SWIFT *Scripting*

AYAKA NONAKA

@AYANONAGON

Designers

ENGINEERS

- Designer designs and exports assets

- Designer designs and exports assets
- Upload to Dropbox or send over Slack

- Designer designs and exports assets
- Upload to Dropbox or send over Slack
- Engineer adds to venmo-ios codebase

- So many Dropbox links & Slack messages, gets confusing

- So many Dropbox links & Slack messages, gets confusing
- Sometimes assets sizes are wrong

- So many Dropbox links & Slack messages, gets confusing
- Sometimes assets sizes are wrong
- No easy way to see all the assets in our app


POD 'VENMO-IOS-IMAGES'

- Designer designs and exports assets

- Designer designs and exports assets
- Designer makes pull-request to venmo-ios-images repo

- Designer designs and exports assets
- Designer makes pull-request to venmo-ios-images repo
- Engineer reviews changes and merges

- Designer designs and exports assets
- Designer makes pull-request to venmo-ios-images repo
- Engineer reviews changes and merges
- Engineer generates UIImage category for the new image(s)

- Designer designs and exports assets
- Designer makes pull-request to venmo-ios-images repo
- Engineer reviews changes and merges
- Engineer generates UIImage category for the new image(s)
- pod update 🚀

- Engineer generates UIImage category for the new image(s)

WE'RE *LAZY*

AUTOMATE ALL THE THINGS!

Script ALL THE THINGS!

SCRIPTING

- Run from CLI

SCRIPTING

- Run from CLI
- Light-weight (no Xcode projects?)


WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
PERL!
TAP TAP

SCRIPTING

- Run from CLI
- Light-weight (no Xcode projects?)

SCRIPTING

- Run from CLI
- Light-weight (no Xcode projects?)
- Tooling

JANUARY 22ND, 2015

Ayaka Nonaka [11:07 AM]
ok time to write..... ruby? or.... swift?!

JANUARY 22ND, 2015

Ayaka Nonaka [11:07 AM]
ok time to write..... ruby? or.... swift?!
actually not sure if i want to go down that path

JANUARY 22ND, 2015

Ayaka Nonaka [11:07 AM]
ok time to write..... ruby? or.... swift?!
actually not sure if i want to go down that path
a 2 day long project is going to turn into like a 7 day long project lol

JANUARY 22ND, 2015

Ayaka Nonaka [11:07 AM]
ok time to write..... ruby? or.... swift?!
actually not sure if i want to go down that path
a 2 day long project is going to turn into like a 7 day long project lol
i might look into swift though

JANUARY 22ND, 2015

Ayaka Nonaka [11:07 AM]
ok time to write..... ruby? or.... swift?!
actually not sure if i want to go down that path
a 2 day long project is going to turn into like a 7 day long project lol
i might look into swift though
that might be kind of fun

JANUARY 22ND, 2015

Eli Perkins [11:08 AM]
swift could be fun!

CHALLENGE ACCEPTED

```
#!/usr/bin/env xcrun swift

import Foundation

// MARK: - Helpers

func writeNewline(outputStream: NSOutputStream) {
    outputStream.write("\n", maxLength: 1)
}

func writeLine(outputStream: NSOutputStream, string: String) {
    outputStream.write(string, maxLength: countElements(string))
    writeNewline(outputStream)
}

func methodDeclarationFromFileName(fileName: String) -> String {
    let noUnderscoresFileName = fileName.stringByReplacingOccurrencesOfString("_", withString: "", options: .LiteralSearch, range: nil)
    let uncapitalizedFileName = uncapitalized(noUnderscoresFileName)
    return "+ (UIImage *)ven_\(uncapitalizedFileName)"
}

func uncapitalized(string: String) -> String {
    var index = advance(string.startIndex, 1)
    let first = string.substringToIndex(index)
    let rest = string.substringFromIndex(index)
    return "\(first.lowercaseString)\(rest)"
}

func fileNameWithoutExtension(fileName: String) -> String {
    let pattern = "(([a-zA-Z]|_)*)(@2x|@3x)?\\.png"
    let expression = NSRegularExpression(pattern: pattern, options: nil, error: nil)
    let result = expression!.matchesInString(fileName, options: nil, range: NSRange(location: 0, length: countElements(fileName)))
    let textCheckingResult = result[0] as NSTextCheckingResult
    let range = textCheckingResult.rangeAtIndex(1)
    return (fileName as NSString).substringWithRange(range)
}

func imageFileNames(path: String) -> [String] {
    let fileManager = NSFileManager.defaultManager()
    let enumerator = fileManager.enumeratorAtPath(path)

var fileNames = NSMutableOrderedSet()
    while let fileName = enumerator?.nextObject() as? String {
        if fileName.hasSuffix("png") {
            fileNames.addObject(fileNameWithoutExtension(fileName))
        }
    }
    return fileNames.array as [String]
}


// MARK: - Script that generates UIImage+VenmoImages category class

let fileNames = imageFileNames("generated-assets")

let categoryName = "UIImage+VenmoImages"
let headerFileName = "\(categoryName).h"

// Generate the interface
if let outputStream = NSOutputStream(toFileAtPath: headerFileName, append: false) {
    outputStream.open()
    writeLine(outputStream, "#import <Foundation/Foundation.h>")
    writeNewline(outputStream)
    writeLine(outputStream, "@interface UIImage (VenmoImages)")
    writeNewline(outputStream)

    for fileName in fileNames {
        writeLine(outputStream, "\(methodDeclarationFromFileName(fileName));")
    }

    writeNewline(outputStream)
    writeLine(outputStream, "@end")
    outputStream.close()
} else {
    println("Unable to open interface file")
}

// Generate the implementation
if let implOutputStream = NSOutputStream(toFileAtPath: "\(categoryName).m", append: false) {
    implOutputStream.open()
    writeLine(implOutputStream, "#import \"\(headerFileName)\"")
    writeNewline(implOutputStream)
    writeLine(implOutputStream, "@implementation UIImage (VenmoImages)")
    writeNewline(implOutputStream)

    for fileName in fileNames {
        writeLine(implOutputStream, "\(methodDeclarationFromFileName(fileName))")
        writeLine(implOutputStream, "{")
        writeLine(implOutputStream, "    return [UIImage imageNamed:@\"Venmo.bundle/\(fileName)\"];")
        writeLine(implOutputStream, "}")
        writeNewline(implOutputStream)
    }

    writeLine(implOutputStream, "@end")
    implOutputStream.close()
} else {
    println("Unable to open implementation file")
}
```

SWIFT

- familiar API's

SWIFT

- familiar API's
- "light" feeling language

SWIFT

- familiar API's
- "light" feeling language
- overly protective compiler is a caring compiler

REDUCE LANGUAGE DEPENDENCIES

NON “PRODUCTION” USE

JUST ANOTHER REASON TO WRITE *Swift*

XCRUN SWIFT HELLO-WORLD.SWIFT

CHMOD +X HELLO-WORLD.SWIFT

#!/USR/BIN/ENV XCRUN SWIFT

./HELLO-WORLD.SWIFT

Live Demo!

DEPENDENCY MANAGEMENT?

GIT SUBMODULES

- Simple, does very little.
  - Maybe too little?

COCOAPODS

- Swift!
- Does dependency management.
- Relies on Xcode project.

CARTHAGE

- Swift!
- Does dependency management.
- Manually add framework to project.

CARTFILE

```
github "jdhealy/PrettyColors"
github "Alamofire/Alamofire"
```

$ CARTHAGE BOOTSTRAP

```
*** No Cartfile.resolved found, updating dependencies
*** Fetching PrettyColors
*** Fetching Alamofire
*** Checking out Alamofire at "1.1.4"
*** Checking out PrettyColors at "v1.0.0"
*** xcodebuild output can be found in /var/folders/blah
*** Building scheme "Alamofire iOS" in Alamofire.xcworkspace
*** Building scheme "Alamofire OSX" in Alamofire.xcworkspace
*** Building scheme "PrettyColors iOS" in PrettyColors.xcodeproj
*** Building scheme "PrettyColors Mac" in PrettyColors.xcodeproj
```

```
➜  swiftsummit git:(master) ✗ ls Carthage/Build/iOS
Alamofire.framework    PrettyColors.framework
```

/LIBRARY/FRAMEWORKS

?

```
➜  swiftsummit git:(master) ✗ swift -help
OVERVIEW: Swift compiler

USAGE: swift [options] <inputs>

OPTIONS:
  ...
  -F <value>                      Add directory to framework search path
```

```
#!/usr/bin/env xcrun swift -F Carthage/Build/iOS
```

THE MISSING STUFF

- Auto-completion

THE MISSING STUFF

- Auto-completion
- A better way to do dependency management

THE MISSING STUFF

- Auto-completion
- A better way to do dependency management
- More tools

- Keithbsmiley/swift.vim
- jdhealy/PrettyColors
- thoughtbot/Argo
- kylef/CLIKit
- nomothetis/SemverKit
- BananaKit? 🍌
- ???


PLOT TWIST

●●●○○ giffgaff 06:59 100%

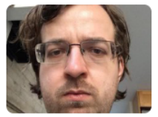
Tweet

NeoNacho
@NeoNacho

@ayanonagon github.com/neonichu/Rome 💥

GitHub

neonichu/Rome



Rome makes it easy to build a list of frameworks. Contribute to Rome development by creating an account on GitHub.

View on web

3/21/15, 02:38 from London, England

1 FAVOURITE

Reply to NeoNacho

Timelines Notifications Messages Me

neonichu / Rome

Watch 0 | Star 2 | Fork 0

Rome makes it easy to build a list of frameworks.

2 commits | 1 branch | 1 release | 1 contributor

branch: master | Rome / +

First prototype.

neonichu authored 6 hours ago — latest commit 88b33e11f8

lib	First prototype.	6 hours ago
spec	First prototype.	6 hours ago
.gitignore	Initial commit	6 hours ago
Gemfile	Initial commit	6 hours ago
Gemfile.lock	First prototype.	6 hours ago
LICENSE.txt	Initial commit	6 hours ago
README.md	First prototype.	6 hours ago
Rakefile	Initial commit	6 hours ago
cocoapods-rome.gemspec	Initial commit	6 hours ago

README.md

cocoapods-rome

Rome makes it easy to build a list of frameworks for consumption outside of Xcode, e.g. for a Swift script.

Code

Issues 2

Pull requests 0

Wiki

Pulse

Graphs

SSH clone URL

git@github.com:neonichu

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

cocoapods-rome

Rome makes it easy to build a list of frameworks for consumption outside of Xcode, e.g. for a Swift script.

Installation

```
$ gem install cocoapods-rome
```

Usage

Write a simple Podfile like this:

```
platform :osx, '10.10'
use_frameworks!

plugin 'cocoapods-rome'

pod 'Alamofire'
```

then run this:

```
pod install --no-integrate --no-repo-update
```

and you will end up with dynamic frameworks:

```
$ tree Rome/
Rome/
└── Alamofire.framework
```

BIT.LY/SWIFTSCRIPTING

@AYANONAGON