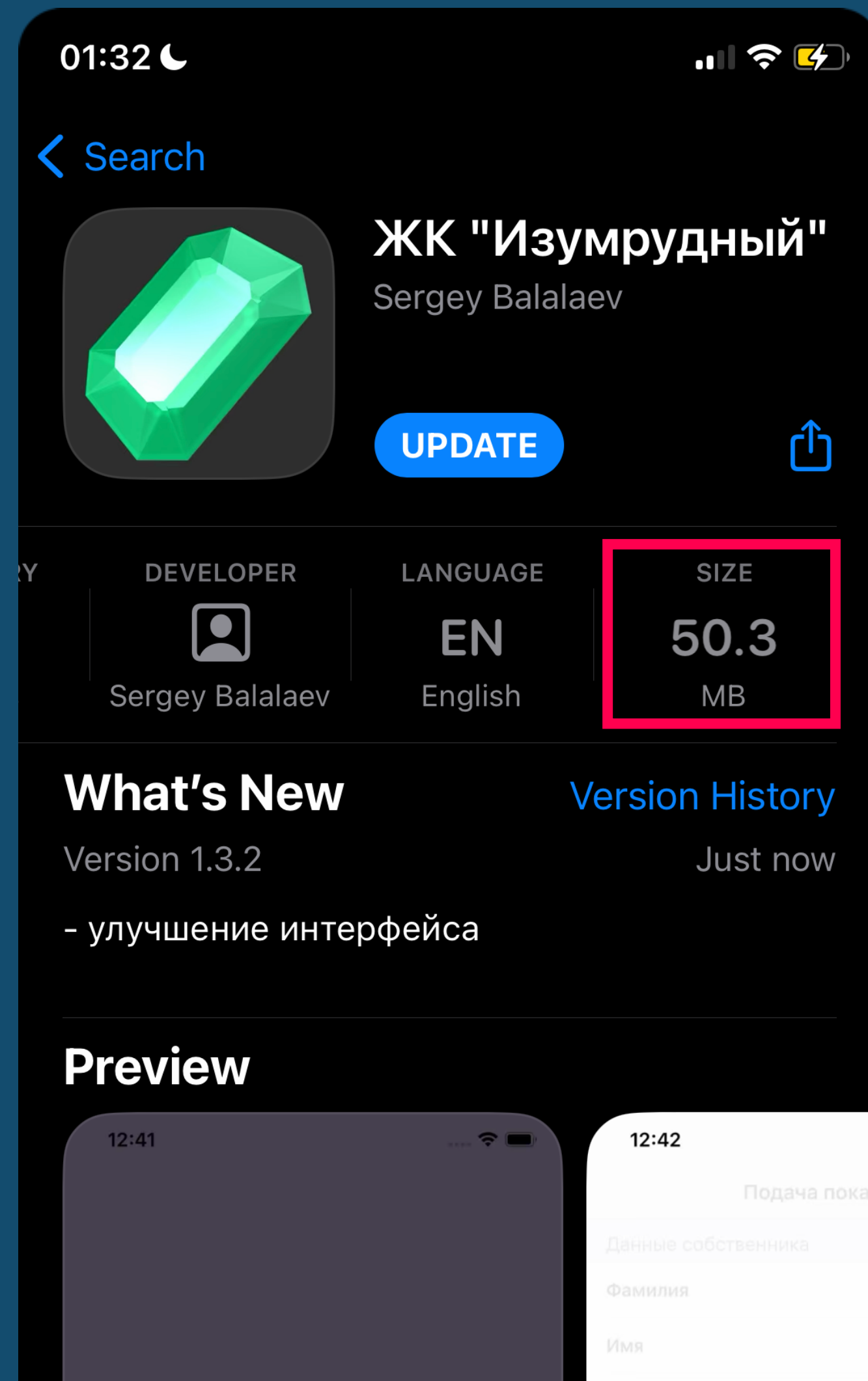
Проблемы на типичном проекте iOS

1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения
4. Растр вместо вектора
5. **Мусор: не используемые и дубли**



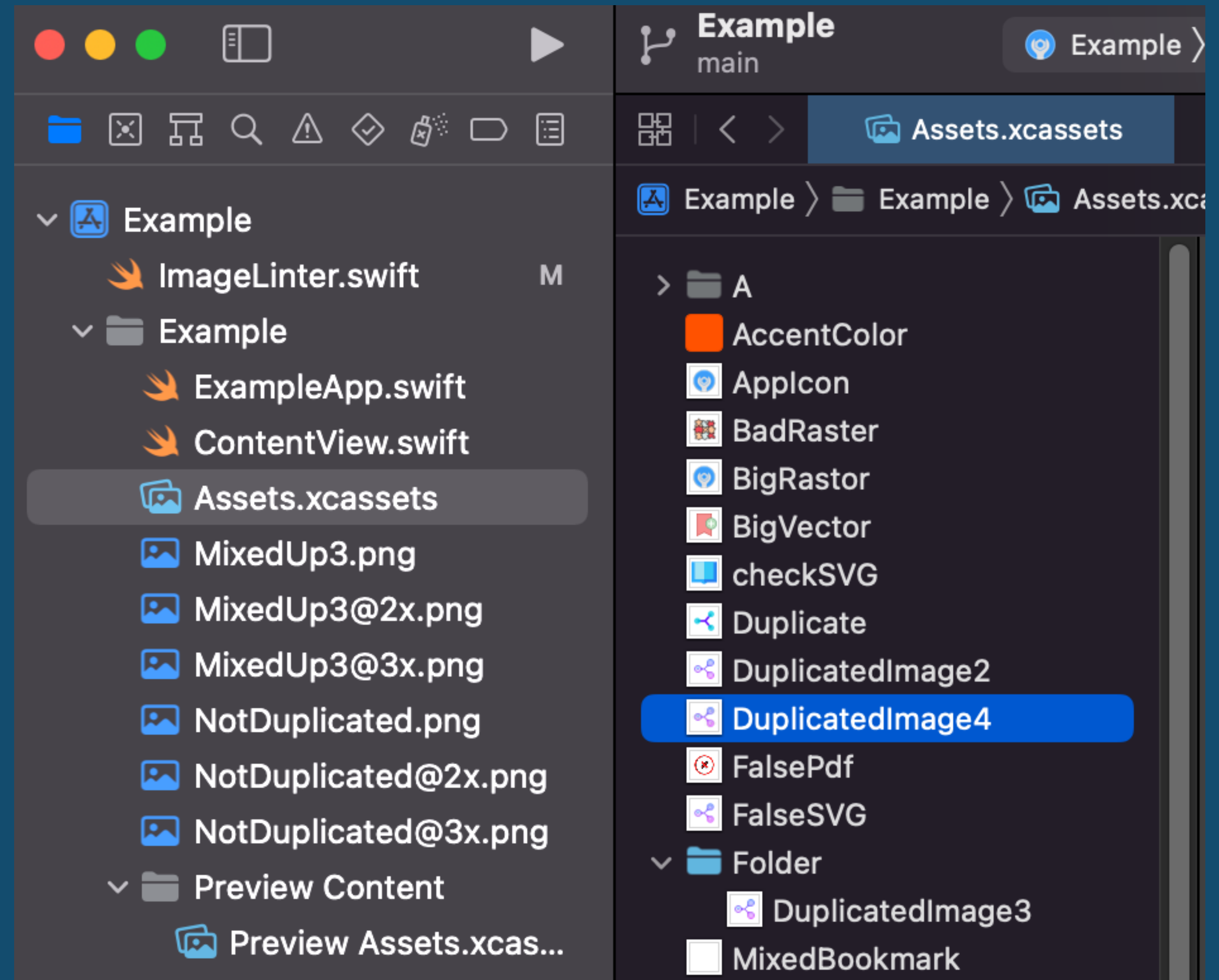
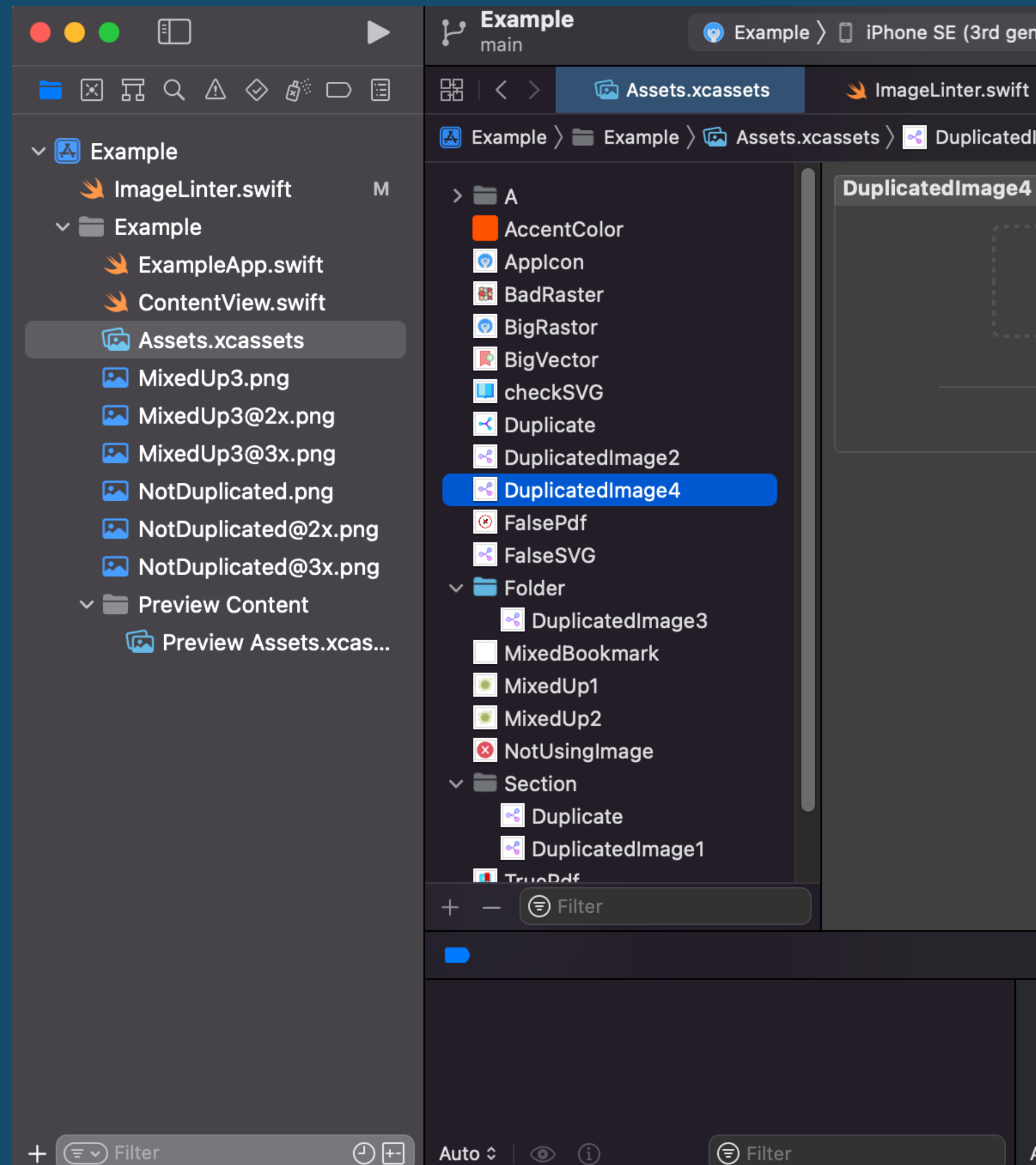
Мусор: не используемые изображения

Проблемы



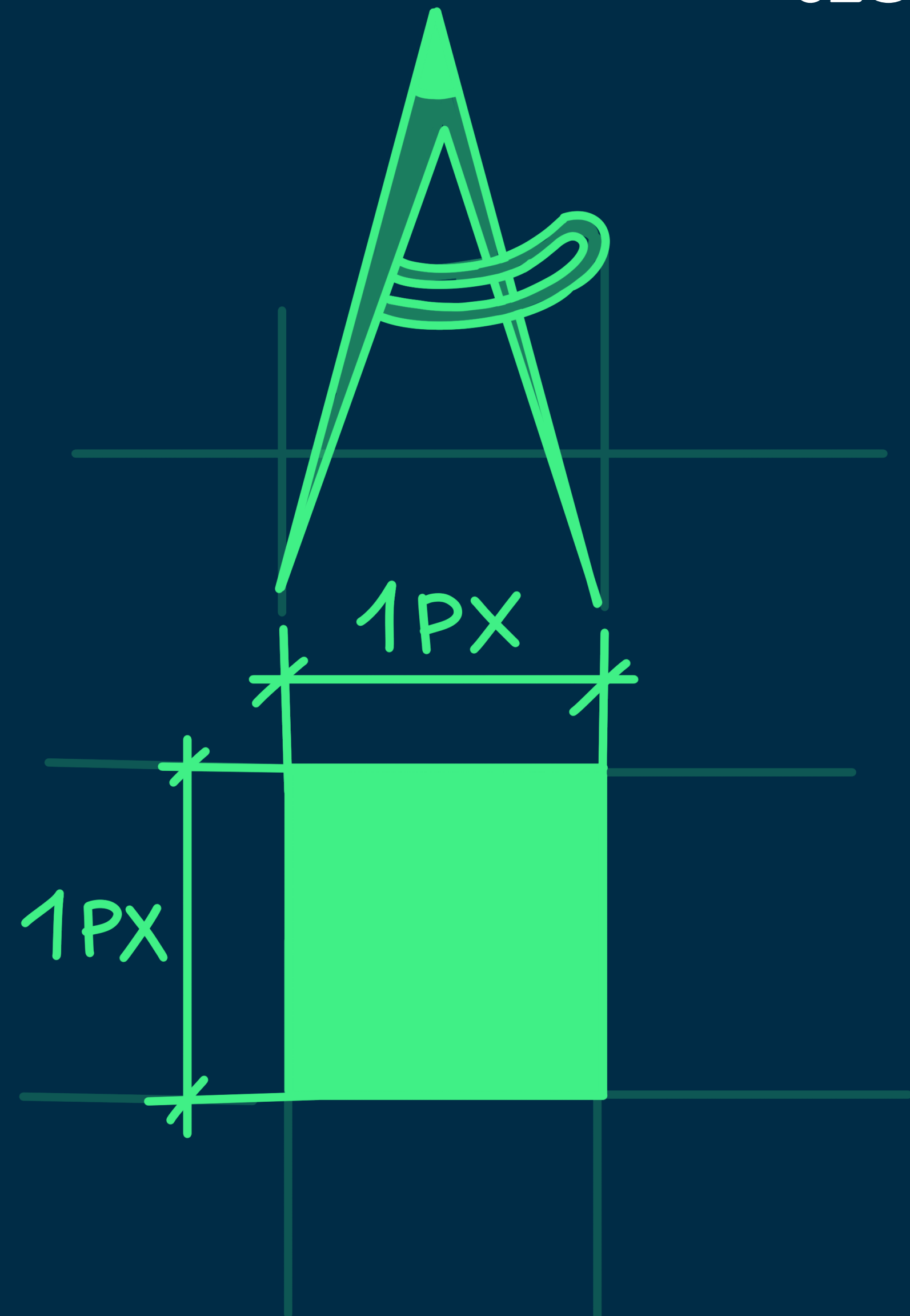
Мусор: дубли — КОПИПАСТ КОНТЕНТА

Проблемы



Проблемы на типичном проекте iOS

1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения
4. Растр вместо вектора
5. Мусор: не используемые и дубли
6. Ошибки размера на пиксели



Ошибки размера на пиксели

Проблемы

100x100



99x99



1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения
4. Растр вместо вектора
5. Мусор: не используемые и дубли
6. Ошибки размера на пиксели

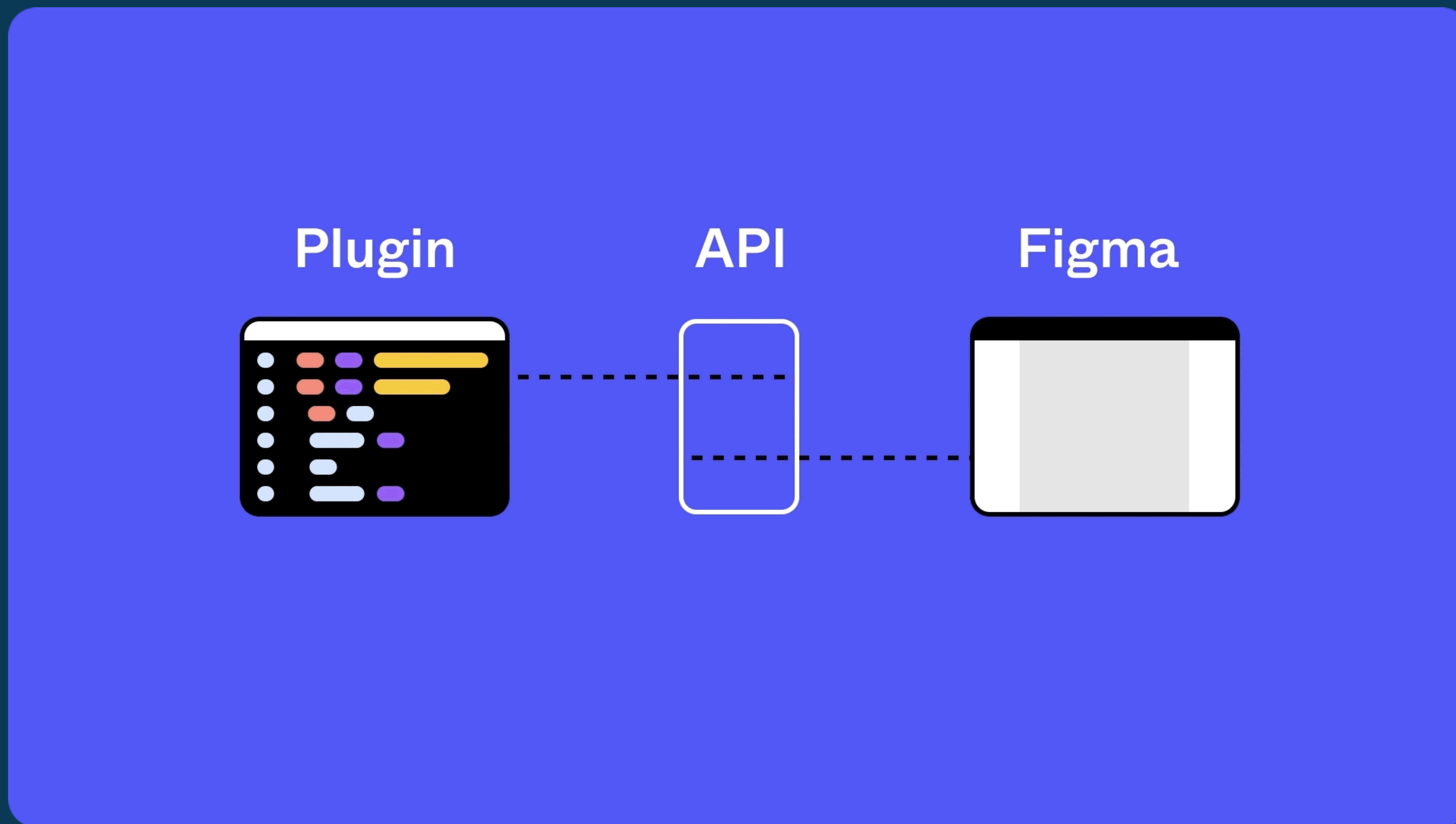
4

Инструменты решения проблем

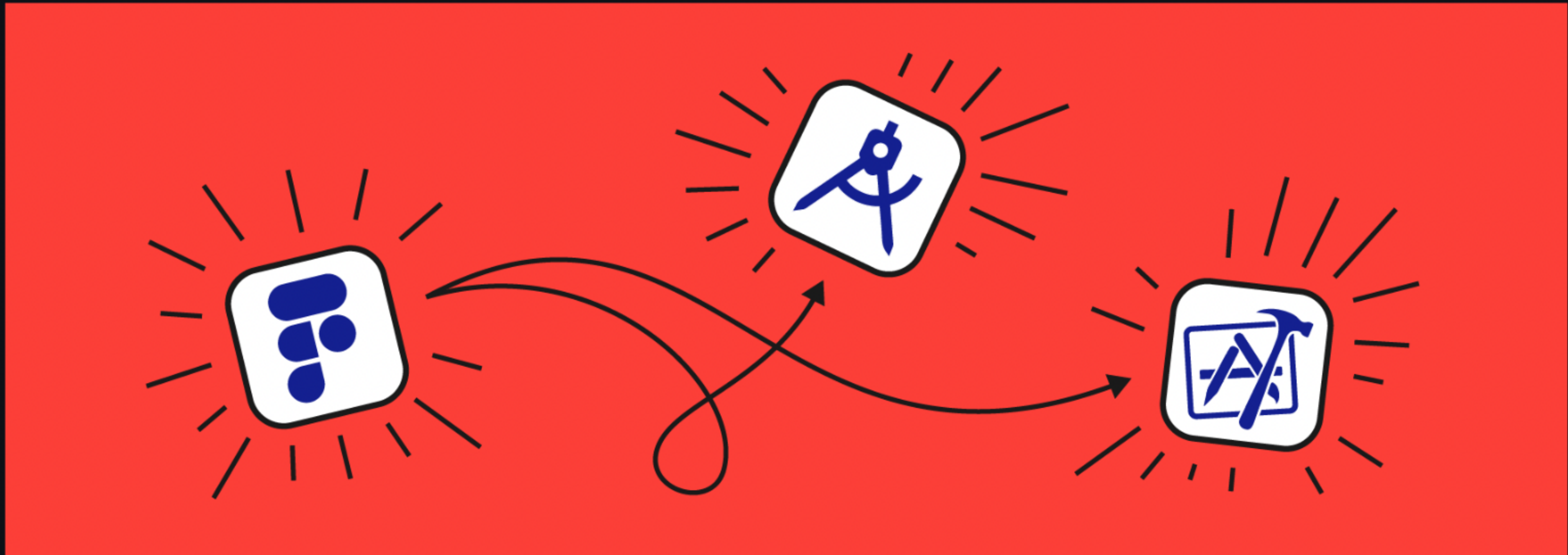


1. Figma-экспорт





FigmaExport



spm compatible license MIT pod v0.37.1 codebeat A Test passing

Starter	Figma Professional	Figma Organization	Figma Enterprise
Free	\$12 per editor/month	\$45 per editor/month	\$75 per editor/month
Forever	Billed annually or \$15 month-to-month	Annual billing only	Annual billing only
<ul style="list-style-type: none">✓ 3 Figma and 3 FigJam files✓ Unlimited personal files✓ Unlimited collaborators✓ Plugins, widgets, and templates✓ Mobile app	<ul style="list-style-type: none">✓ Unlimited Figma files✓ Unlimited version history✓ Sharing permissions✓ Shared and private projects✓ Team libraries✓ Audio conversations	<p>Everything in Professional, plus...</p> <ul style="list-style-type: none">✓ Org-wide libraries✓ Design system analytics✓ Branching and merging✓ Centralized file management✓ Unified admin and billing✓ Private plugins and widgets✓ Single sign-on	<p>For advanced security and more flexible controls, the Enterprise plan helps you scale design processes company-wide.</p>
Choose Starter	Choose Professional	Choose Organization	Contact sales
	Free for students and educators	Contact sales	Learn more

Проблемы, решаемые Figma export API

Buyer

1. **Отсутствие изображения**
2. **Дубликаты названий**
3. **Ошибки разрешения**
4. **Растр вместо вектора**
5. **Мусор: не используемые и дубли**
6. **Ошибки размера на пиксели**



2. Кодогенерация

Известные инструменты на текущий момент

1. SwiftGen

- Работа с другими ресурсами
- Высокая скорость генерации
- Есть шаблонирование и поддержка SwiftUI



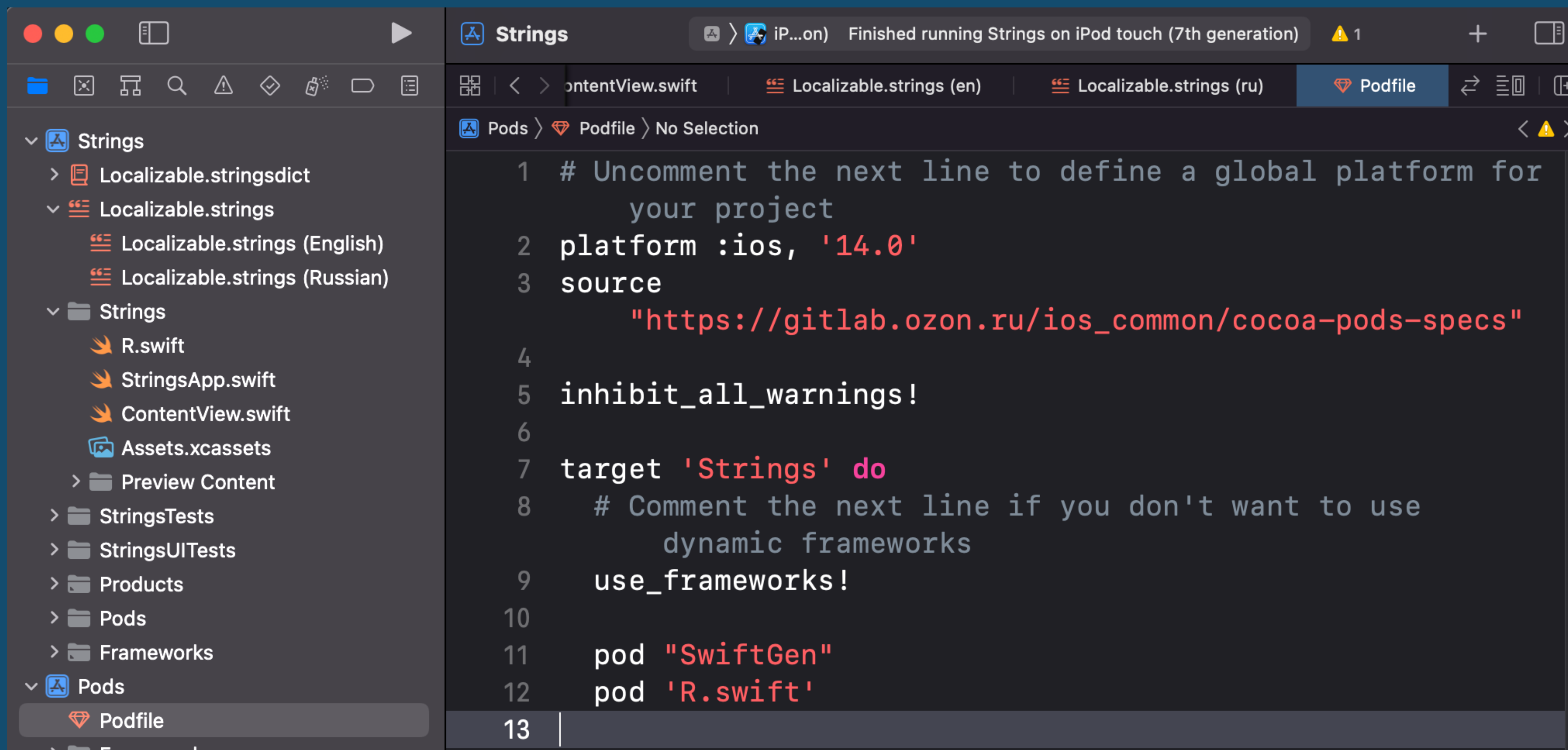
2. R.swift

- Работа с другими ресурсами
- Отсутствует шаблонирование нет поддержки SwiftUI
- Скорость обработки низкая



Подключаем через Pods

SwiftGen



The screenshot shows the Xcode interface with a Podfile open. The left sidebar displays the project structure, including folders for Localizable.strings (English and Russian), Strings, Pods, and Frameworks. The main editor shows the Podfile content:

```
1 # Uncomment the next line to define a global platform for
   your project
2 platform :ios, '14.0'
3 source
   "https://gitlab.ozon.ru/ios\_common/cocoa-pods-specs"
4
5 inhibit_all_warnings!
6
7 target 'Strings' do
8   # Comment the next line if you don't want to use
   dynamic frameworks
9   use_frameworks!
10
11   pod "SwiftGen"
12   pod 'R.swift'
```


Конфигурация YAML в корне проекта

SwiftGen

```
swiftgen.yml
1
2
3  input_dir: Example/
4  output_dir: Example/Generated/
5
6  ## see https://www.raywenderlich.com/23709326-swiftgen-tutorial-for-ios
7
8  xcassets:
9    inputs:
10     - Assets.xcassets
11    outputs:
12     ##templateName: swift5
13     templatePath: SwiftGen/assets.stencil
14     output: Assets+Generated.swift
15
```

Настраиваем на генерацию SwiftUI сущностей

SwiftGen

```
● ● ●  
  
@available(iOS 8.0, tvOS 9.0, watchOS 2.0, macOS 10.7, *)  
{{accessModifier}} var image: Image {  
    let bundle = {{param.bundle|default:"BundleToken.bundle"}}  
    #if os(iOS) || os(tvOS)  
    let image = Image(name, bundle: bundle)  
    #elseif os(macOS)  
    let name = UIImage.Name(self.name)  
    let image = (bundle == .main) ? UIImage(named: name) : bundle.image(forResource: name)  
    #elseif os(watchOS)  
    let image = Image(named: name)  
    #endif  
    return image  
}
```

SwiftGen запускаем в Build Phases

SwiftGen

The screenshot shows the Xcode interface with the 'Build Phases' tab selected. The 'Build Phases' section is expanded to show a list of build phases. The 'SwiftGen' phase is expanded to show its configuration. The shell is set to '/bin/sh' and the script content is as follows:

```
Shell /bin/sh

1 if [[ -f "${PODS_ROOT}/SwiftGen/bin/swiftgen" ]]; then
2   "${PODS_ROOT}/SwiftGen/bin/swiftgen"
3 else
4   echo "warning: SwiftGen is not installed. Run 'pod install --repo-update' to
   install it."
5 fi
```

Получаем генерируемый файл

SwiftGen

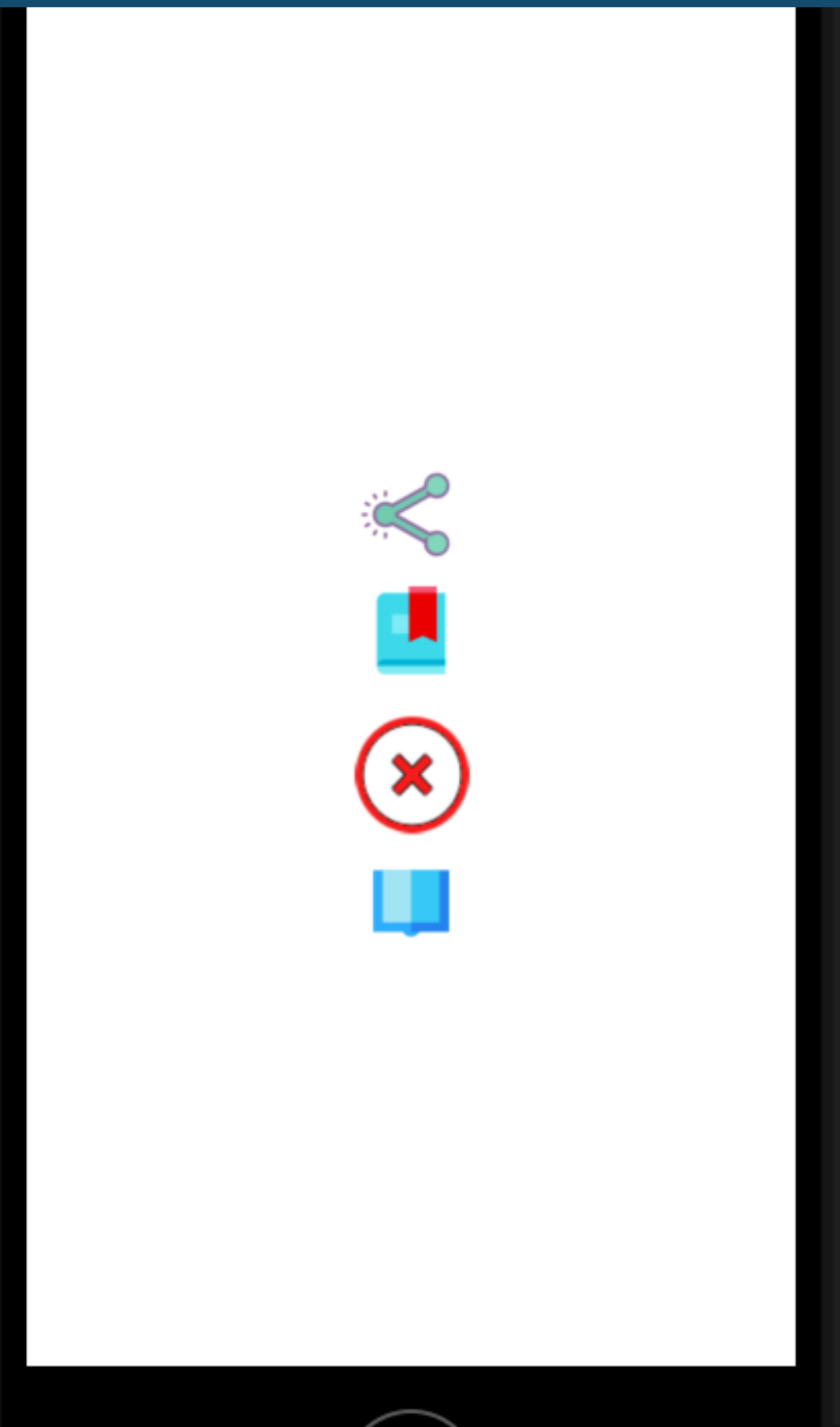
The screenshot shows the Xcode IDE with a project named 'Example' running on an iPhone SE (3rd generation). The project structure on the left includes a 'Generated' folder containing 'Assets+Generated.swift', which is highlighted in blue. The main editor displays the Swift code for 'Assets+Generated.swift'. The code defines an internal enum 'Asset' with various static let properties and an internal enum 'Folder' with a static let property. Two lines of code are highlighted with red boxes and error messages: line 29, 'internal static let duplicate = ImageAsset(name: "Duplicate")', and line 43, 'internal static let duplicate = ImageAsset(name: "Duplicate")'. Both lines have an error message 'Invalid redeclarati...' next to them, indicating a duplicate declaration of the 'duplicate' property.

```
23 internal enum Asset {
24     internal static let duplicate = ImageAsset(name: "Duplicate")
25     internal static let accentColor = ColorAsset(name: "AccentColor")
26     internal static let badRaster = ImageAsset(name: "BadRaster")
27     internal static let bigRaster = ImageAsset(name: "BigRaster")
28     internal static let bigVector = ImageAsset(name: "BigVector")
29     internal static let duplicate = ImageAsset(name: "Duplicate")
30     internal static let duplicatedImage2 = ImageAsset(name: "DuplicatedImage2")
31     internal static let duplicatedImage4 = ImageAsset(name: "DuplicatedImage4")
32     internal static let empty = ImageAsset(name: "Empty")
33     internal static let falsePdf = ImageAsset(name: "FalsePdf")
34     internal static let falseSVG = ImageAsset(name: "FalseSVG")
35     internal enum Folder {
36         internal static let duplicatedImage3 = ImageAsset(name: "Folder/DuplicatedImage3")
37     }
38     internal static let mixedBookmark = ImageAsset(name: "MixedBookmark")
39     internal static let mixedUp1 = ImageAsset(name: "MixedUp1")
40     internal static let mixedUp2 = ImageAsset(name: "MixedUp2")
41     internal static let notFoundFile = ImageAsset(name: "NotFoundFile")
42     internal static let notUsingImage = ImageAsset(name: "NotUsingImage")
43     internal static let duplicate = ImageAsset(name: "Duplicate")
44     internal static let duplicatedImage1 = ImageAsset(name: "DuplicatedImage1")
45     internal static let truePdf = ImageAsset(name: "TruePdf")
46     internal static let truePng = ImageAsset(name: "TruePng")
```

Используем полученную константу/функцию в коде

SwiftGen

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         VStack {
13             Asset.truePng.image
14                 .imageScale(.large)
15                 .foregroundColor(.accentColor)
16             Asset.truePdf.image
17
18             if let image = Asset.falsePdf.uiImage {
19                 Image(uiImage: image)
20             }
21
22             Asset.checkSVG.image
23         }
24         .padding()
25     }
26 }
```



Проблемы, решаемые SwiftGen

Buyer/Seller

1. Отсутствие изображения
2. Дубликаты названий
3. Ошибки разрешения
4. Растр вместо вектора
5. Мусор: не используемые и дубли
6. Ошибки размера на пиксели



3. SwiftSnapshotTesting



Snapshot тесты

swift-snapshot-testing



<https://github.com/pointfreeco/swift-snapshot-testing>



1. Упростилось кодревью у разработчиков
2. Возможность дизайнревью
3. Безболезненная модификация визуальных компонентов
4. Отслеживание изменений дизайн-системы

Проблемы, отслеживаемые SnapshotTesting

Seller

1. ~~Отсутствие изображения~~
2. ~~Дубликаты названий~~
3. ~~Ошибки разрешения~~
4. Растр вместо вектора
5. Мусор: не используемые и дубли
6. Ошибки размера на пиксели



5

Решение в iOS приложениях для ПВЗ



UIKit из Figma

SwiftGen

swift-snapshot-testing

Imagelinter

Imagelinter

Идея возникла после внедрения Localinter

ozon{tech

Localinter



<https://habr.com/ru/companies/ozontech/articles/734214/>

Imagelinter

1. Проверка размеров файла и изображения
2. Проверка PDF и SVG на растр
3. Проверка не используемых в коде изображений
4. Поиск неопределенных изображений
5. Сравнение масштабируемых x1, x2, x3 изображений между собой
6. Проверка дубликатов по имени и по содержимому
7. Поиск пустых и битых изображений в Assets

Просто скачиваем Imagerlinter.swift

Imagerlinter

OZON{tech

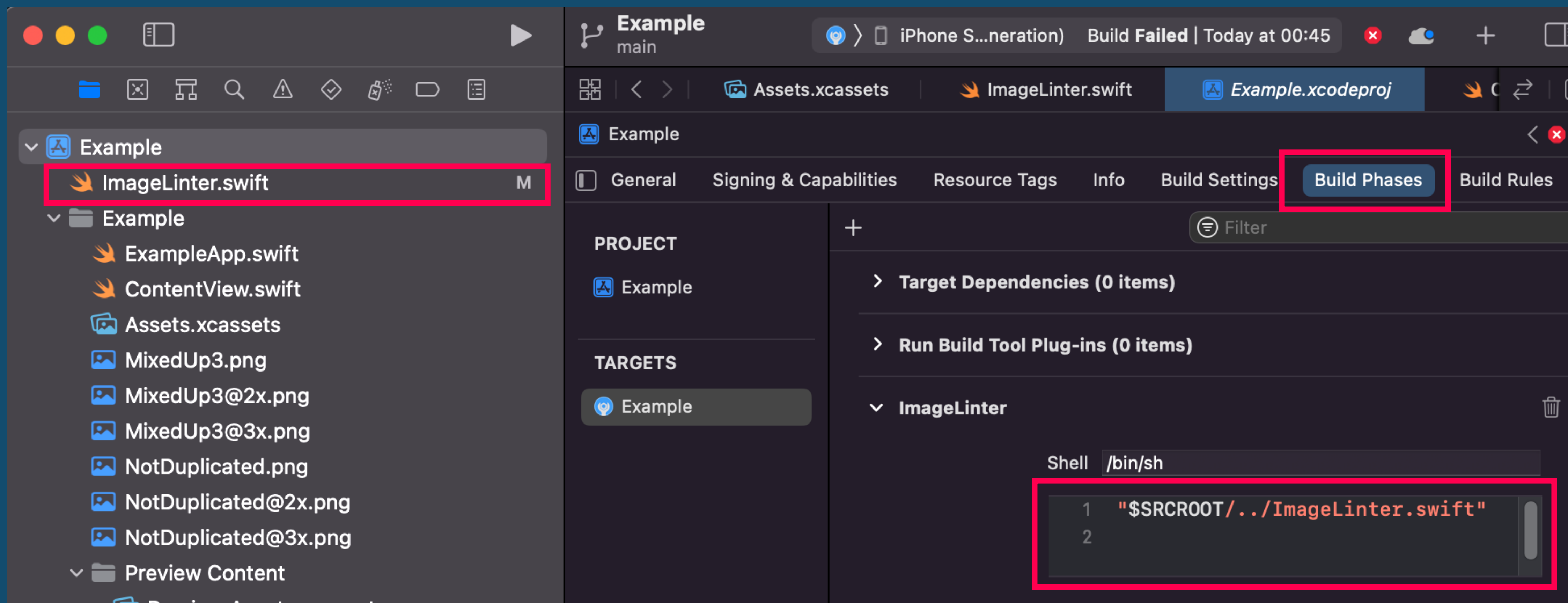


<https://github.com/ByteriX/ImageLinter>



Добавляем ImageLinter.swift в проект

ImageLinter



Настраиваем ImageLinter.swift






ImageLinter

```
Example > ImageLinter.swift > No Selection
28 let isEnabled = true
29
30 /// Path to folder with images files. For example "/YourProject/Resources/Images"
31 let relativeImagePath = "/"
32
33 /// Path of the source folder which will used in searching for localization keys you actually use in your
    project. For Example "/YourProject/Source"
34 let relativeSourcePath = "/"
35
36 /// Using localizations type from code. If you use custom you need define regex pattern
37 enum UsingType {
38     case swiftUI
39     case uiKit
40     case swiftGen(enumName: String = "Asset")
41     case custom(pattern: String)
42 }
43
44 /// yuo can use many types
45 let usingTypes: [UsingType] = [
46     .swiftGen(), .swiftUI, .uiKit
47 ]
48
```

Imagelinter

```
57 let ignoredUnusedImages: Set<String> = [  
58 ]  
59 let ignoredUndefinedImages: Set<String> = [  
60 ]  
61  
62 let rasterExtensions = ["png", "jpg", "jpeg"]  
63 let vectorExtensions = ["pdf", "svg"]  
64 let rasterSetExtensions = Set<String>(rasterExtensions.map{$0.uppercased()})  
65 let vectorSetExtensions = Set<String>(vectorExtensions.map{$0.uppercased()})  
66 let imageSetExtensions = rasterSetExtensions.union(vectorSetExtensions)  
67  
68 // Maximum size of Vector files  
69 let maxVectorFileSize: UInt64 = 20_000  
70 let maxVectorImageSize: CGSize = CGSize(width: 100, height: 100)  
71  
72 // Maximum size of Raster files  
73 let maxRasterFileSize: UInt64 = 200_000  
74 let maxRasterImageSize: CGSize = CGSize(width: 1000, height: 1000)  
75  
76 let isCheckingFileSize = true  
77 let isCheckingImageSize = true  
78 let isCheckingPdfVector = true  
79 let isCheckingSvgVector = true  
80 let isCheckingScaleSize = true  
81 let isCheckingDuplicatedByName = true  
82 let isCheckingDuplicatedByContent = true
```

Imagelinter

- ✗ Image has floating size from scaled images. Real size is (64.0, 64.0) and scale = 3. Found for image 'mixedUp2'
- ✓  2_3.png
 - ✗ File unused from code. Found for image 'mixedUp2'
- ✓  a1.png
 - ✗ File unused from code. Found for image 'badRaster'
- ✓  a2.png
 - ✗ File unused from code. Found for image 'badRaster'
 - ✗ Image has floating size from scaled images. Real size is (99.0, 99.0) and scale = 2. Found for image 'badRaster'
- ✓  BigRastor.png
 - ✗ File unused from code. Found for image 'bigRastor'
- ✗ The raster image has very biggest image size (1024, 1024). Max image size for raster is (1000.0, 1000.0). Fo...
- ✓  BigVector.pdf
 - ✗ File unused from code. Found for image 'bigVector'
 - ⚠ It is vector image. But it has scale = 3. Found for image 'bigVector'



Во время внедрения на проекте Seller

Imagelinter

```
seller-app-ios — -zsh — 80x24
Resources/Sources/SellerResources/Resources/Assets/Images.xcassets/Badges/badgeColorSample.imageset/badge-3.pdf: error: File unused from code. Found for image 'badgeColorSample'
Resources/Sources/SellerResources/Resources/Assets/Images.xcassets/Logo/SellerLogo.imageset/SellerLogo.svg: error: File unused from code. Found for image 'sellerLogo'
Number of images: 203
Number of warnings: 0
Number of errors: 187
Path:
Time: 1.6954489946365356 sec.
seller-app-ios % scripts/ImageLinter.swift
UI/Sources/SellerUI/ViewModels/DefaultDisclaimerViewModel.swift:22: error: Not found image with name 'confirmed' or 'confirmed'
Number of images: 203
Number of warnings: 0
Number of errors: 1
Path:
Time: 1.7170519828796387 sec.
```


Проблемы, решаемые на нашем проекте

ПВЗ

1. ~~Отсутствие изображения~~
 2. ~~Дубликаты названий~~
 3. ~~Ошибки разрешения~~
 4. ~~Растр вместо вектора~~
 5. ~~Мусор: не используемые и дубли~~
 6. ~~Ошибки размера на пиксели~~
- + Анализ контента, размеров
 - + Время выполнения ~0,8 сек (85 img's)



6

Заключение



1. Экспорт из дизайн системы

- API Figma

2. Кодогенерация SwiftGen

- Работает быстро и сердито, поддерживает SwiftUI

3. Review SnapShot tests

- Для визуального ревью

4. Автоматизация проверки Imagerinter

- Swift-скрипт, покрытие ошибок большее и однозначнее



Спасибо за внимание

Сергей Балалаев

Руководитель отдела разработки
мобильного приложения “Ozon ПВЗ”

