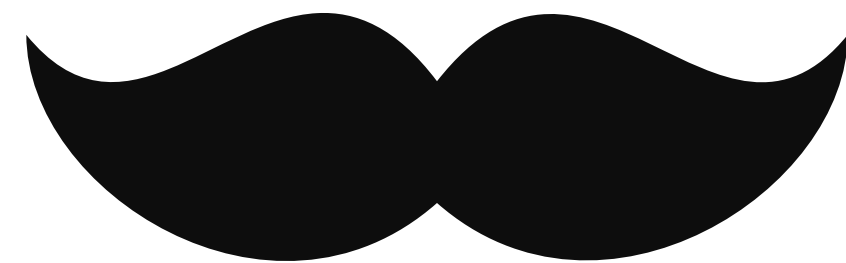
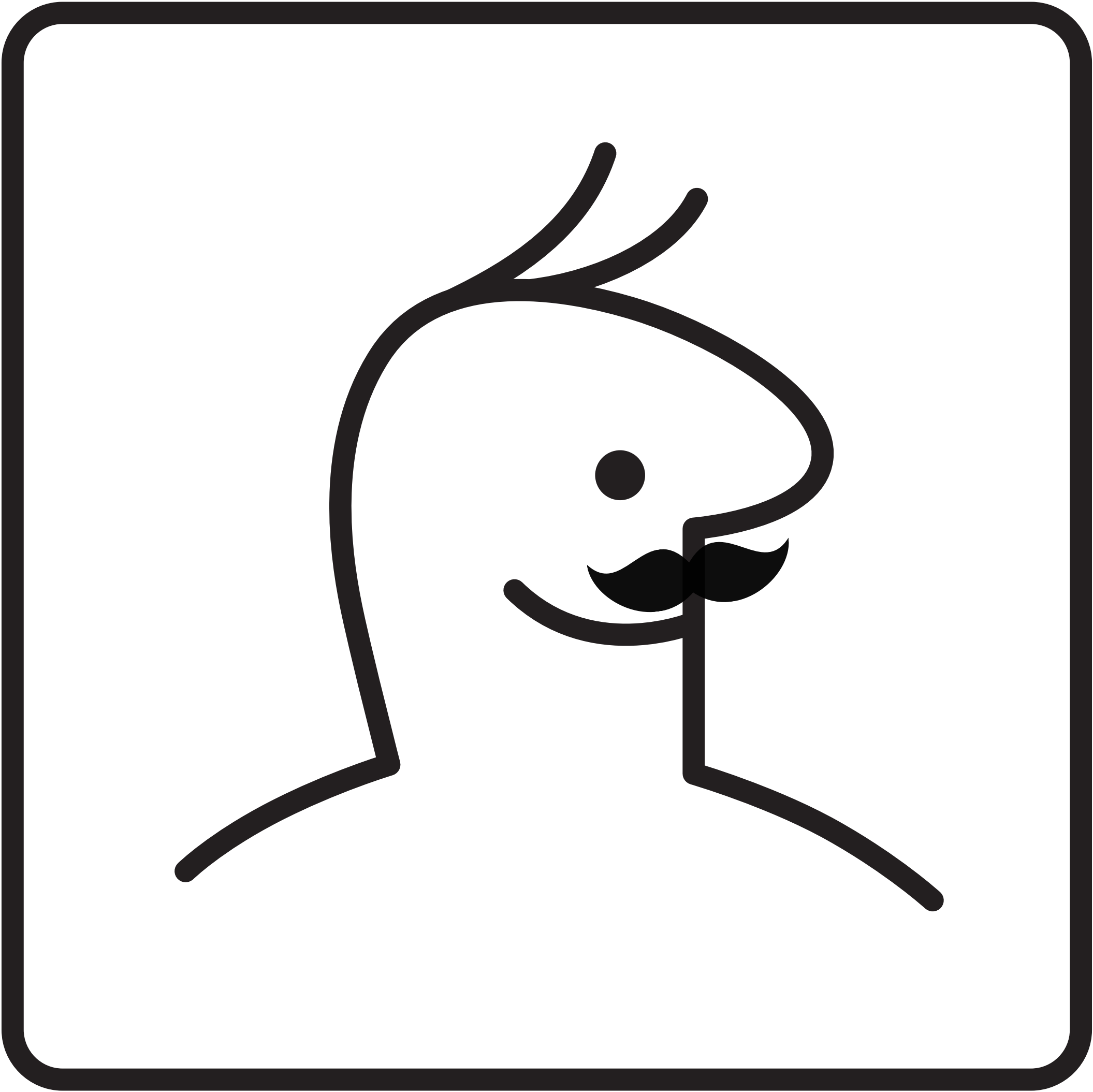
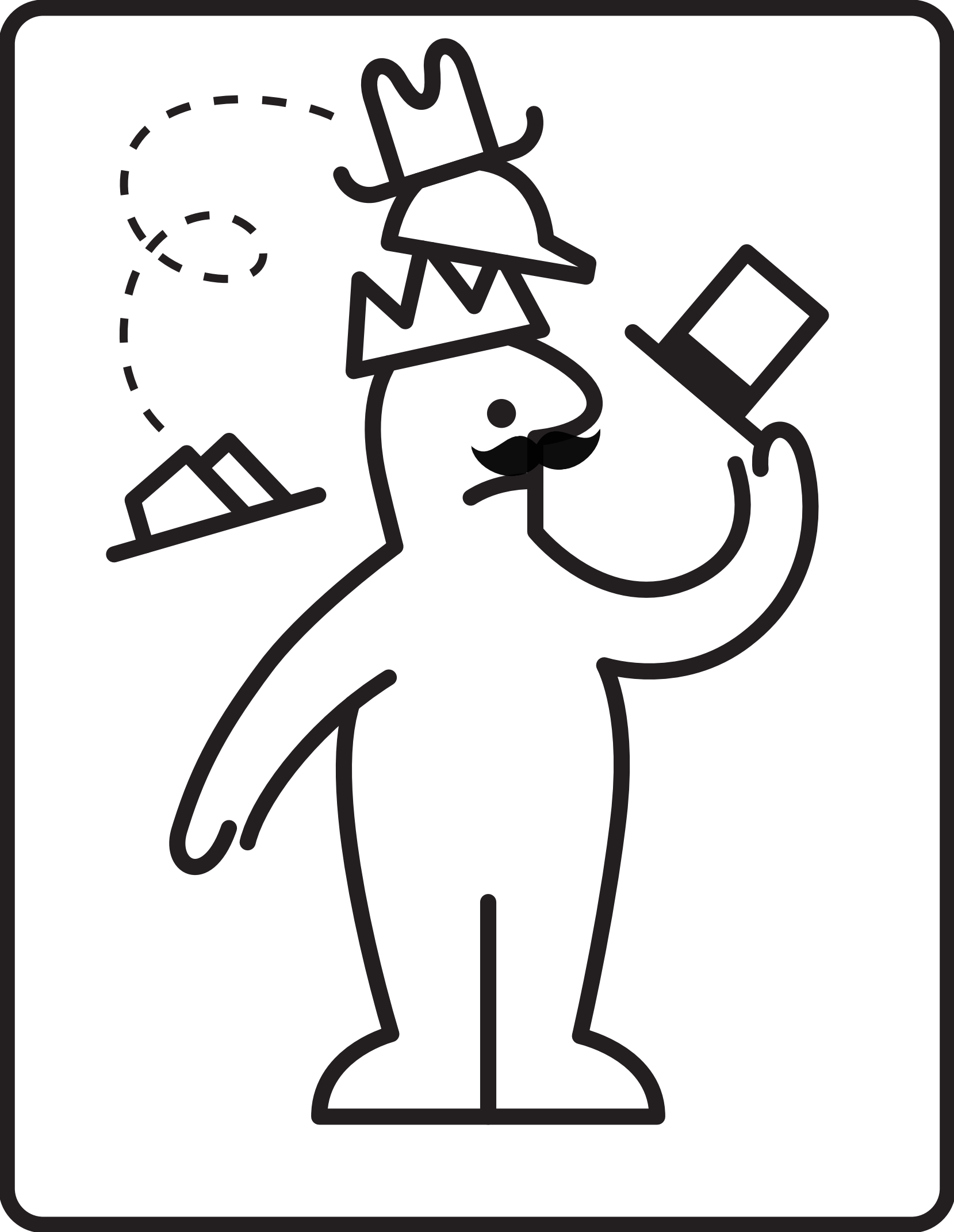


Le buone pratiche di ingegneria

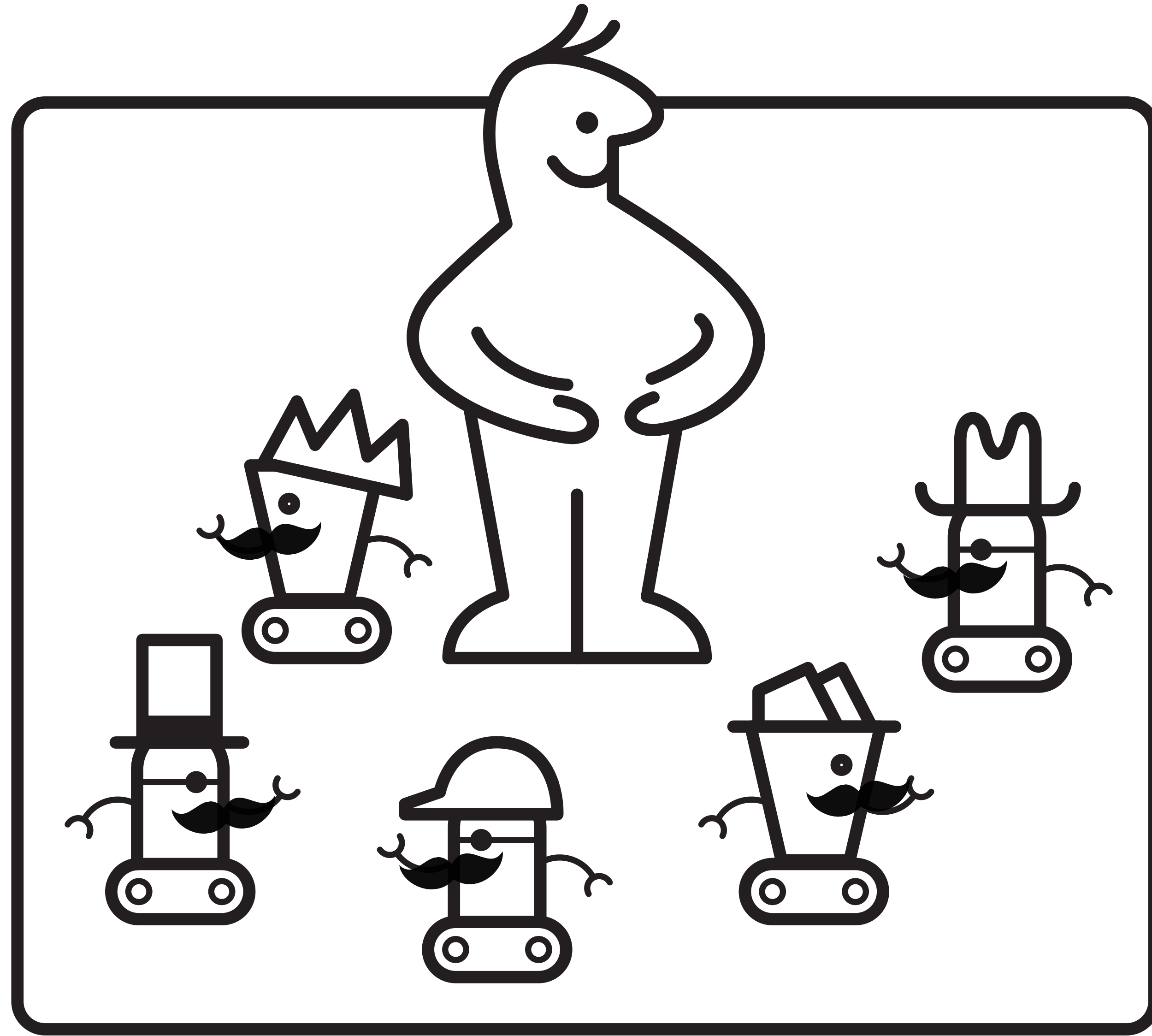


Mi fan un baffo!









swift-format

```
$ brew tap nshipster/formulae
```

```
$ brew install swift-format
```

```
$ swift-format Example.swift
```

```
    struct ShippingAddress : Codable {  
var recipient: String  
    var streetAddress : String  
    var locality :String  
    var region      :String;var postalCode:String  
    var country:String  
  
    init(        recipient: String,            streetAddress:  
String,  
locality: String,region: String,postalCode:  
String,country:String        ) { ... }
```

```
struct ShippingAddress: Codable {  
    var recipient: String  
    var streetAddress: String  
    var locality: String  
    var region: String;  
    var postalCode: String  
    var country: String  
  
    init(  
        recipient: String,  
        streetAddress: String,  
        locality: String,  
        region: String,  
        postalCode: String,  
        country: String  
    ) { ... }
```


SwiftLint

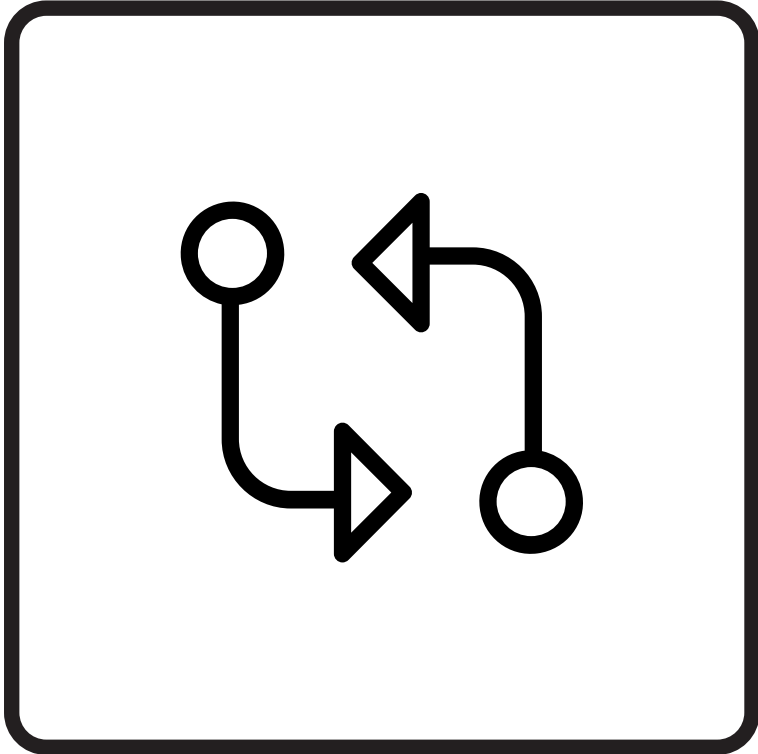
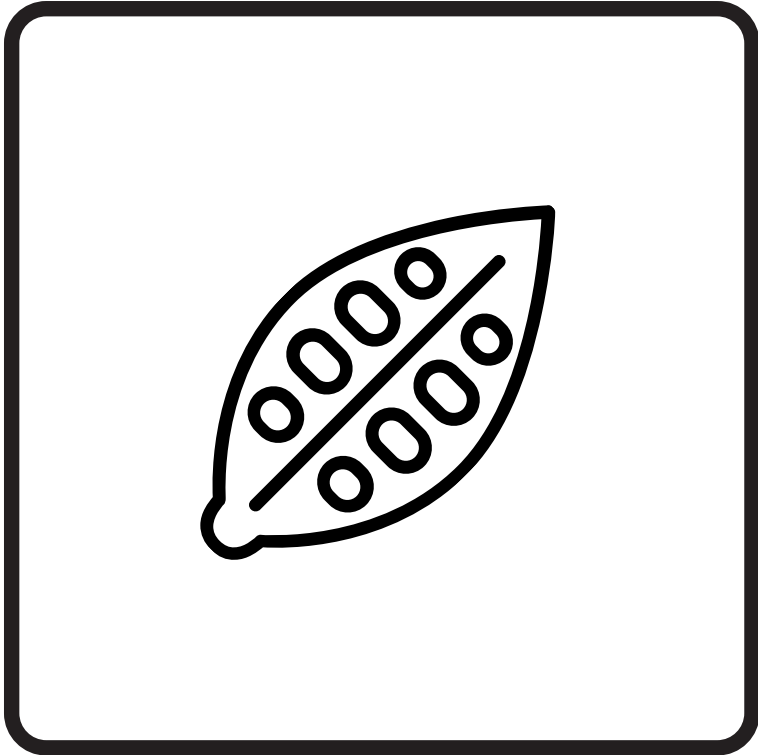
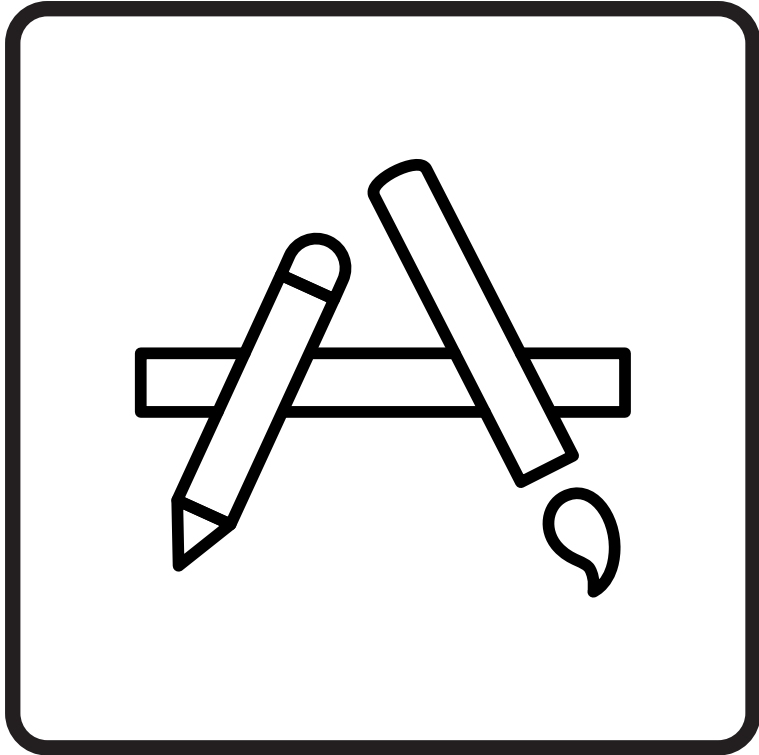
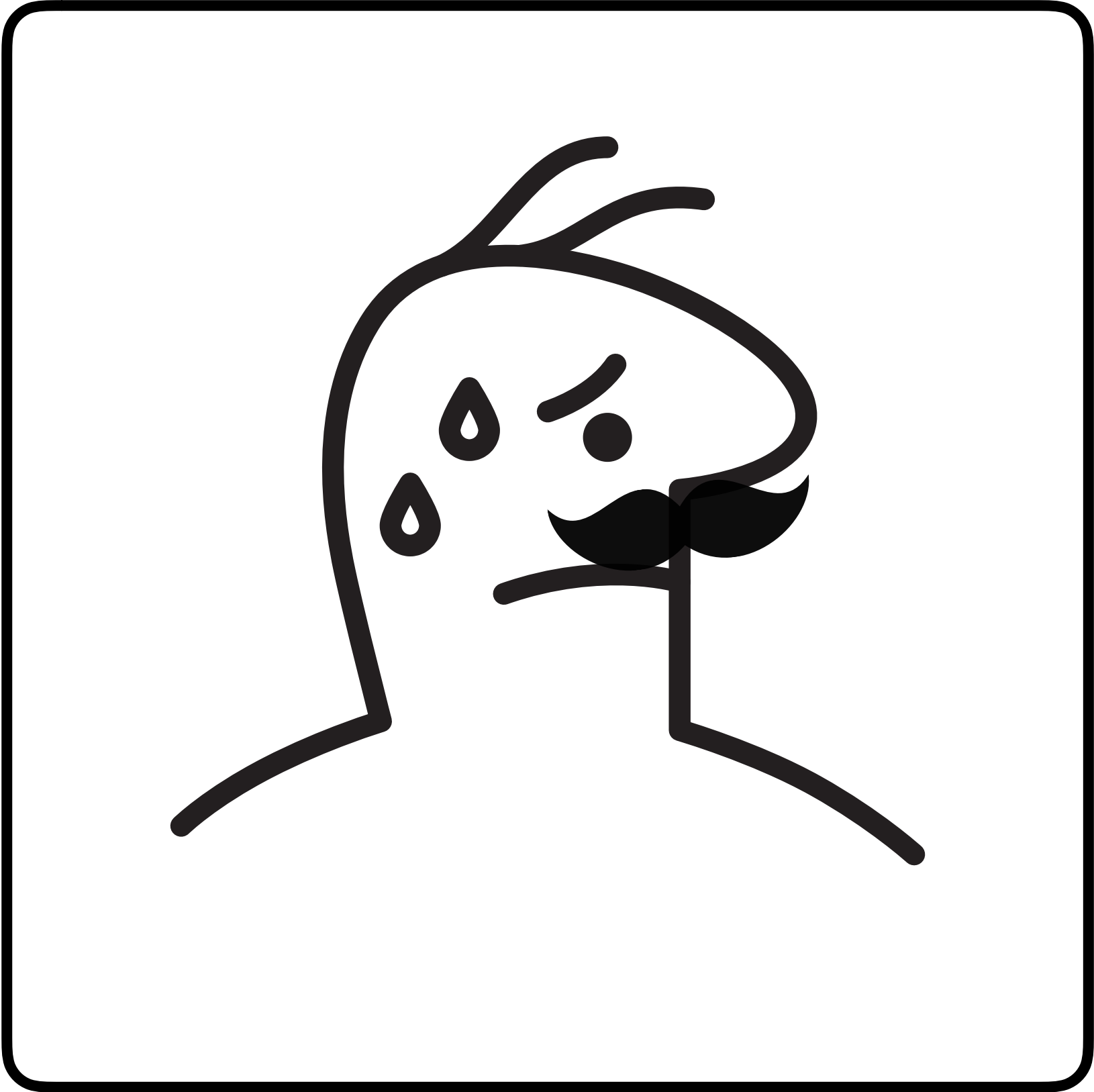
```
$ brew install swiftlint
```

```
$ swiftlint lint
```

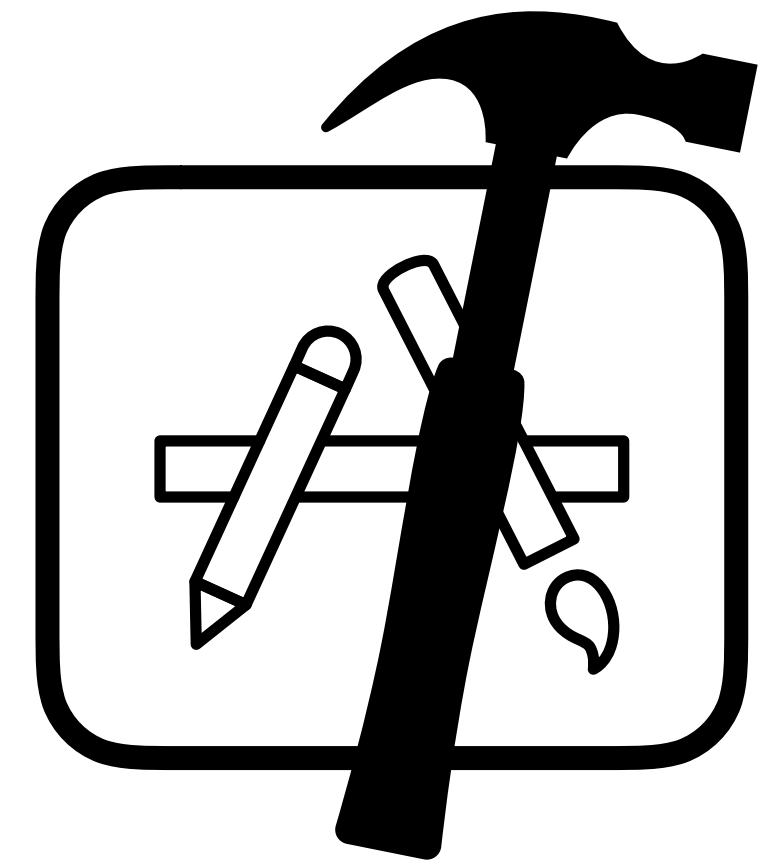
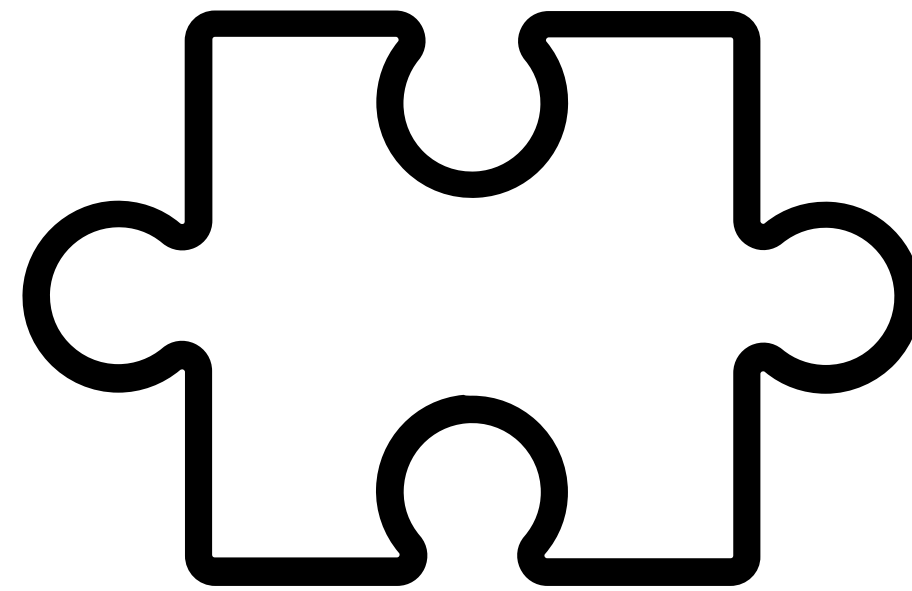
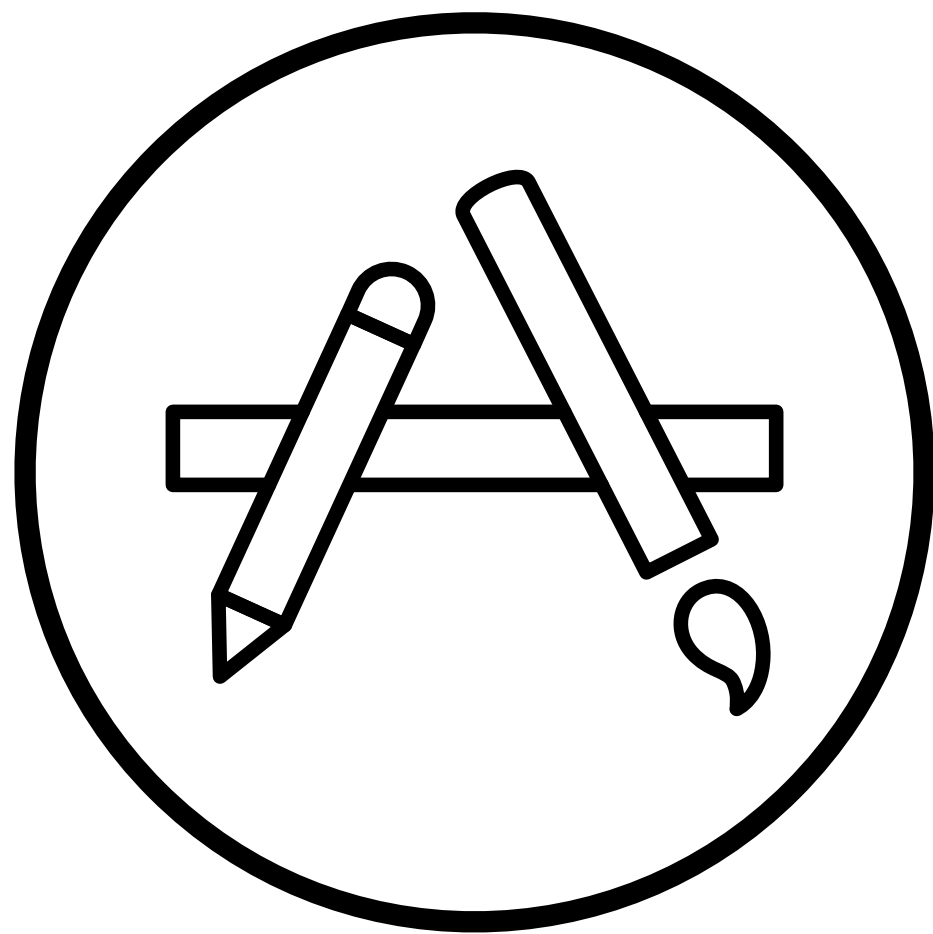
SwiftLint

```
$ swiftlint analyze \  
  --compiler-log-path \  
  /path/to/xcodebuild.log
```

- Enforce use of explicit self
- Detect unused declarations
- Detect unused imports



Xcode Source Extension



Homebrew + Build Phase

```
$ brew install swiftlint
```

▼ Run Script

Shell `/bin/sh`

```
if which swiftlint >/dev/null; then
  swiftlint
else
  echo "warning: SwiftLint not installed, \
      download from https://github.com/realm/SwiftLint"
fi
```

CocoaPods + Build Phase

```
$ echo "pod 'SwiftLint'" >> Podfile
```

```
$ pod install
```

▼ Run Script

Shell `/bin/sh`

```
if which swiftlint >/dev/null; then
  swiftlint
else
  echo "warning: SwiftLint not installed, \
      download from https://github.com/realm/SwiftLint"
fi
```

Homebrew + Git Hook

```
$ brew install swiftlint
```

```
$ mv script.sh .git/hooks/pre-push
```

```
$ chmod +x .git/hooks/pre-push
```

Continuous Integration

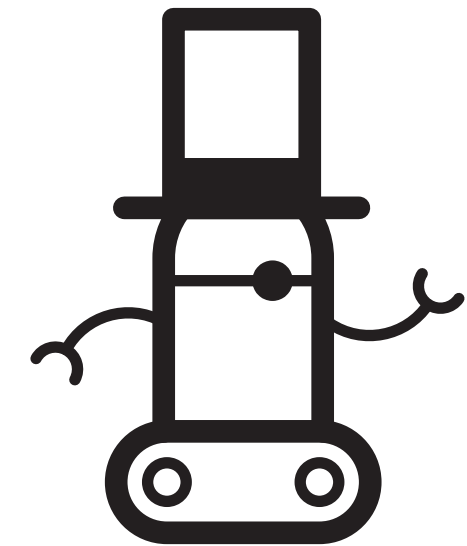


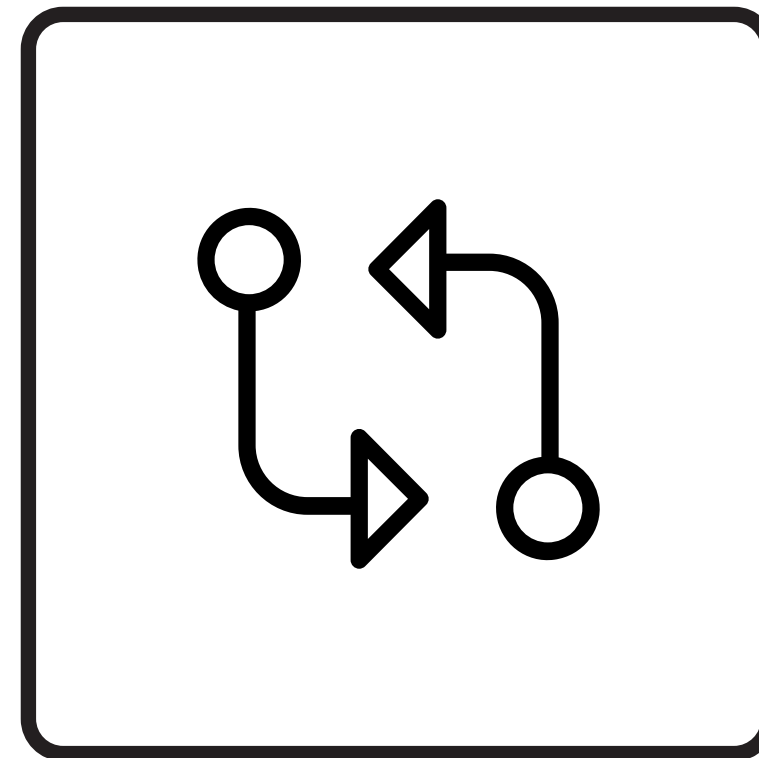
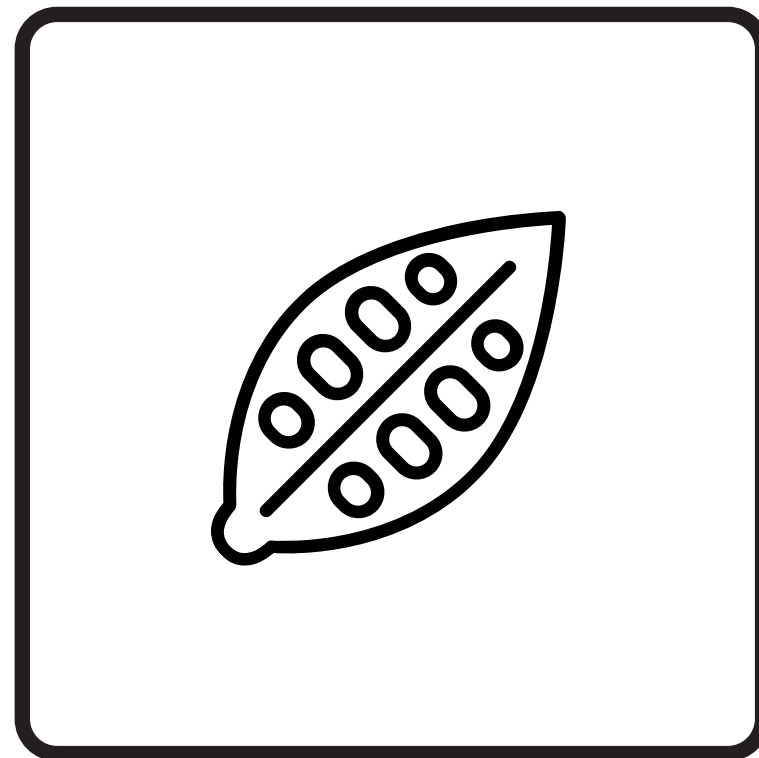
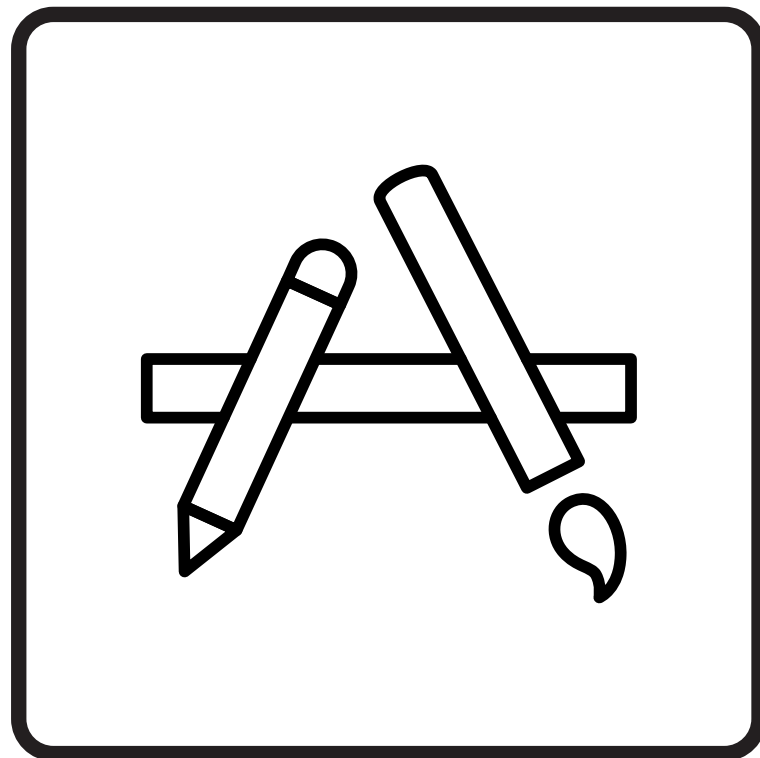
```
commit 0cecf1bfccc841ef06d510fdd87c5a0405481b95
Author: Some Developer <dev@company.biz>
Date:   Fri Sep 27 13:15:22 2019 -0700
```

Add code

```
Mustache/AppDelegate.swift:16:1:
warning: Line Length Violation:
Line should be 120 characters or less:
currently 145 characters (line_length)
```

...





Brewfile + Build Phases

```
tap 'nshipster/formulae'  
tap 'IBDecodable/tap'
```

```
brew 'swift-format'  
brew 'swiftlint'  
brew 'iblinter'
```

▼ Run Script

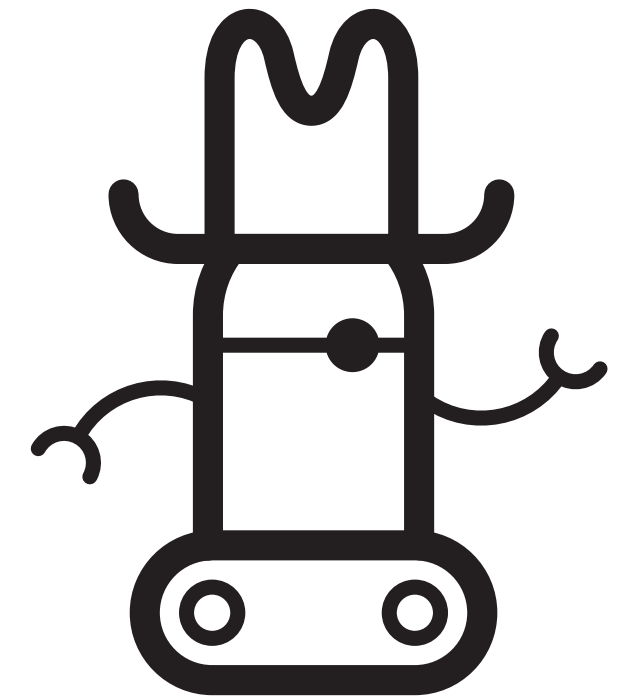
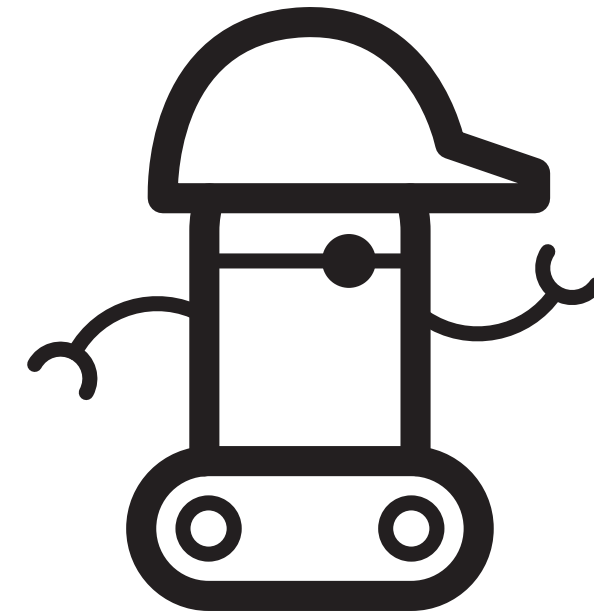
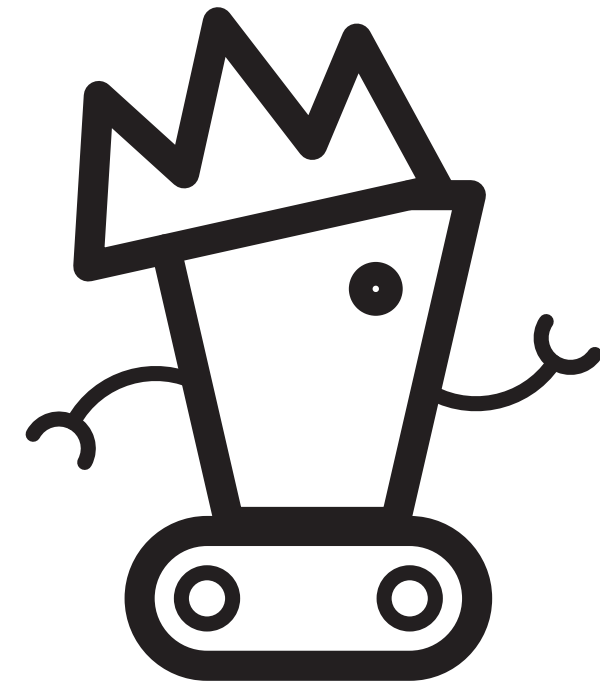
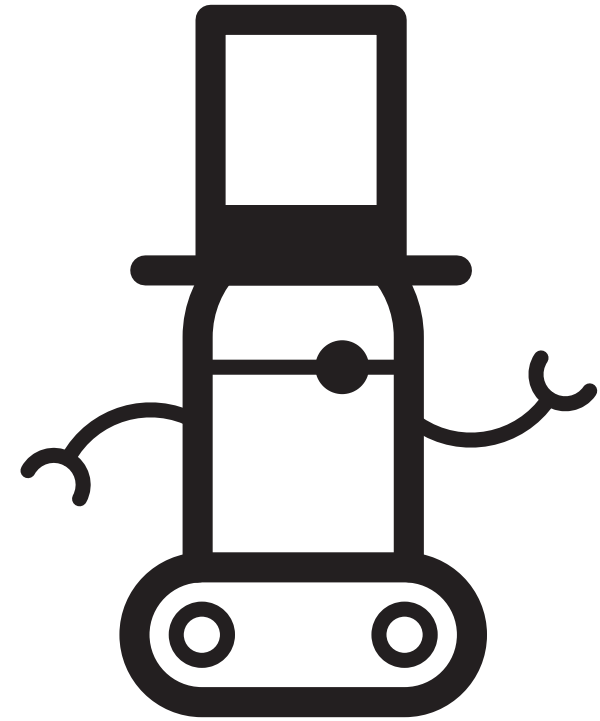
Shell `/bin/sh`

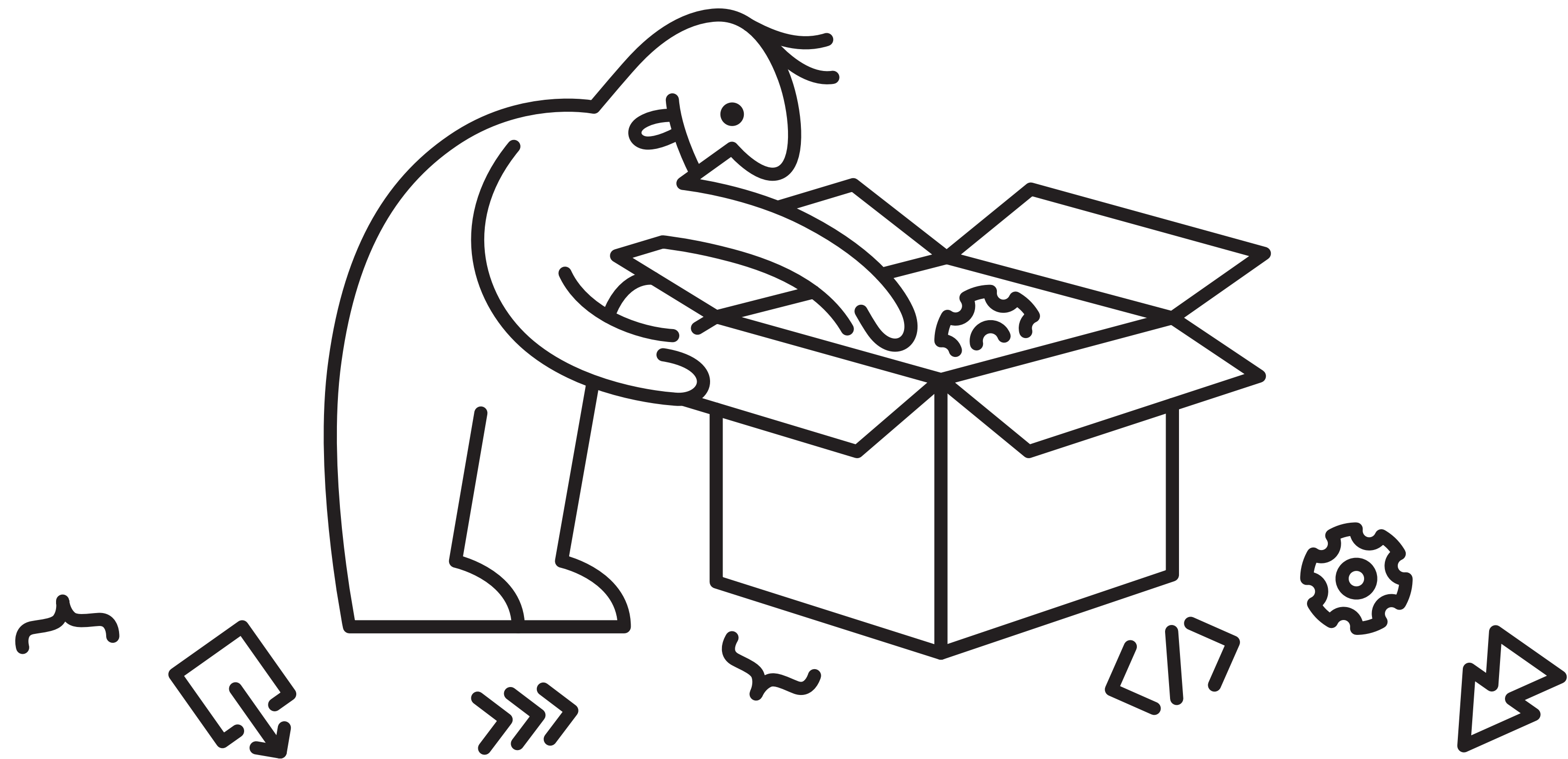
```
if which brew >/dev/null; then  
  brew bundle  
else  
  echo "warning: Homebrew not installed"  
fi
```

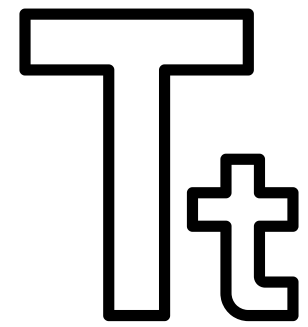
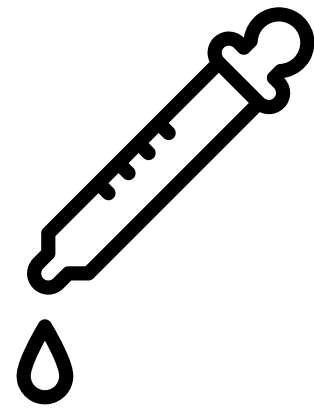
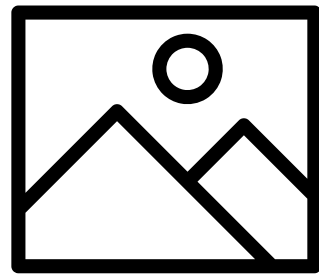
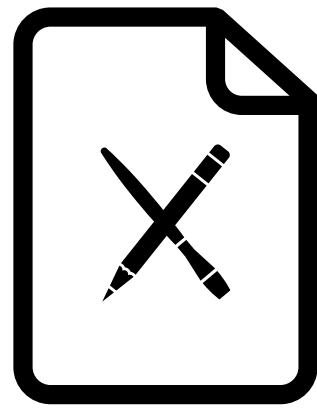
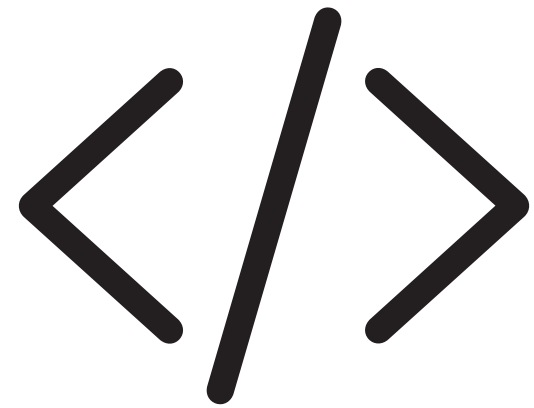
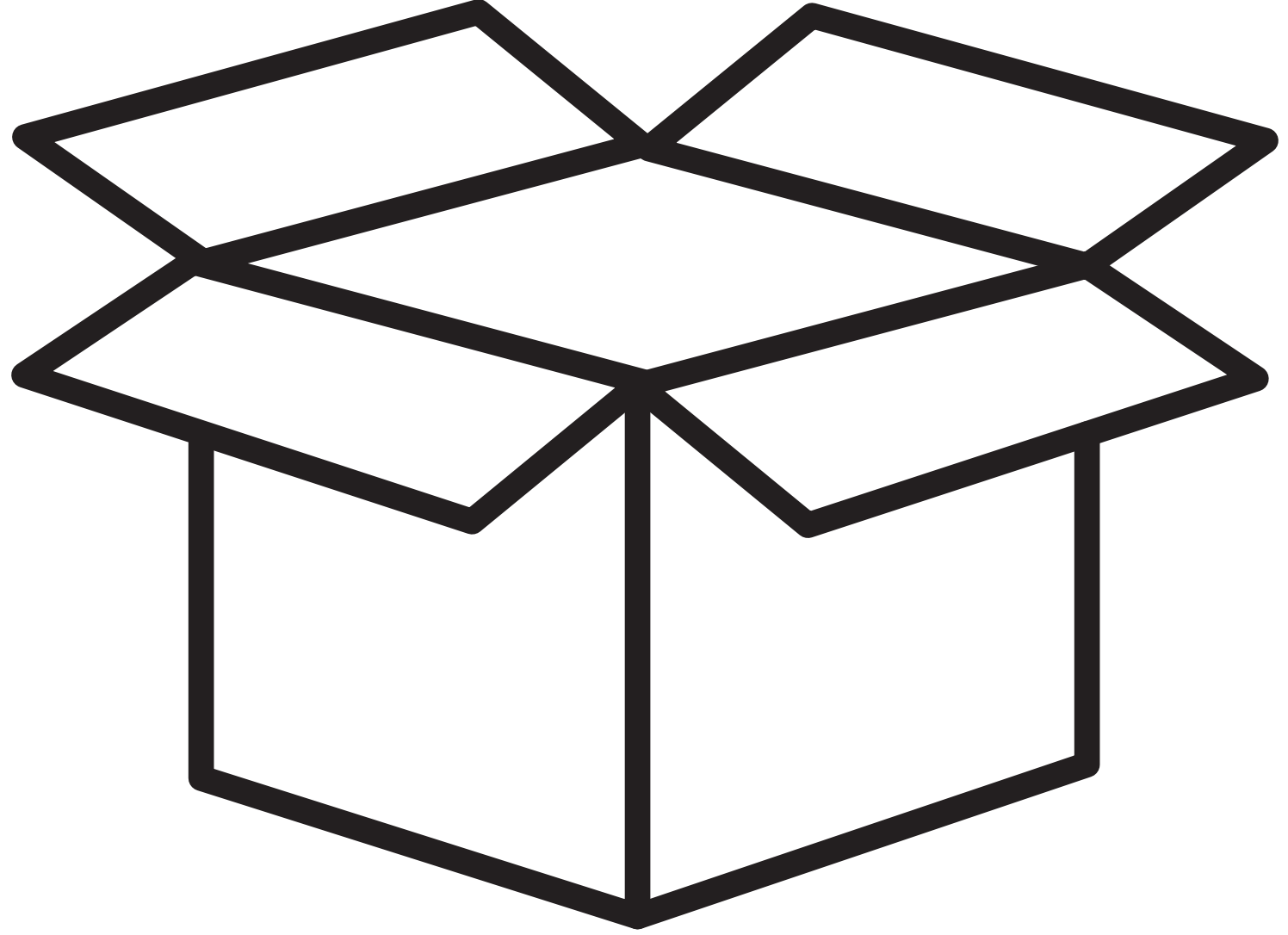
▼ Run Script

Shell `/bin/sh`

```
if which swiftlint >/dev/null; then  
  swiftlint  
else  
  echo "warning: SwiftLint not installed"  
fi
```





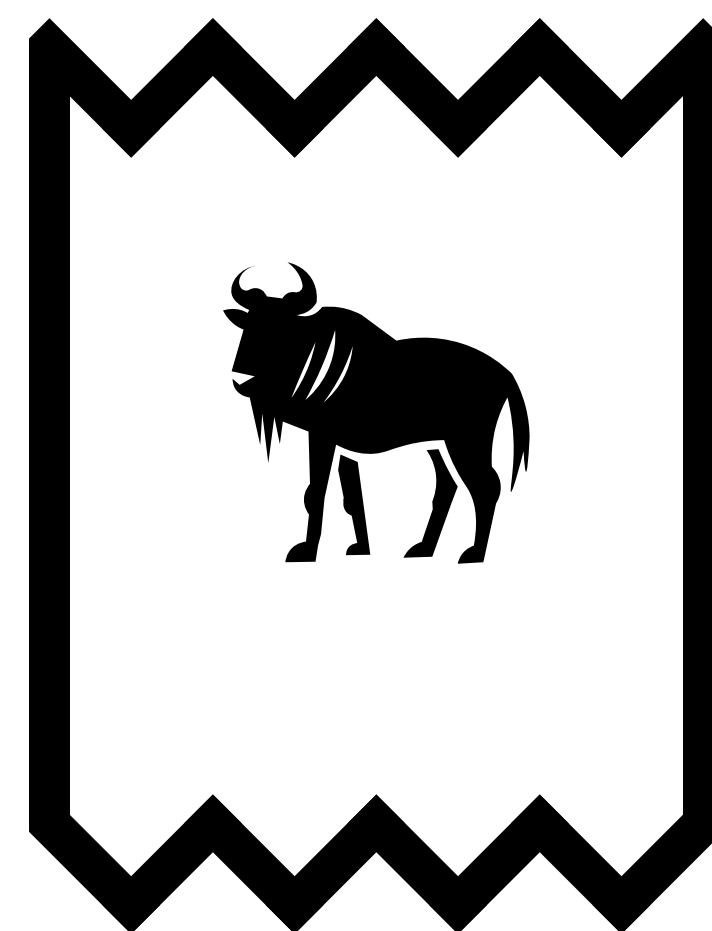
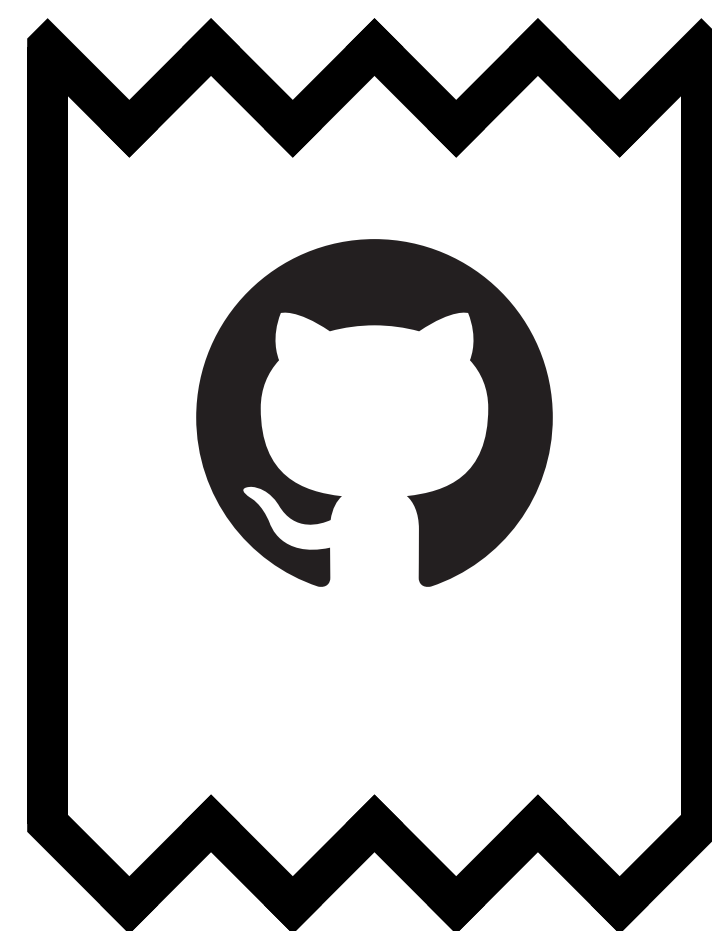
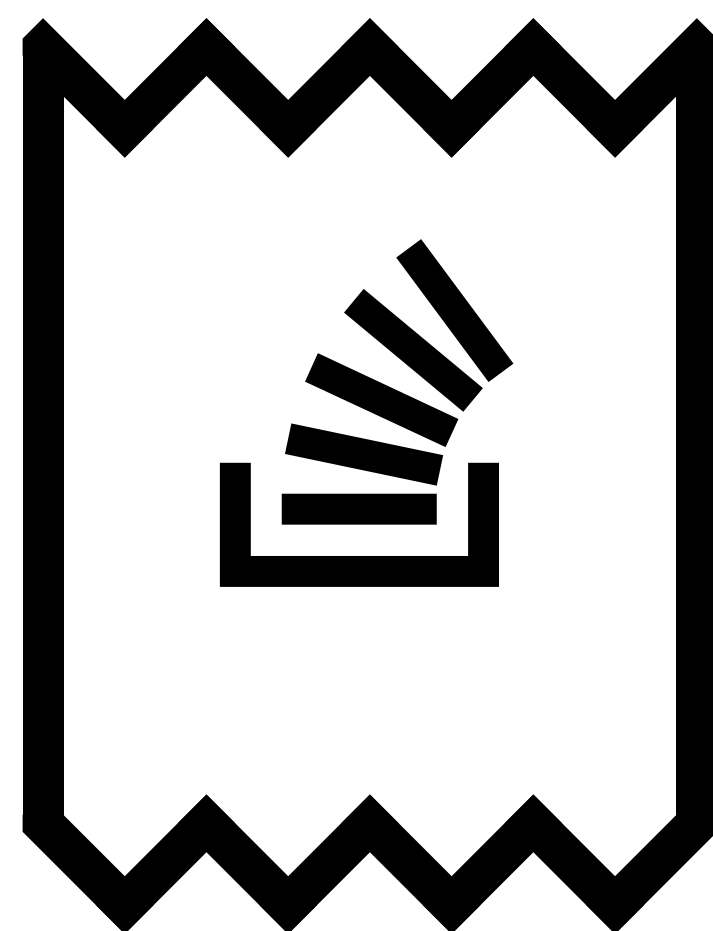
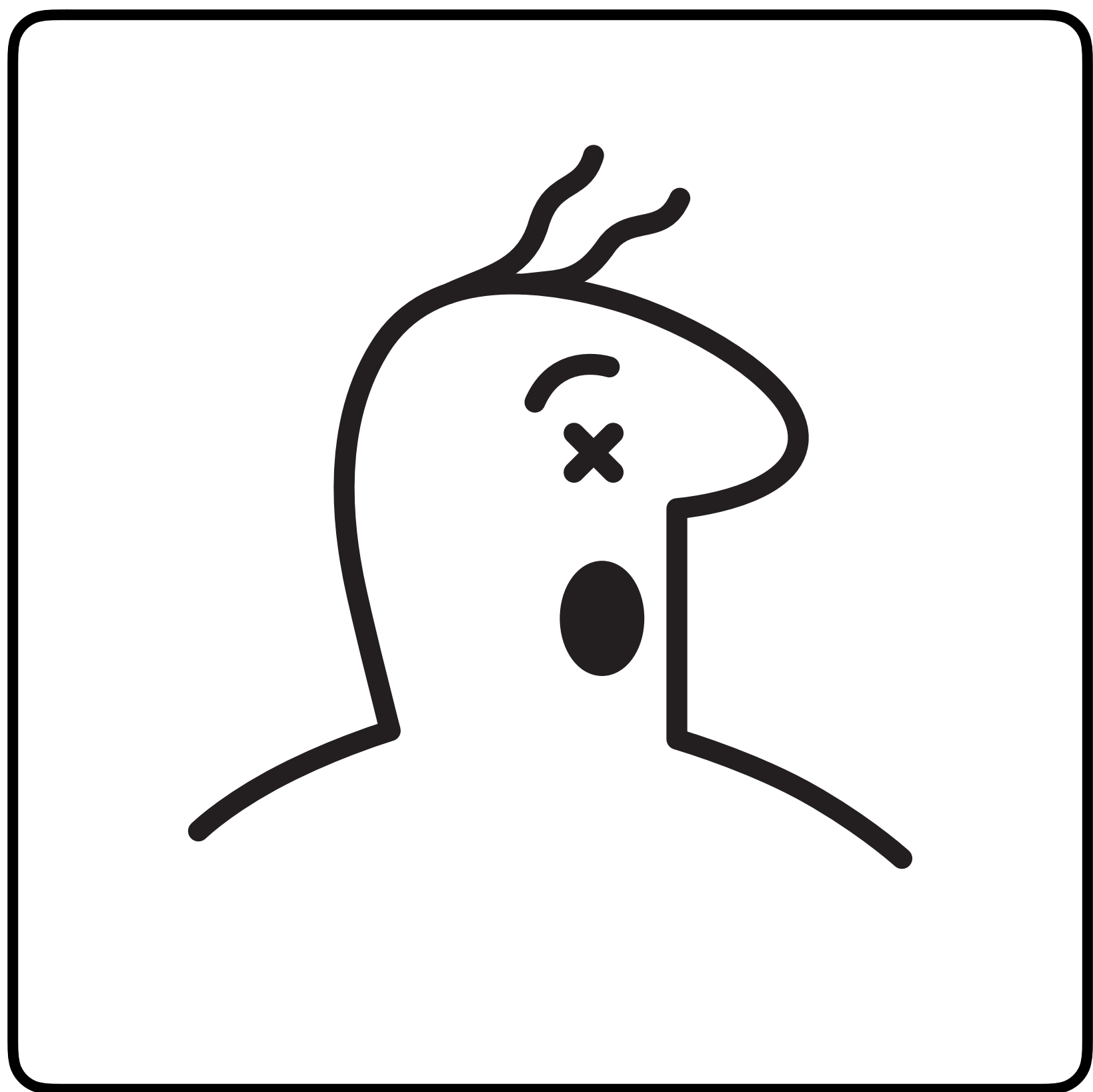


`</>` Code

scancode

```
$ pip install scancode-toolkit
```

```
$ scancode -c -e -u Sources/ \  
    --html results.html
```



periphery

```
$ brew tap peripheryapp/periphery
```

```
$ brew cask install periphery
```

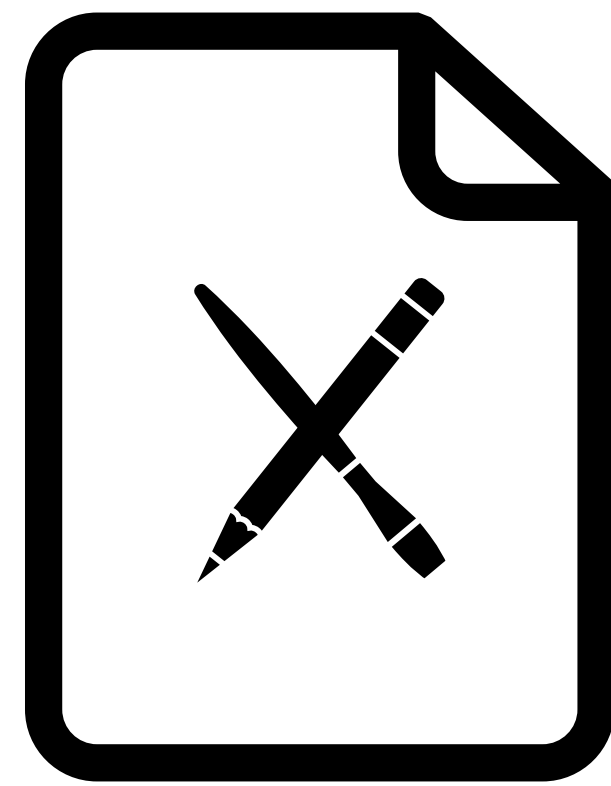
```
$ periphery scan \
```

```
  --workspace MyApp.xcworkspace \
```

```
  --schemes MyApp \
```

```
  --targets MyApp, MyFramework \
```

```
  --format xcode
```



XIBs & Storyboards

ibtool

```
$ find . -type f -iname "*.xib" -exec \  
    xcrun ibtool --notices \  
        --warnings \  
        --errors \  
        --output-format \  
        human-readable-text \  
    {} \;
```

ibtool

Example.xib:SQT-bD-ue5:

warning: Automatically Adjusts Font
requires using a
Dynamic Type text style

xiblint

```
$ pip install xiblint
```

```
$ xiblint
```

- Accessibility Labels
- Ambiguous / Misplaced Views
- Named Colors
- Font Families and Sizes

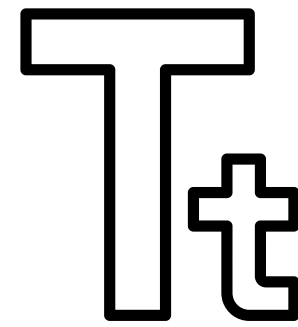
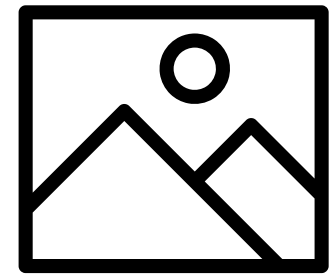
IBLint

```
$ brew tap IBDecodable/tap
```

```
$ brew install iblinter
```

```
$ iblinter
```

- Auto Layout Constraints
- Storyboard IDs
- Custom Classes



Images, Fonts, and Sounds

xcassetlint

New

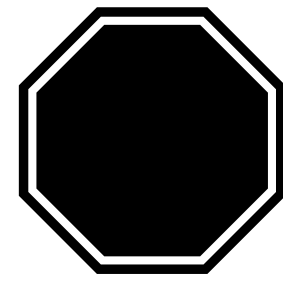
```
$ brew tap nshipster/formulae
```

```
$ brew install xcassetlint
```

```
$ xcassetlint
```


xcassetlint

New



1 Missing Asset

`"avatar.default"`



2 Possibly Unused Assets

`"chevron.right"`

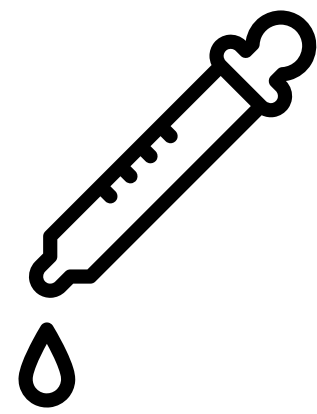
`"badge.fill"`

xcassetcop

```
$ gem install xcassetcop
```

```
$ xcassetcop lint
```

```
---  
- paths:  
  - "../MyApp/Images.xcassets"  
  config:  
    template_rendering_intent: template  
    image_scale: single  
    same_file_and_asset_name: true  
    file_extension: pdf
```



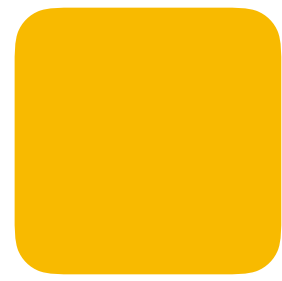
Colors



Red



Pink



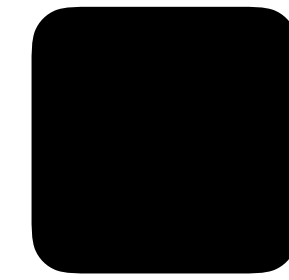
Orange



Green



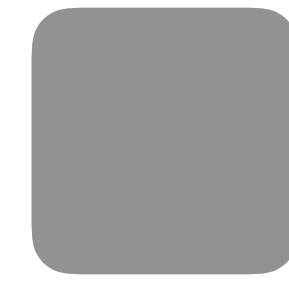
Blue



Black



Dark Gray



Gray



Light Gray



White



ibcolortool

New

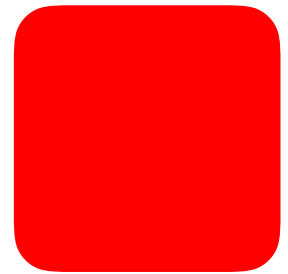
```
$ brew tap nshipster/formulae
```

```
$ brew install ibcolortool
```

```
$ ibcolortool list **/*.{xib,storyboard} \  
    --format rgbhex
```

ibcolortool

New



#EE220C

UIColor(red:1, green:0, blue:0, alpha:1)



#EE503E

UIColor(red:0.931, green:0.312, blue:0.244, alpha:1)



#FF3200

UIColor(red:1, green:0.196, blue:0, alpha:1)



#EF5FA7








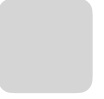


UIColor(red:0.937, green:0.372, blue:0.656, alpha:1)



#F8BA00

UIColor(red:0.974, green:0.73, blue:0, alpha:1)

...

```
let palette: [CGColor] = [           ]  
let colors: [CGColor]
```

```
let palettizedColors = Dictionary(grouping: colors) { color in  
    palette.map { (color: $0, distance: color.distance(to: $0)) }  
        .min(by: { $0.distance < $1.distance })?  
        .color  
}
```



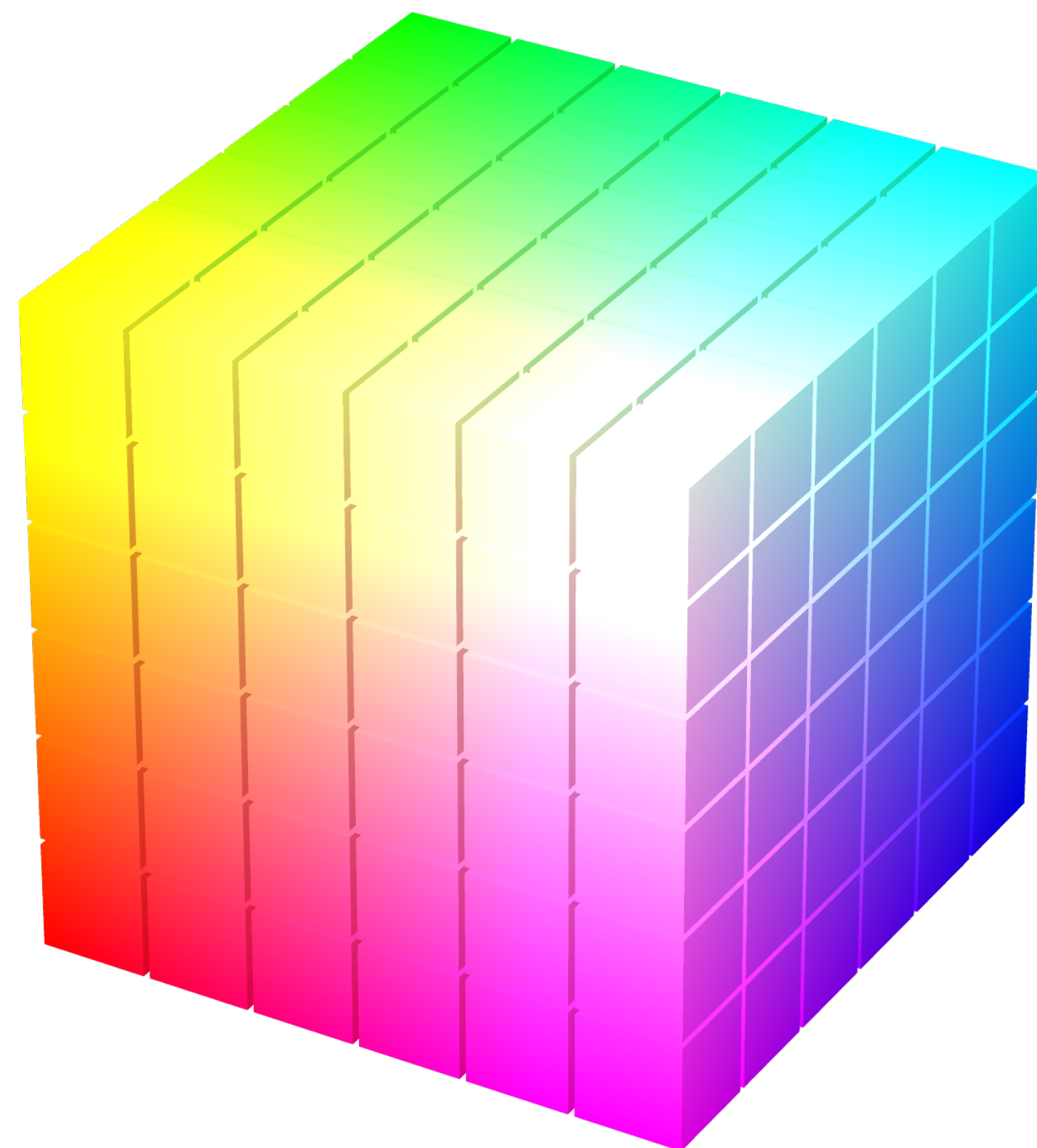
```
import CoreGraphics

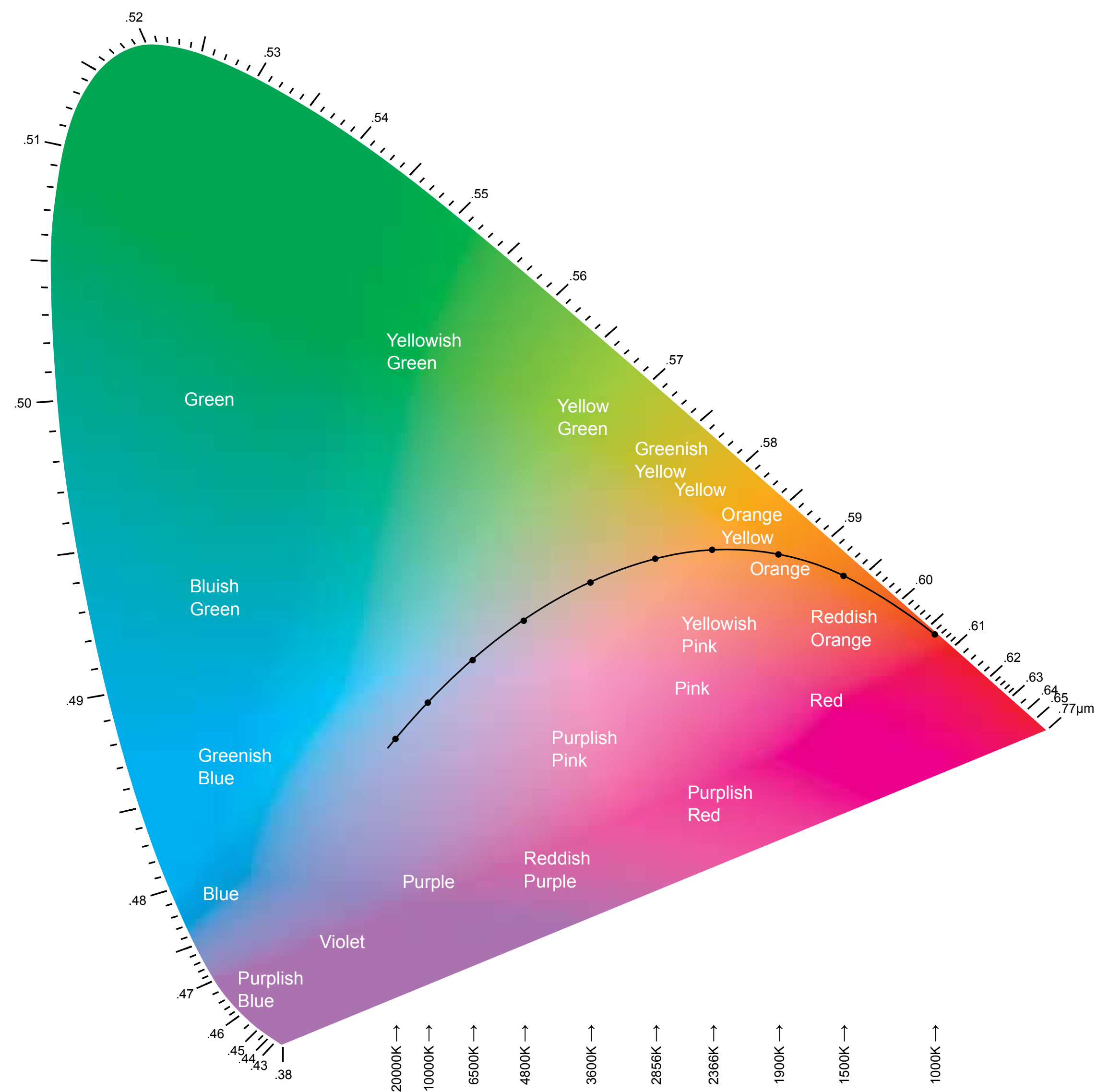
extension CGColor {
    func distance(to color: CGColor) -> CGFloat? {
        guard self != color else { return 0 }

        let deviceRGB = CGColorSpaceCreateDeviceRGB()
        guard let from = self.converted(to: deviceRGB,
                                         intent: .defaultIntent,
                                         options: nil)?.components,
              let to = color.converted(to: deviceRGB,
                                       intent: .defaultIntent,
                                       options: nil)?.components
        else {
            return nil
        }

        return zip(from, to).map(-).map { $0 * $0 }.reduce(0, +).squareRoot()
    }
}
```

<digression>



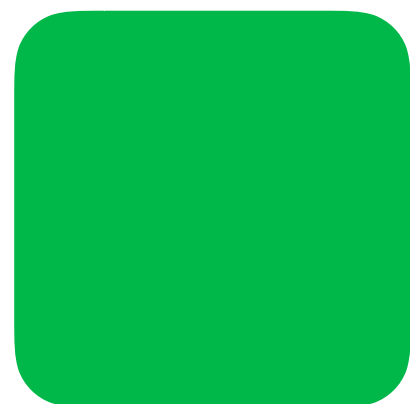
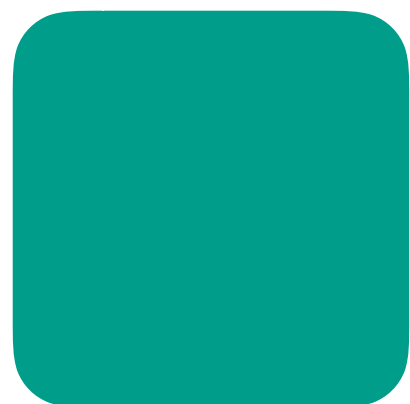
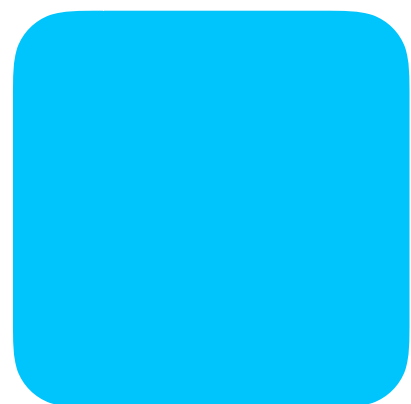


CMC I:c (1984)

$$\Delta E_{CMC}^* = \sqrt{\left(\frac{L_2^* - L_1^*}{lS_L}\right)^2 + \left(\frac{C_2^* - C_1^*}{cS_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{S_H}\right)^2}$$

$$S_L = \begin{cases} 0.511 & L_1^* < 16 \\ \frac{0.040975L_1^*}{1+0.01765L_1^*} & L_1^* \geq 16 \end{cases} \quad S_C = \frac{0.0638C_1^*}{1 + 0.0131C_1^*} + 0.638 \quad S_H = S_C(FT+1-F)$$

$$F = \sqrt{\frac{C_1^{*4}}{C_1^{*4} + 1900}} \quad T = \begin{cases} 0.56 + |0.2 \cos(h_1 + 168^\circ)| & 164^\circ \leq h_1 \leq 345^\circ \\ 0.36 + |0.4 \cos(h_1 + 35^\circ)| & \text{otherwise} \end{cases}$$



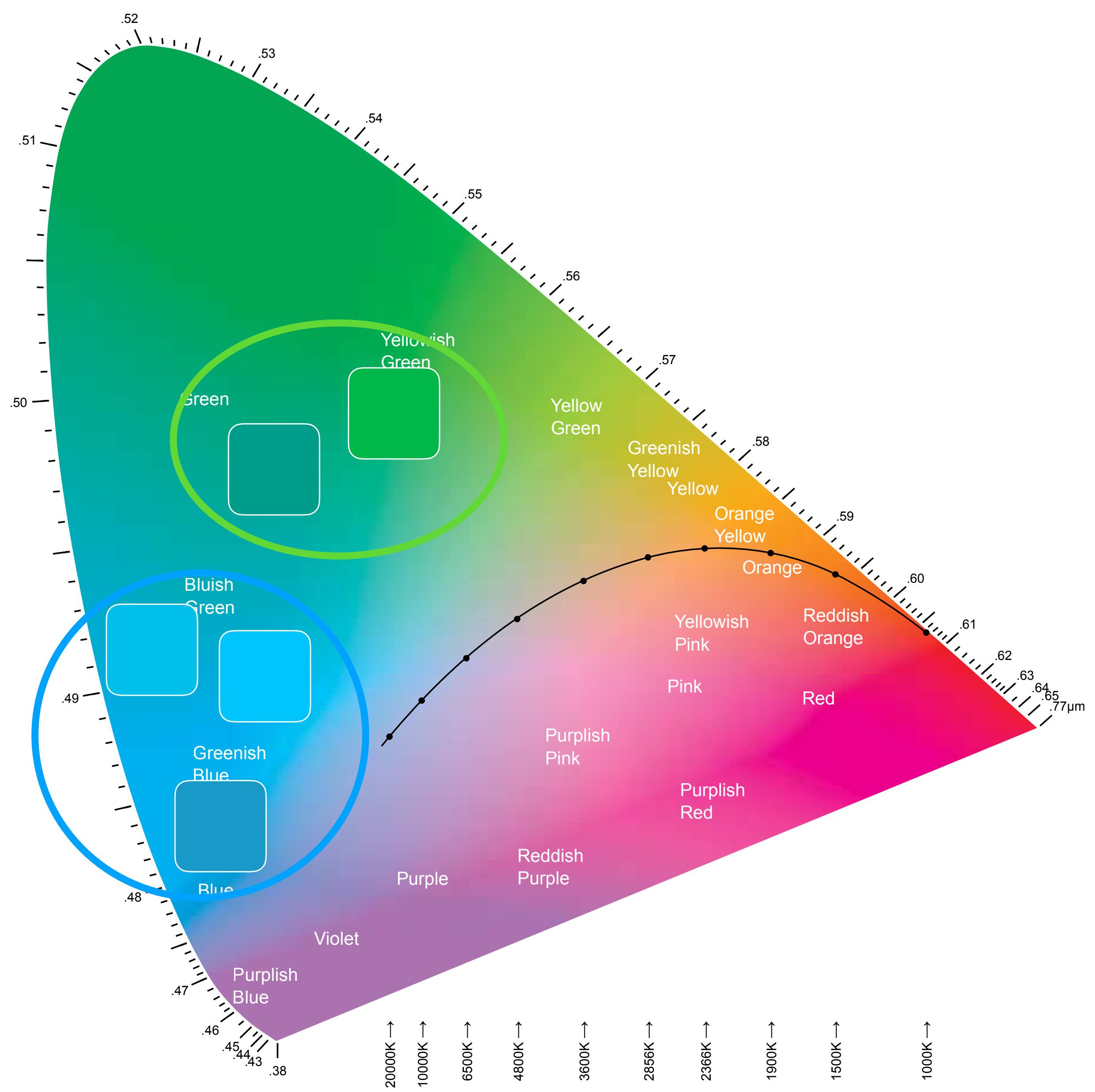
DBSCAN

New

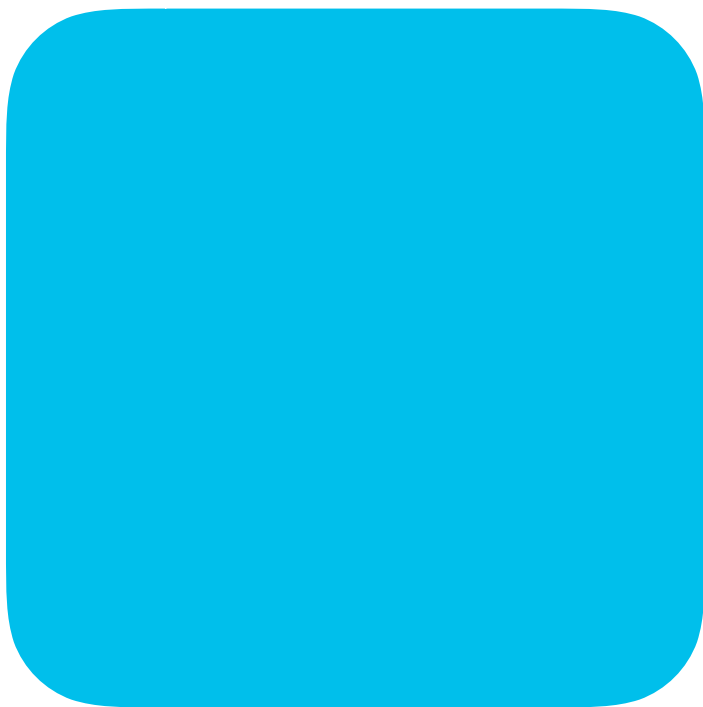
```
func euclideanDistance(lhs: [Double], rhs: [Double]) -> Double {
    precondition(lhs.count == rhs.count)
    zip(lhs, rhs).map(-).map { $0 * $0 }.reduce(0, +).squareRoot()
}
```

```
let input: [[Double]] = [ ... ]  
let dbscan = DBSCAN(input)
```

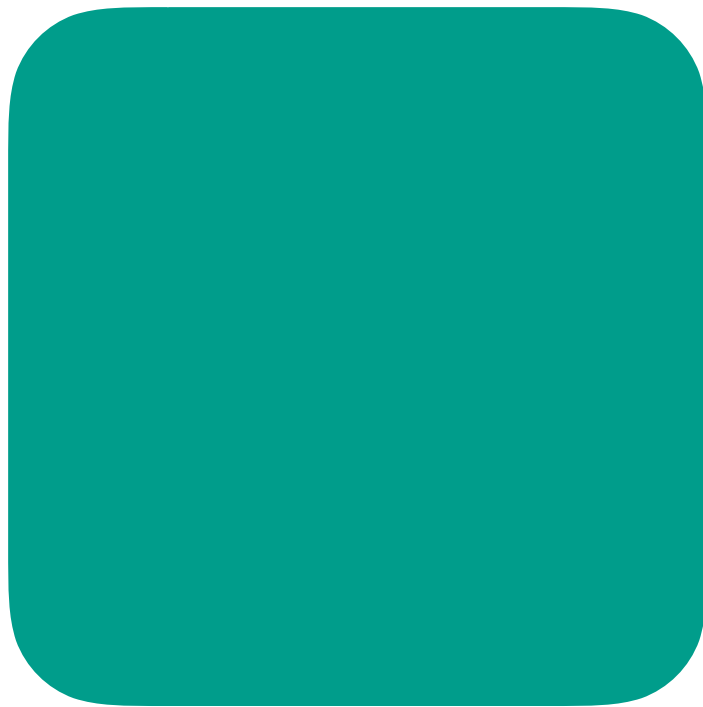
[illegible]

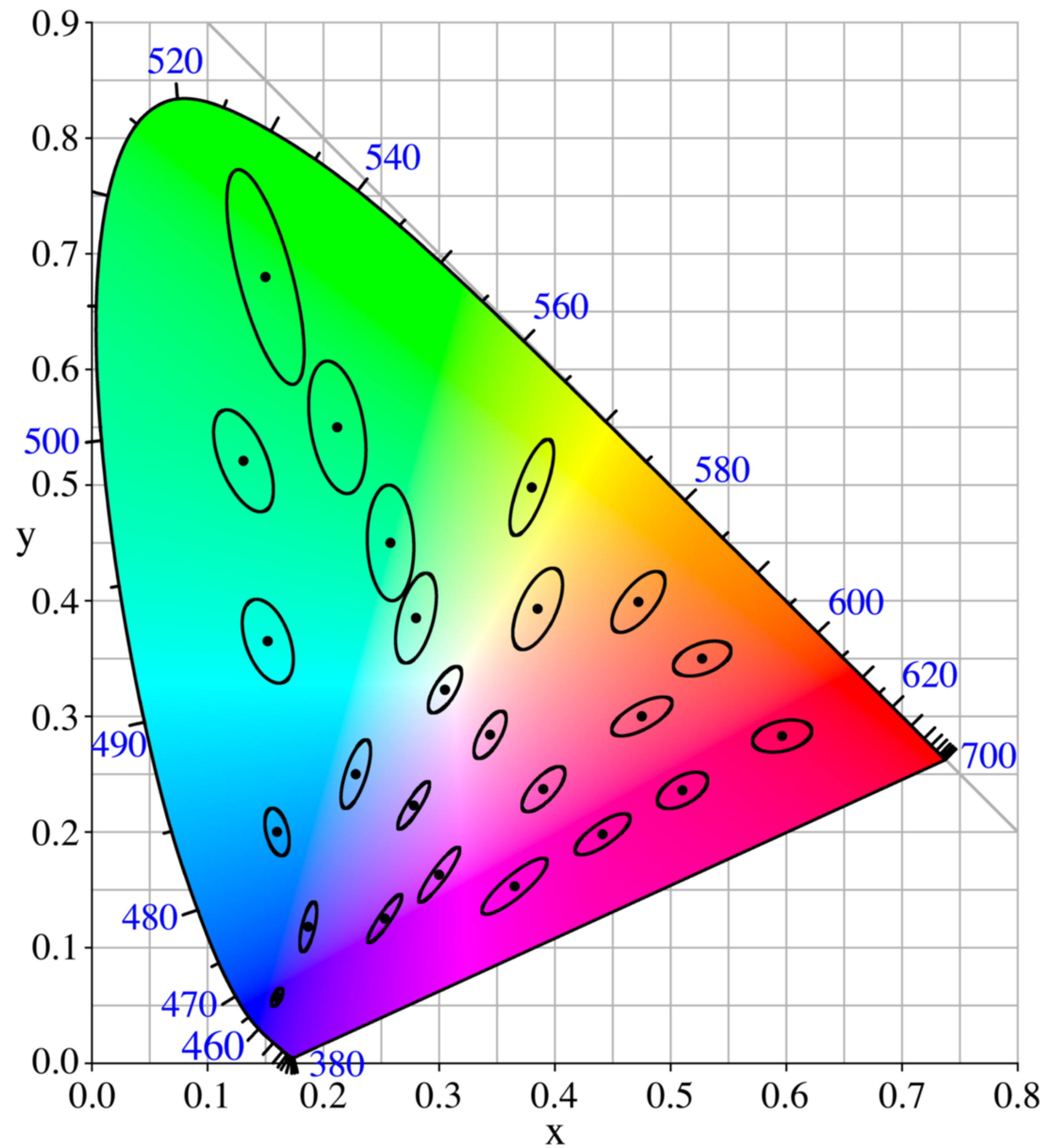


Cluster 1

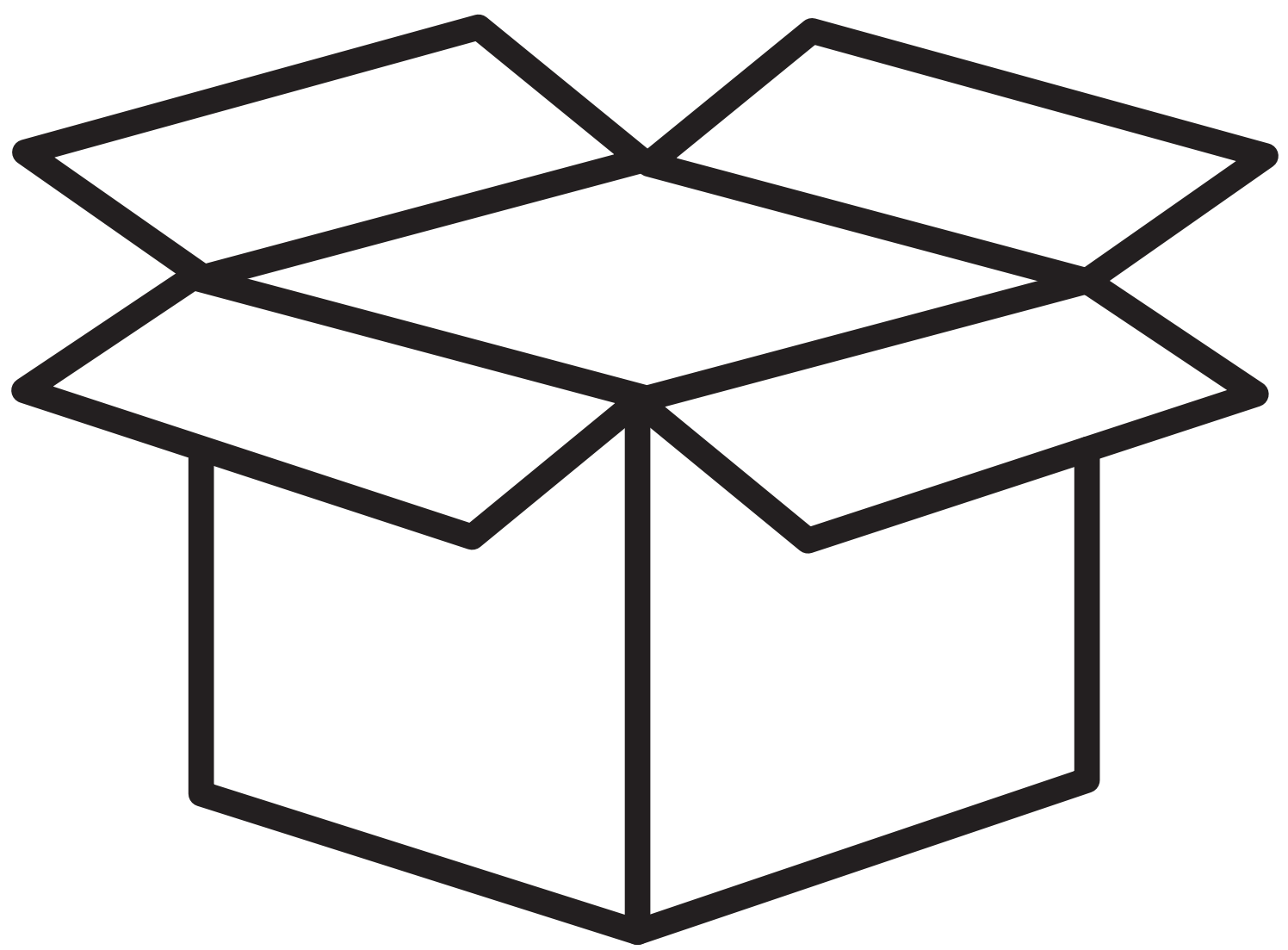


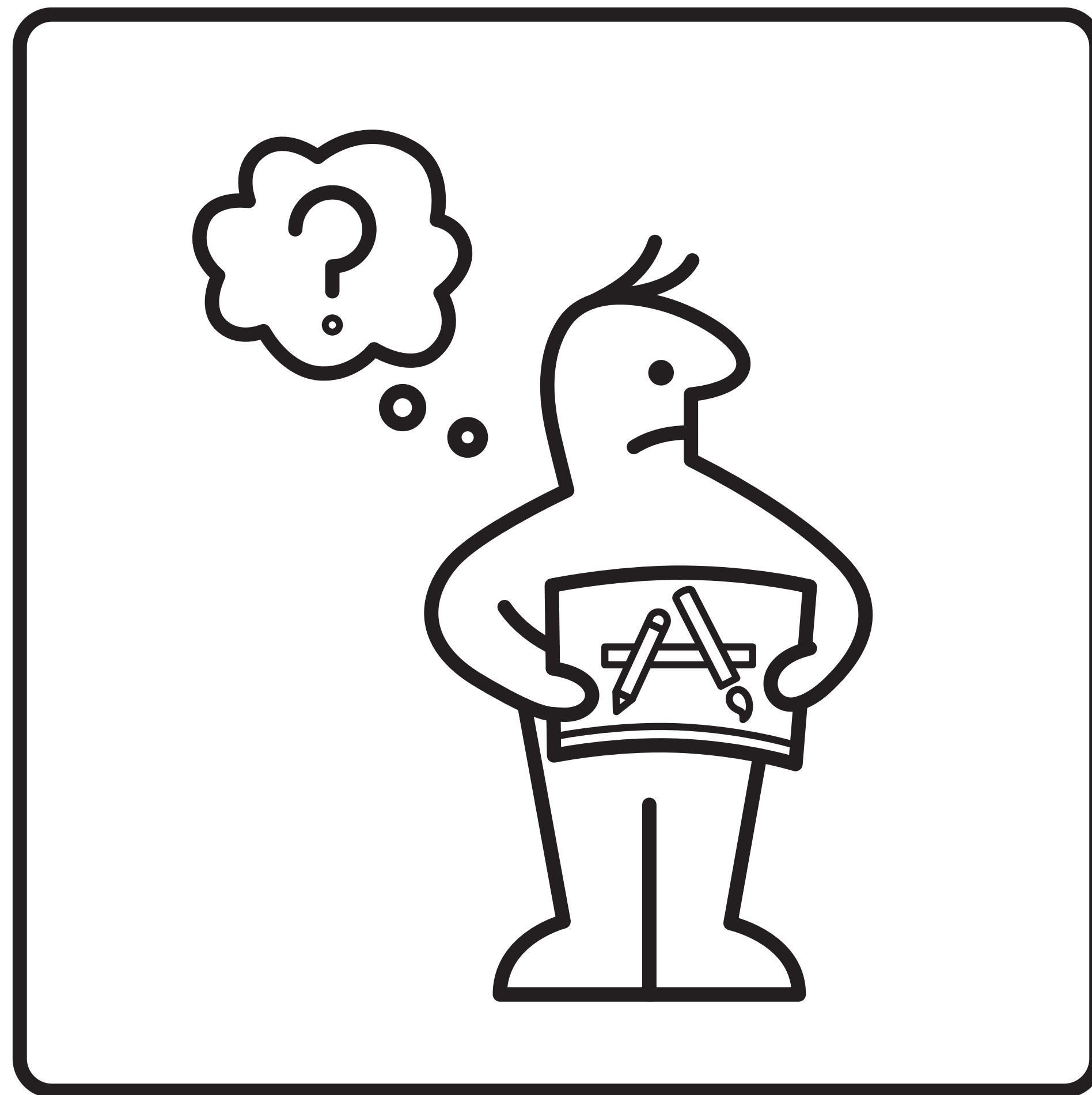
Cluster 2





</digression>



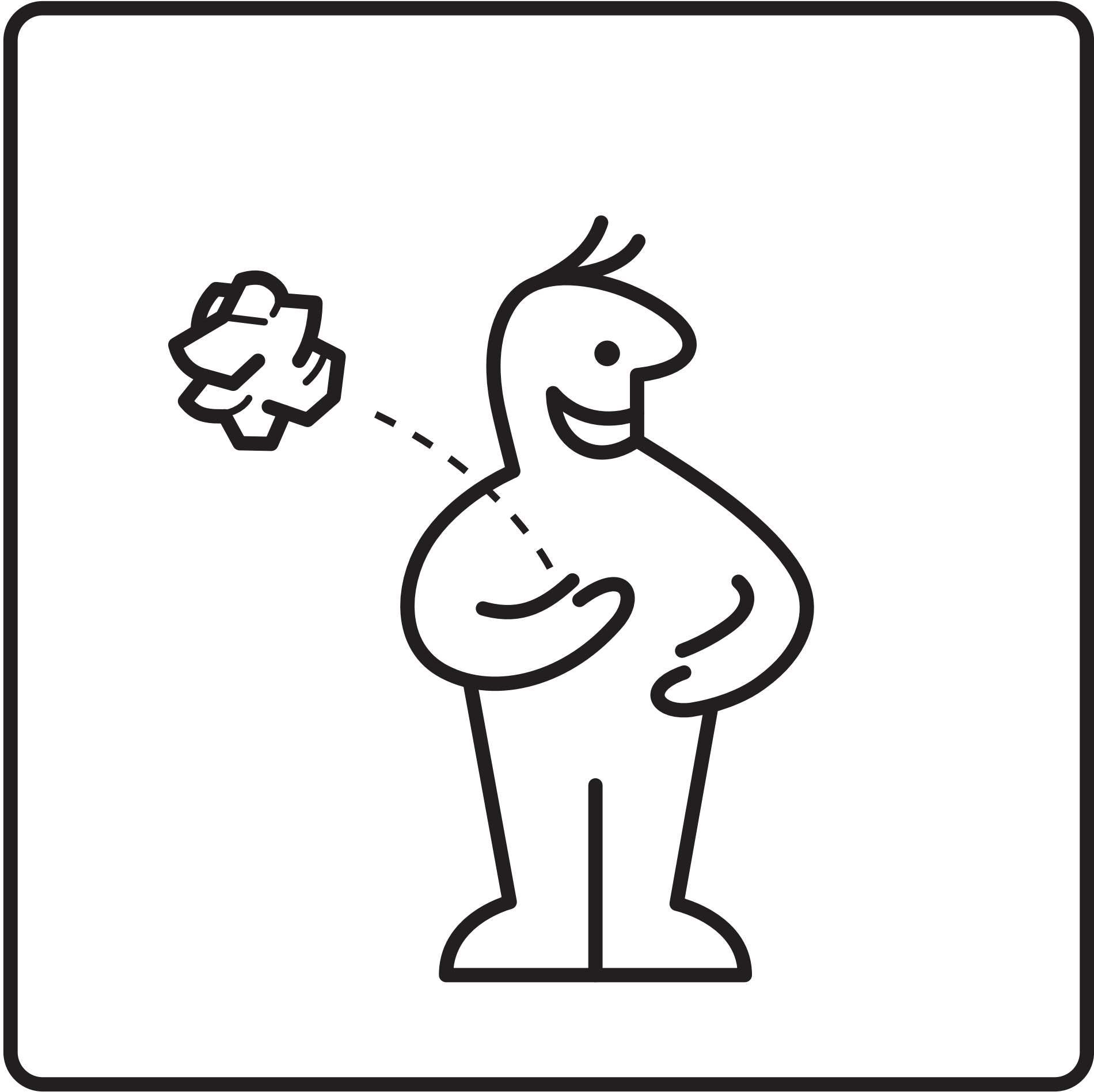


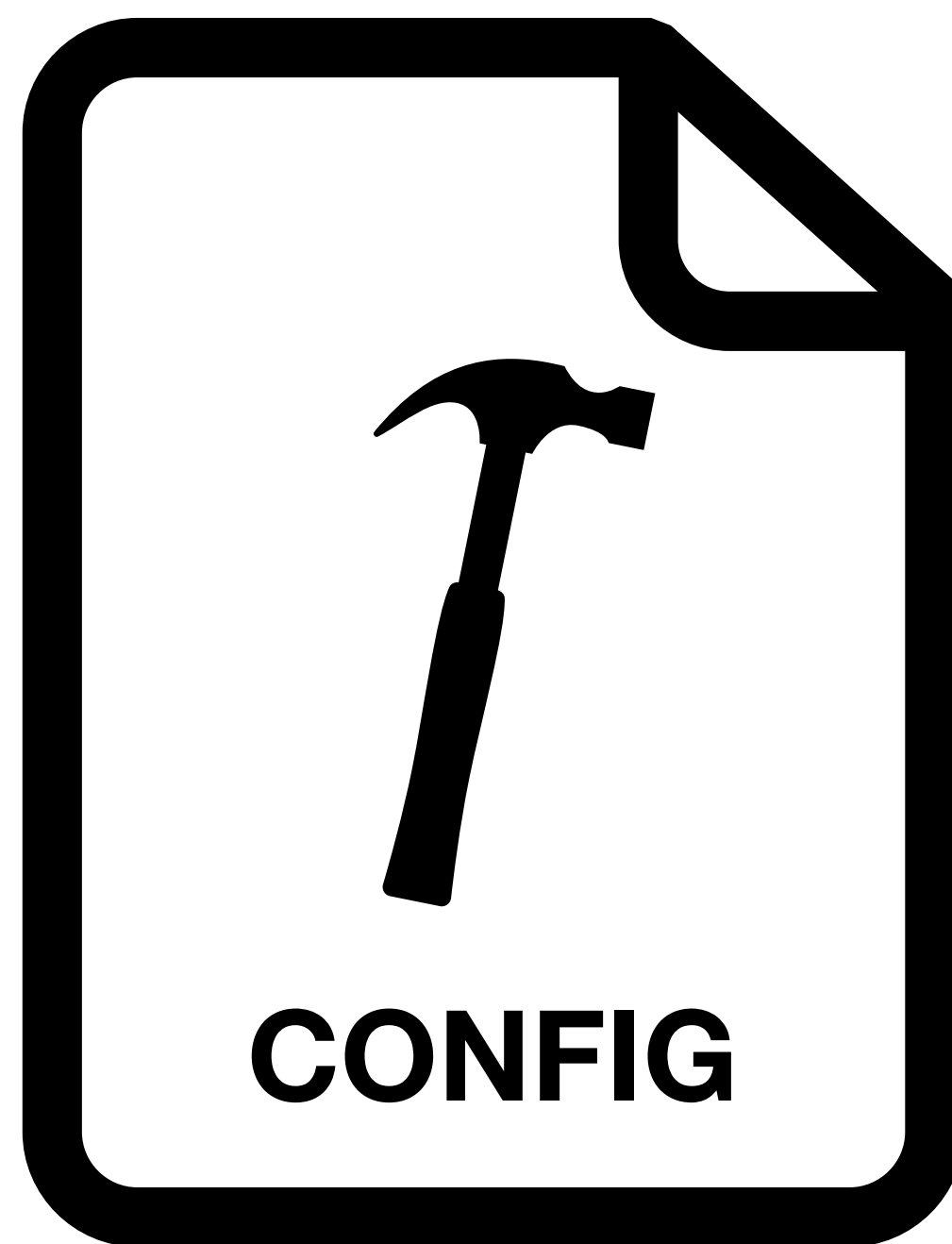

```
App.xcodeproj/  
├── project.pbxproj  
├── project.xcworkspace  
│   └── contents.xcworkspacedata  
├── xcshareddata  
│   └── xcscschemes  
│       └── App.xcscheme  
└── xcuserdata  
    └── jappleseed.xcuserdatad  
        └── xcscschemes  
            └── xcscschememanagement.plist
```

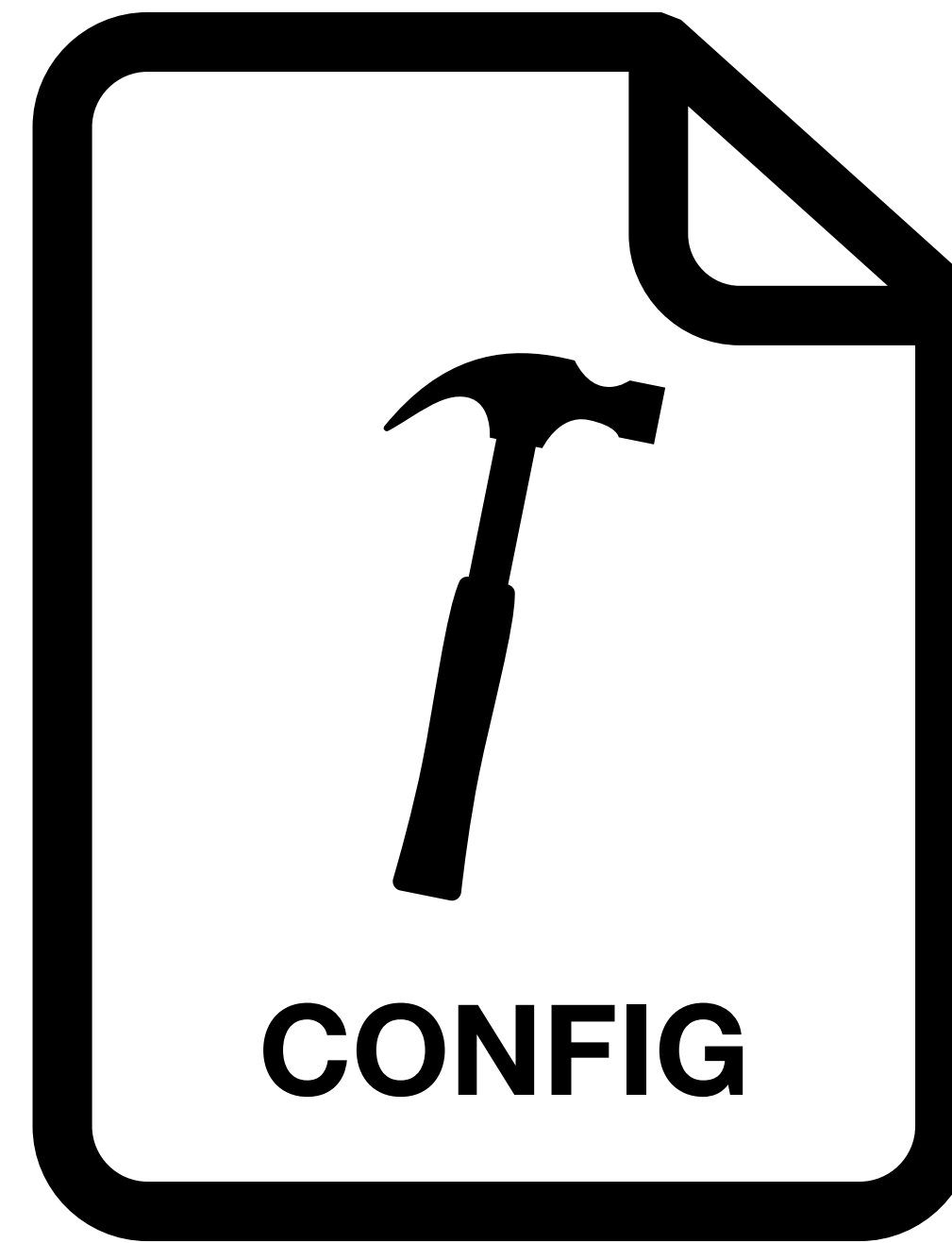
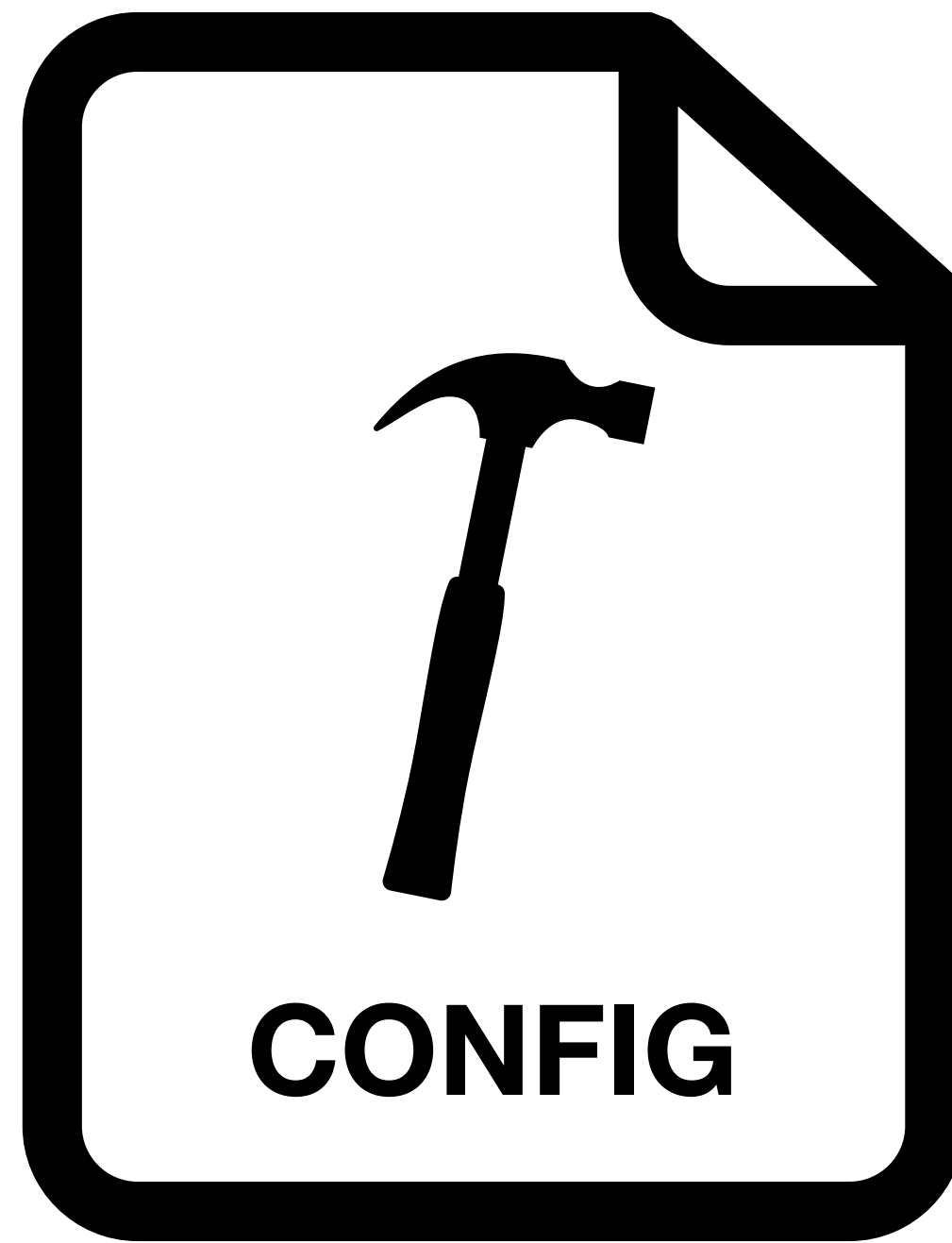
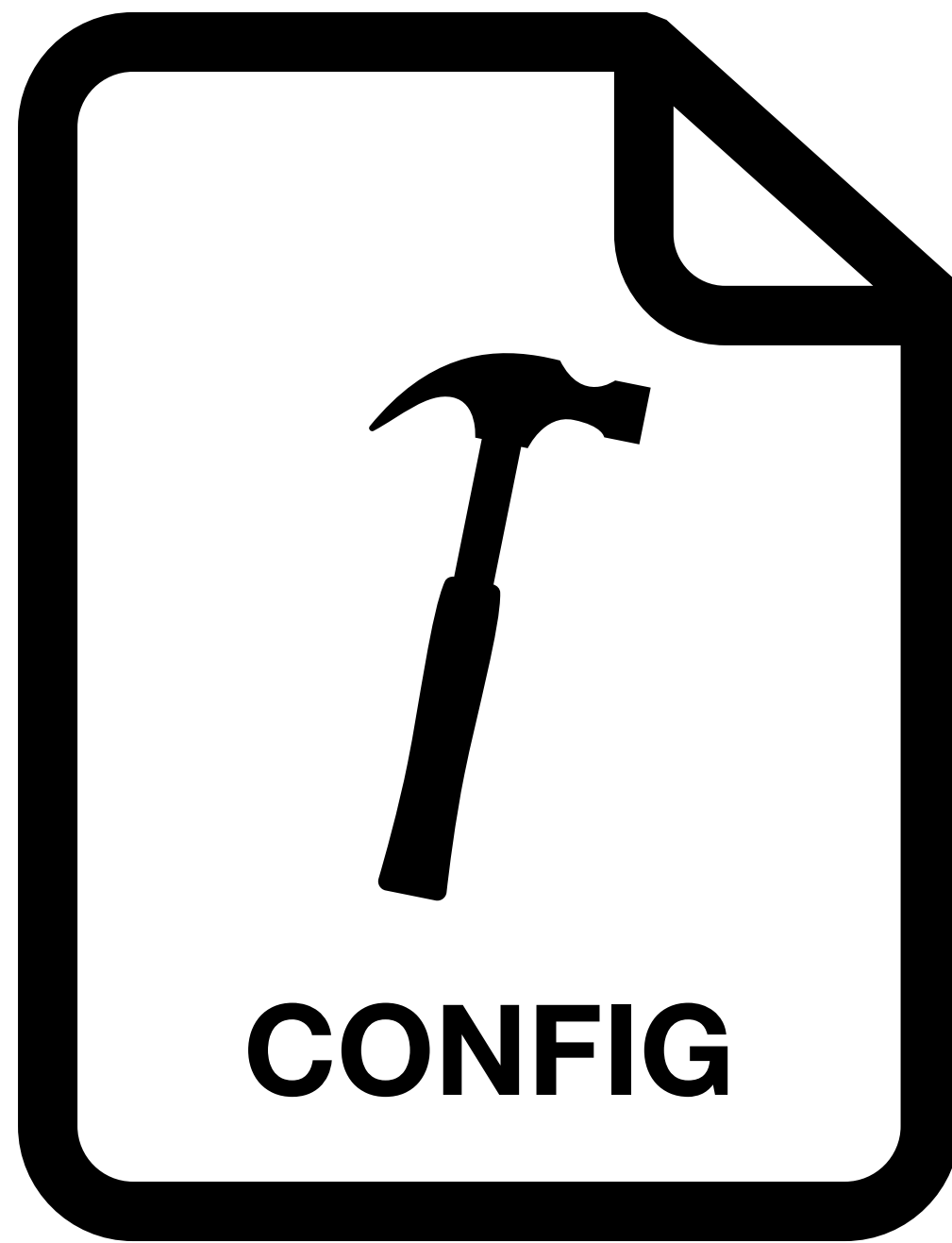
```
// !$*UTF8*$!
{
    archiveVersion = 1;
    classes = {
    };
    objectVersion = 50;
    objects = {

/* Begin PBXBuildFile section */
        F81C998C2288ABBA00664A17 /* AppDelegate.swift in Sources */ = {isa =
PBXBuildFile; fileRef = F81C998B2288ABBA00664A17 /* AppDelegate.swift */; };
        F81C998E2288ABBA00664A17 /* ViewController.swift in Sources */ = {isa =
PBXBuildFile; fileRef = F81C998D2288ABBA00664A17 /* ViewController.swift
*/; };
        F81C99912288ABBA00664A17 /* Main.storyboard in Resources */ = {isa =
PBXBuildFile; fileRef = F81C998F2288ABBA00664A17 /* Main.storyboard */; };
        F81C99932288ABBB00664A17 /* Assets.xcassets in Resources */ = {isa =
PBXBuildFile; fileRef = F81C99922288ABBB00664A17 /* Assets.xcassets */; };
        F81C99962288ABBB00664A17 /* LaunchScreen.storyboard in Resources */ =
{isa = PBXBuildFile; fileRef = F81C99942288ABBB00664A17 /*
LaunchScreen.storyboard */; };
/* End PBXBuildFile section */

/* Begin PBXFileReference section */
        F81C99882288ABBA00664A17 /* App.app */ = {isa = PBXFileReference;
explicitFileType = wrapper.application; includeInIndex = 0; path = App.app;
sourceTree = BUILT_PRODUCTS_DIR; };
    }
```







xcprojectconfig

New

```
$ brew tap nshipster/formulae
```

```
$ brew install xcprojectconfig
```

```
$ xcprojectconfig list \  
  ./path/to/App.xcodeproj
```

xcprojectconfig

New

```
FRAMEWORK_SEARCH_PATHS =  
    $(DEVELOPER_FRAMEWORKS_DIR) $(inherited)
```

```
INFOPLIST_FILE = MyApp/MyApp-Info.plist
```

xcprojectlint

```
$ brew tap nshipster/formulae
```

```
$ brew install xcprojectlint
```

```
$ xcprojectlint --project ./path/to/App.xcodeproj \  
    --validations build-settings-externalized \  
    --report error
```

```
project.pbxproj:848: error:
```

```
Release has settings defined at the project level.
```


xcodegen

```
$ brew install xcodegen
```

```
$ xcodegen generate
```

```
name: MyApp
targets:
  App:
    type: app
    platform: [iOS]
    settings:
      IPHONEOS_DEPLOYMENT_TARGET: 13.0
    scheme:
      testTargets:
        - AppTests
    sources:
      - Sources
      - Supporting Files
      - path: Headers
        buildPhase: headers
        type: folder
        headerVisibility: public
    dependencies:
      - cocoapods: Alamofire
```

Auto-merging project.pbproj

CONFLICT (content): Merge conflict in project.pbproj

Automatic merge failed;

fix conflicts and then commit the result

mergepbx

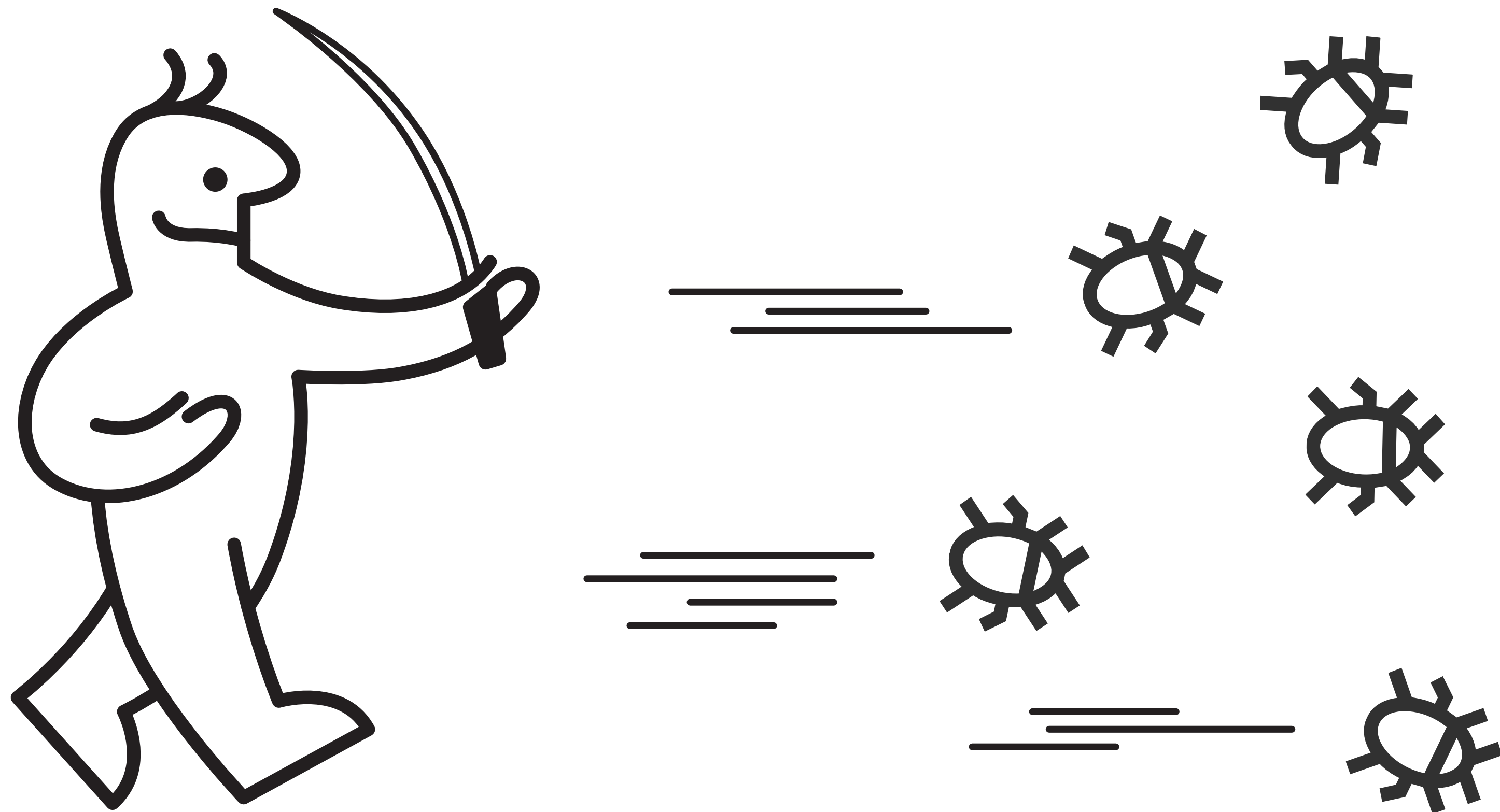
```
$ brew install mergepbx
```

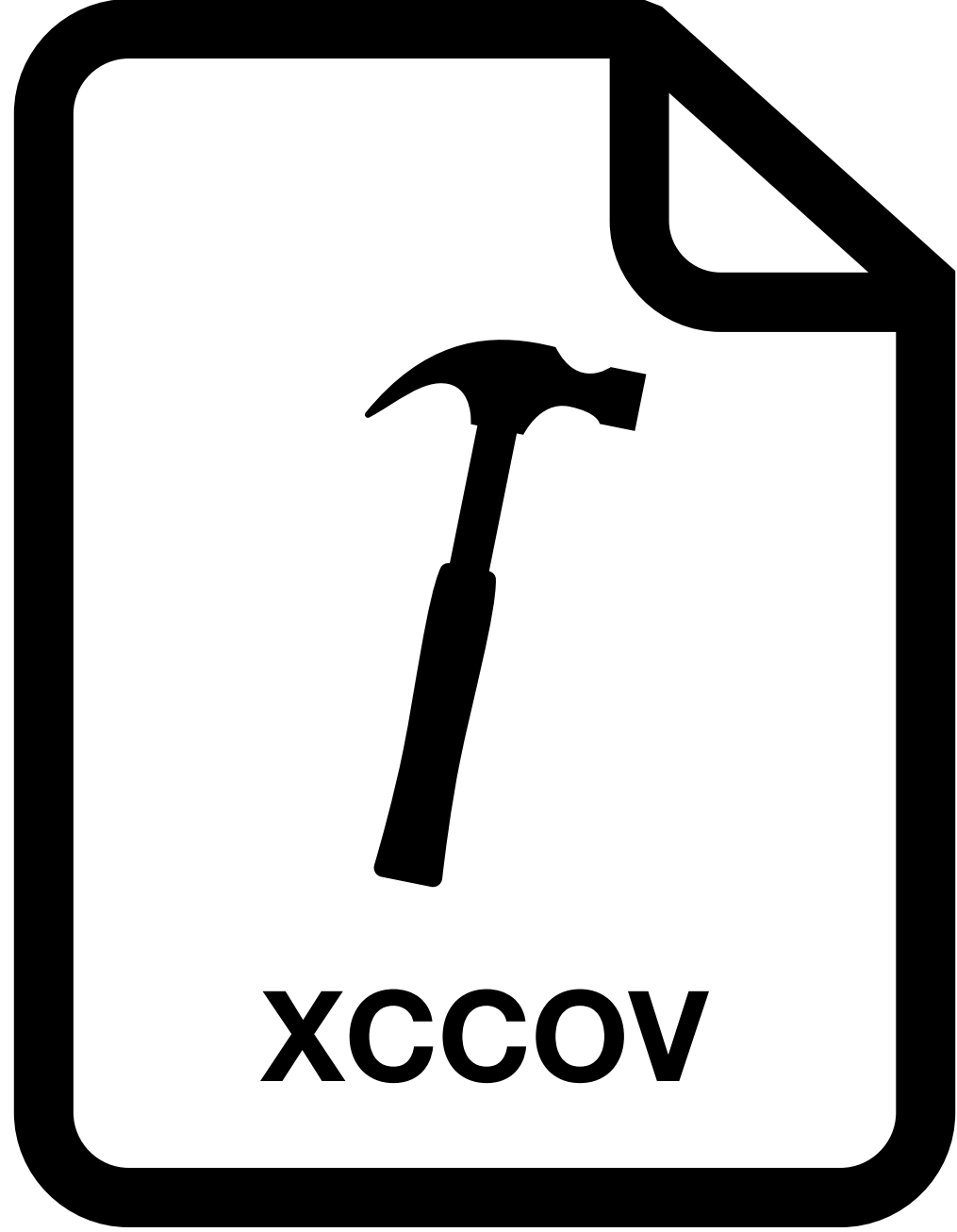
```
$ git mergetool --tool=mergepbx PROJECT.pbxproj
```

mergepbx

```
$ echo 'merge "mergepbx"]  
    name = Xcode project files merger  
    driver = mergepbx %0 %A %B' \  
>> ~/.gitconfig
```

```
$ echo "*.pbxproj merge=mergepbx" \  
>> .gitattributes
```





Code Coverage in Xcode

```
func greeting() -> String {  
    return "Ciao!"  
}
```



```
func closing() -> String {  
    return "Arrivederci!"  
}
```



xcperfect



```
$ gem install xcperfect
```

```
$ xcrun xccov view --json \
    path/to/report.xcresult | \
    xcperfect
```


xcperfect

▼ App.app

▼ Hello.swift

greeting() -> String	1 / 1		100.0 %
closing() -> String	0 / 1		0.0 %

~/Library/Developer/Xcode/DerivedData/MyApp-_____?

xcderiveddata

New

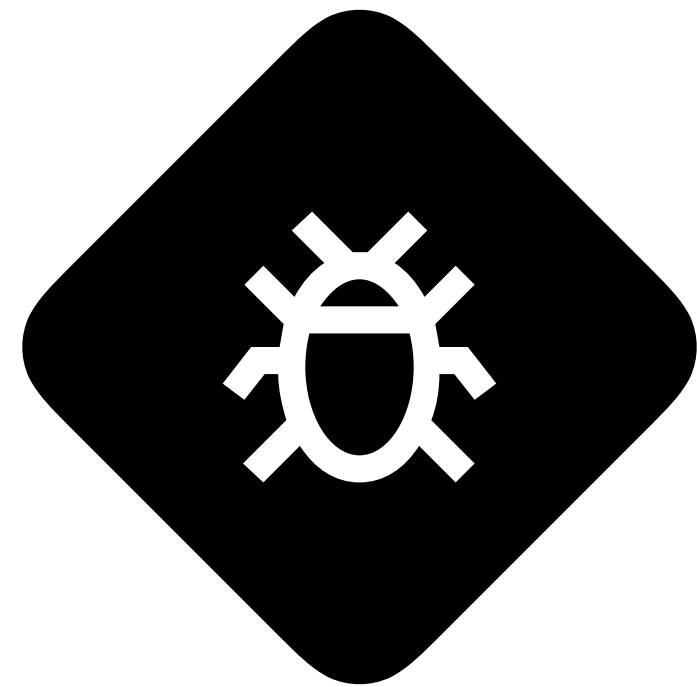
Before

```
$ xcodebuild -showBuildSettings | \
    grep -m 1 "BUILD_DIR" | \
    grep -oEi "\/.*" | \
    sed 's#/Build/Products##'
```

After

```
$ xcderiveddata
```





Fuzz Testing

```
@_cdecl("LLVMFuzzerTestOneInput")  
func f(UnsafePointer<CChar>, CInt) -> Int
```

libFuzzer

```
@_cdecl("LLVMFuzzerTestOneInput")
public func fuzzJSON(data pointer: UnsafePointer<CChar>,
                      size: CInt) -> CInt
{
    let buffer = UnsafeBufferPointer(start: pointer,
                                     count: Int(size))

    let data = Data(buffer: buffer)
    _ = try? JSON.parse(data)
    return 0
}
```

libFuzzer

"{"

":"

"true"

"e"

"}"

","

"false"

"e+"

"["

" "

"null"

"e−"

"]"

"\\n"

"\\u"

"E"

"\""

"\\r"

"\\b"

"E+"

"\""

"/"

"\\f"

"E−"

"\\\""

"*/"

"\\t"

"//"

"."



UI Fuzz Testing

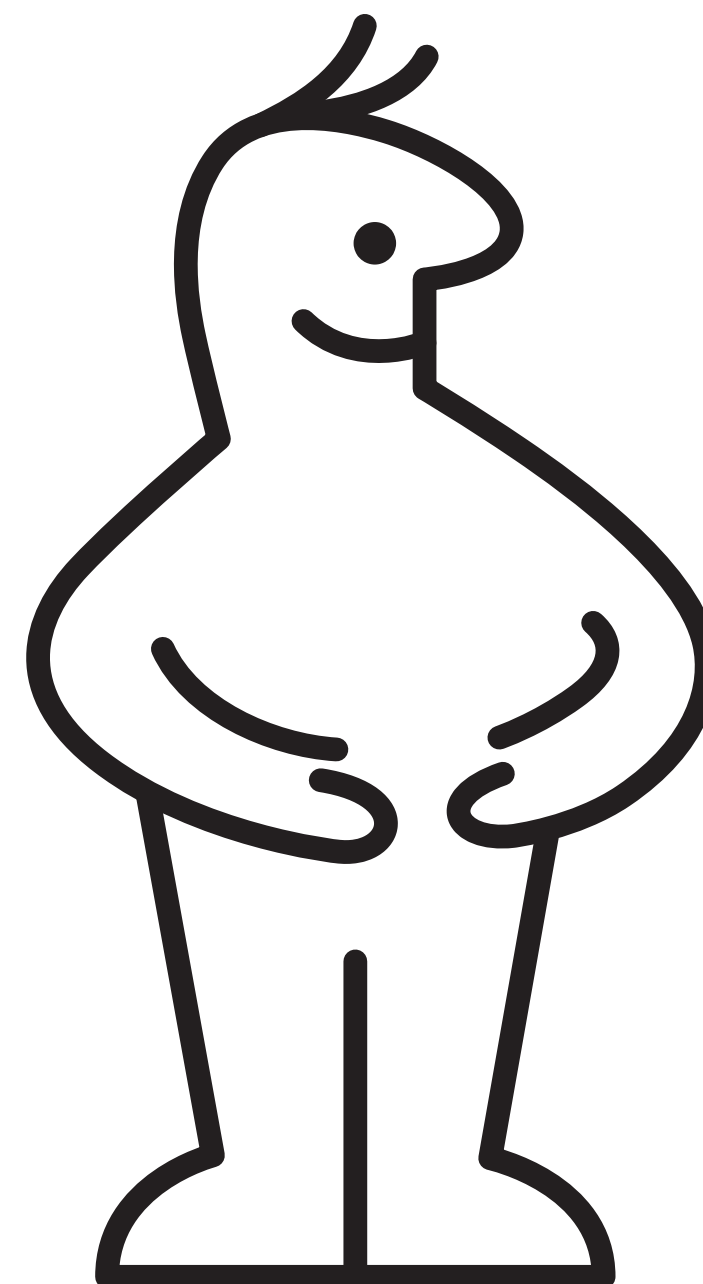
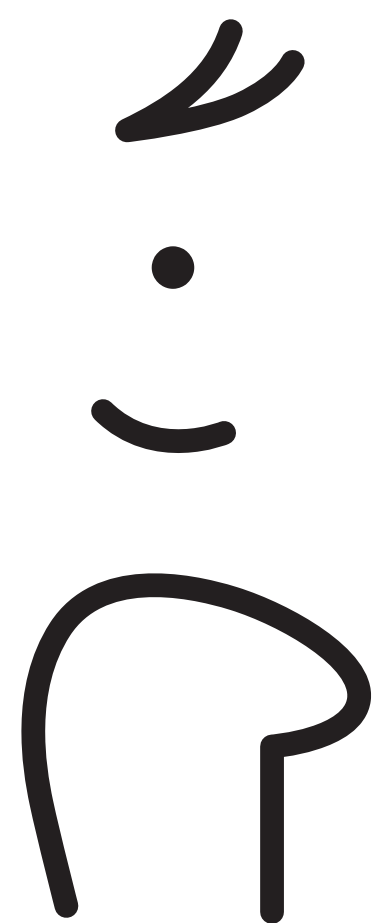
SwiftMonkey

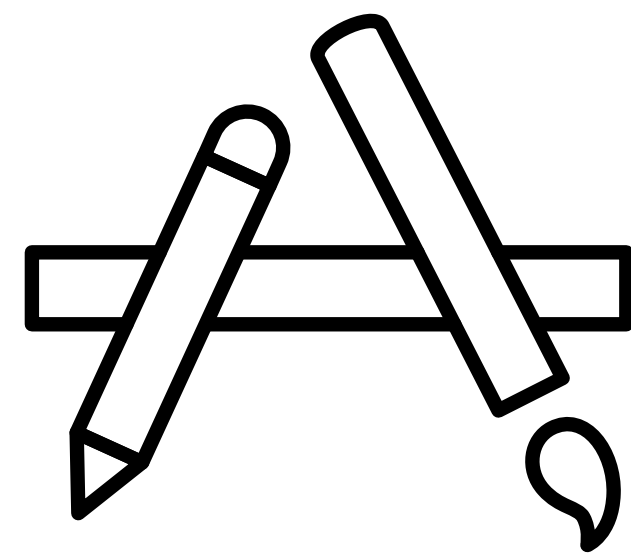
```
target 'App' do
    pod "SwiftMonkeyPaws", '~> 2.1.0'
end
```

```
target 'Tests' do
    pod 'SwiftMonkey', '~> 2.1.0'
end
```

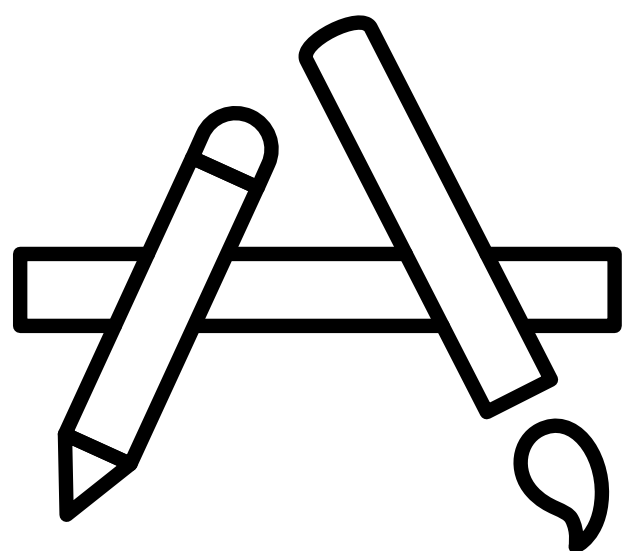
SwiftMonkey



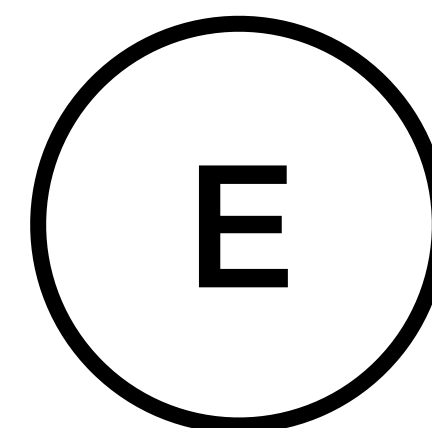




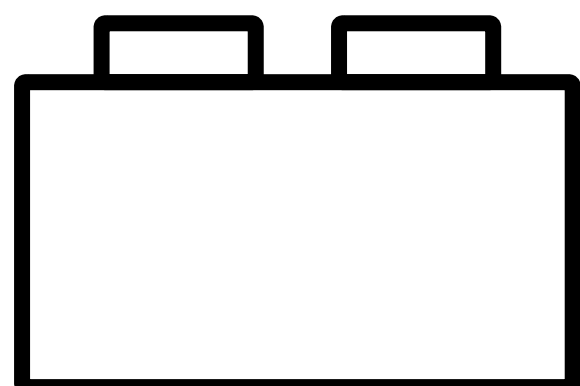
MyApp



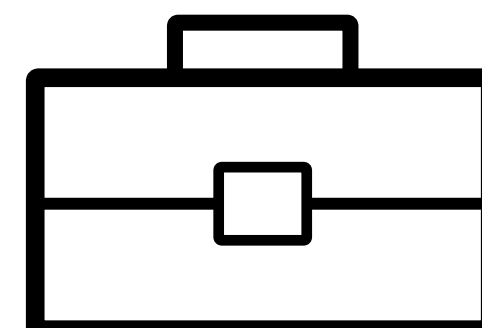
MyApp



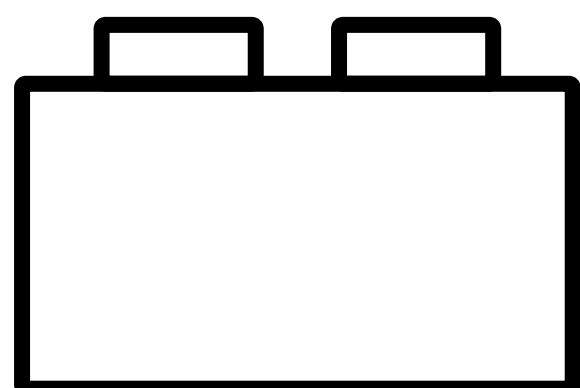
SomeExtension



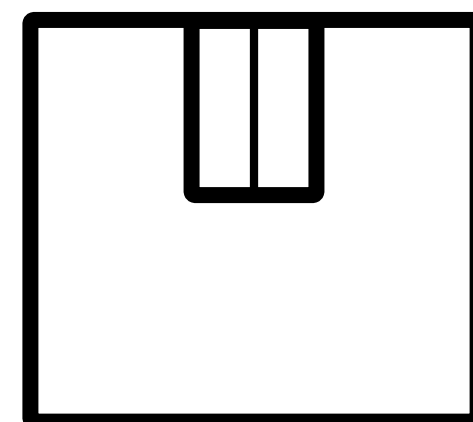
MyAppTests



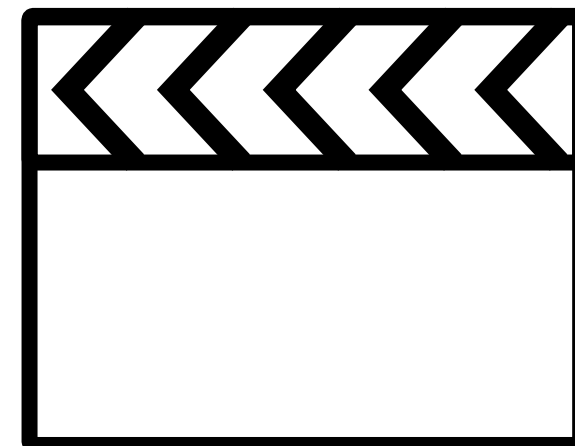
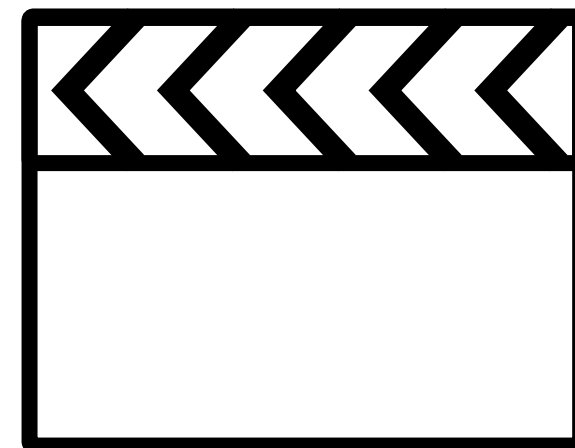
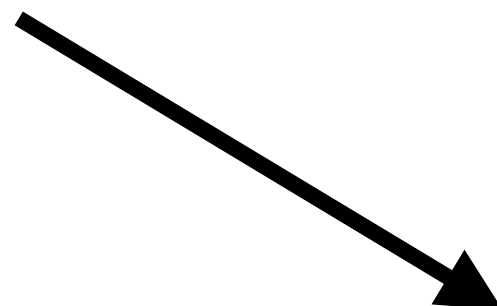
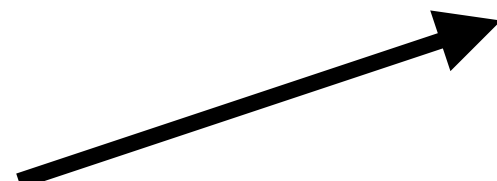
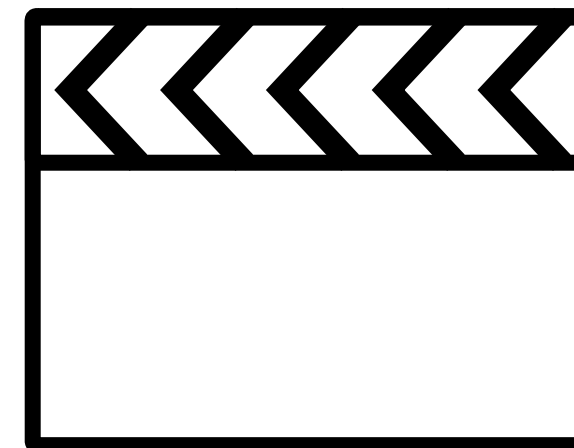
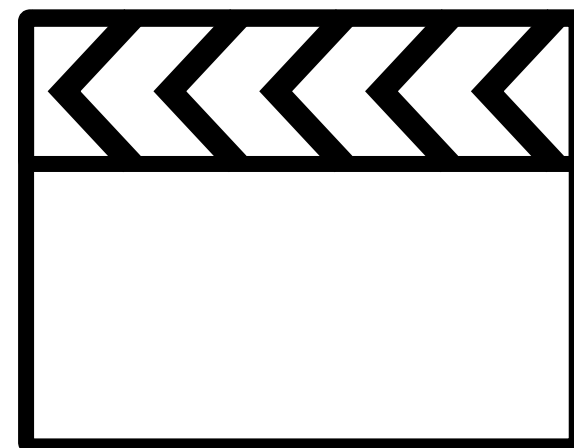
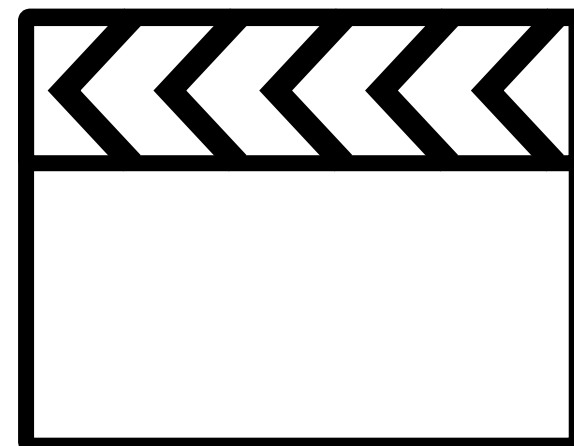
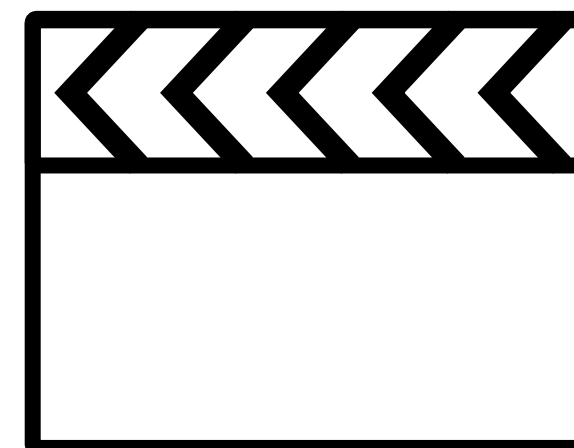
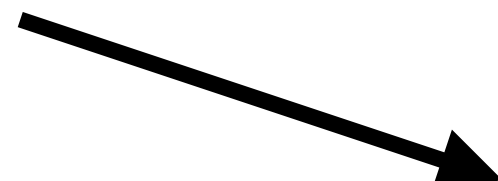
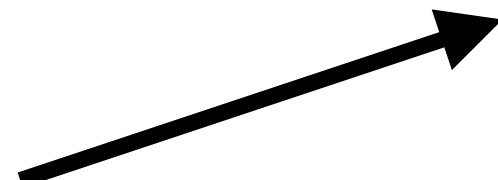
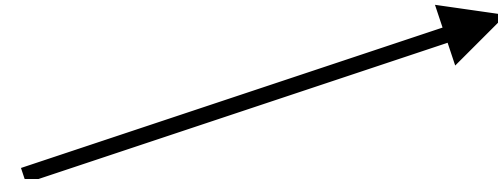
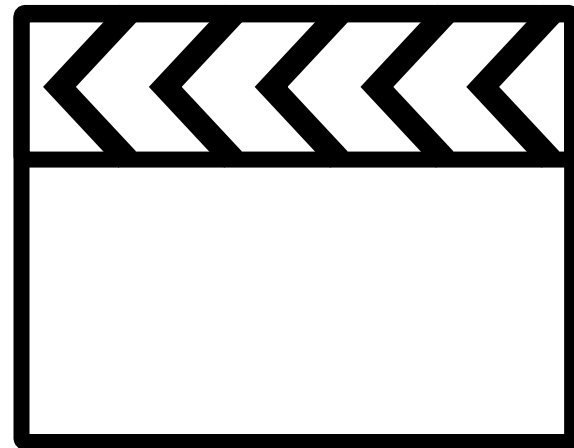
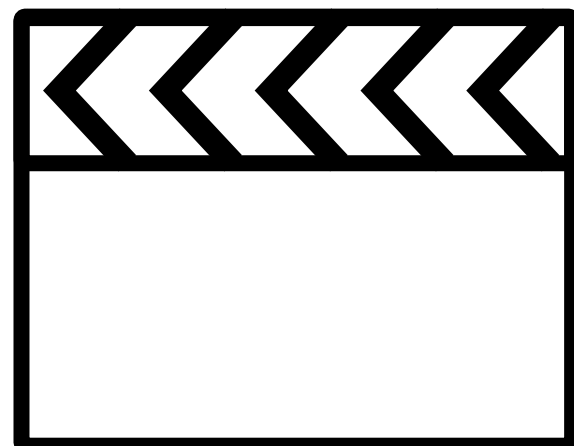
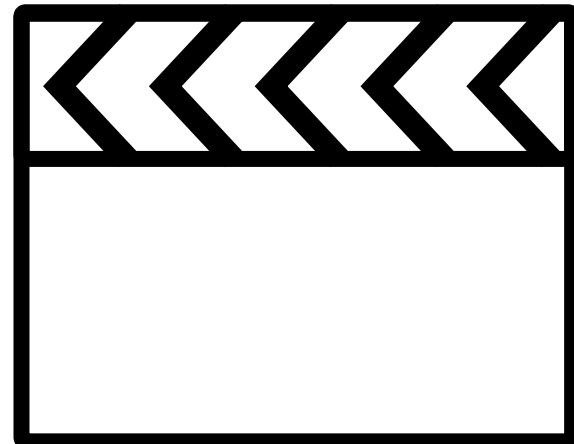
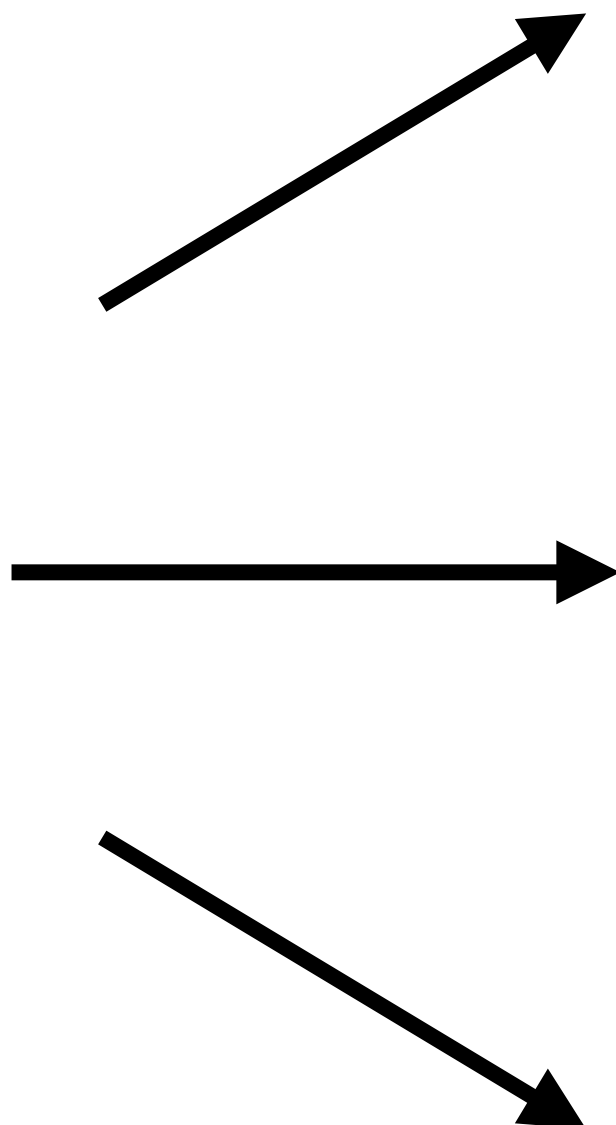
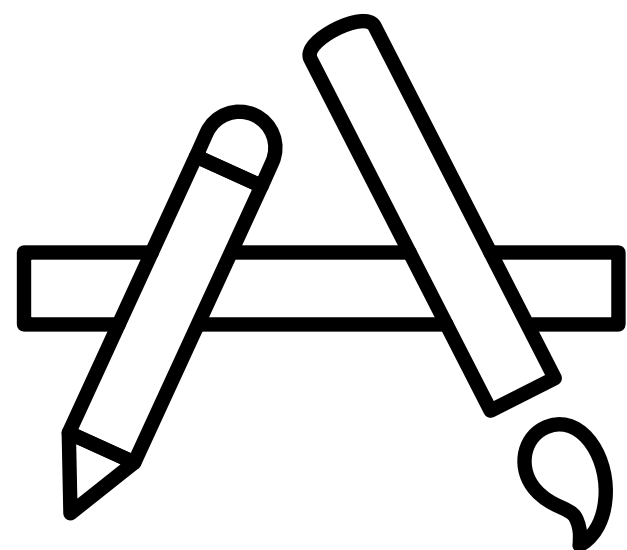
MyAppUI

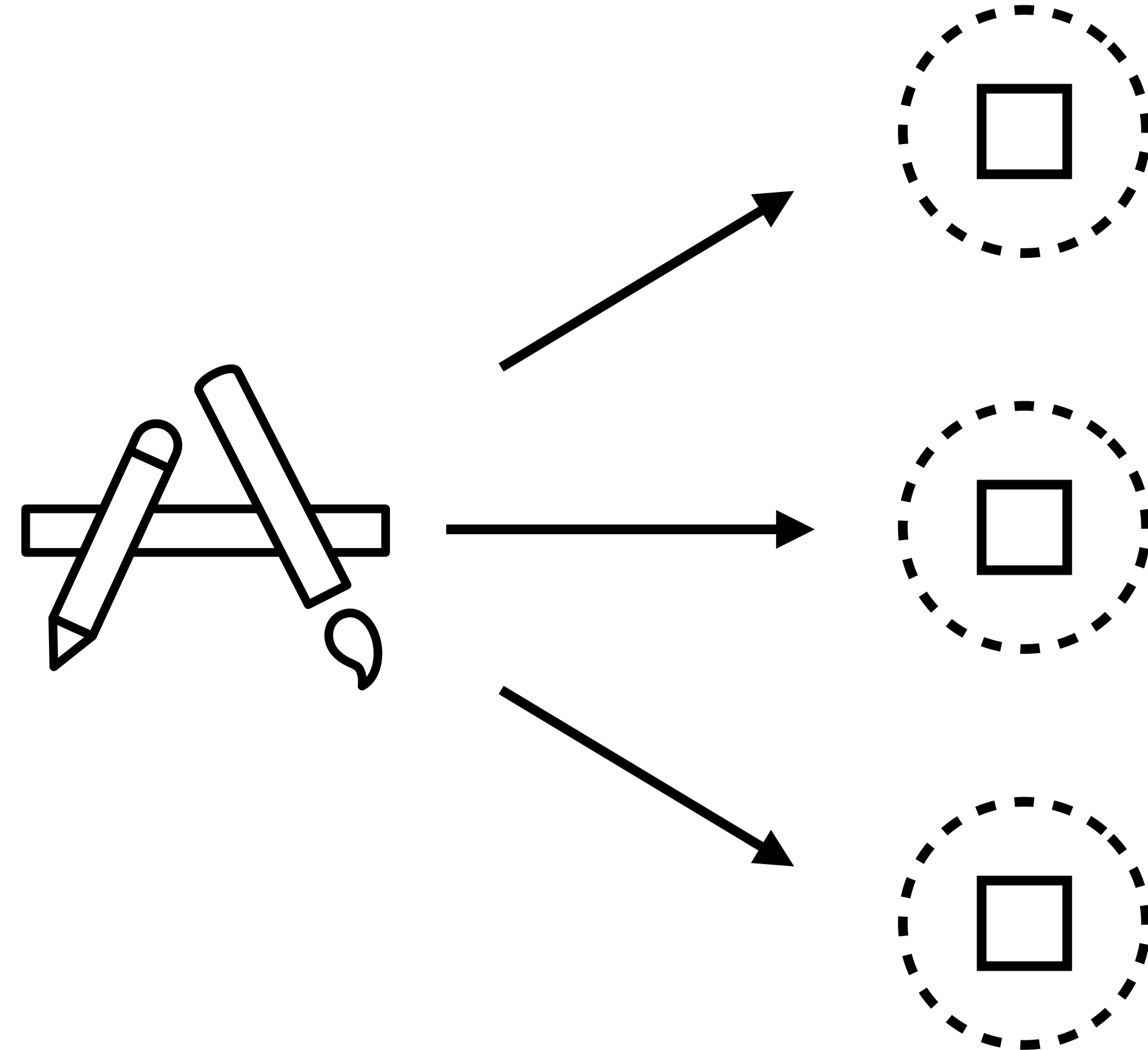


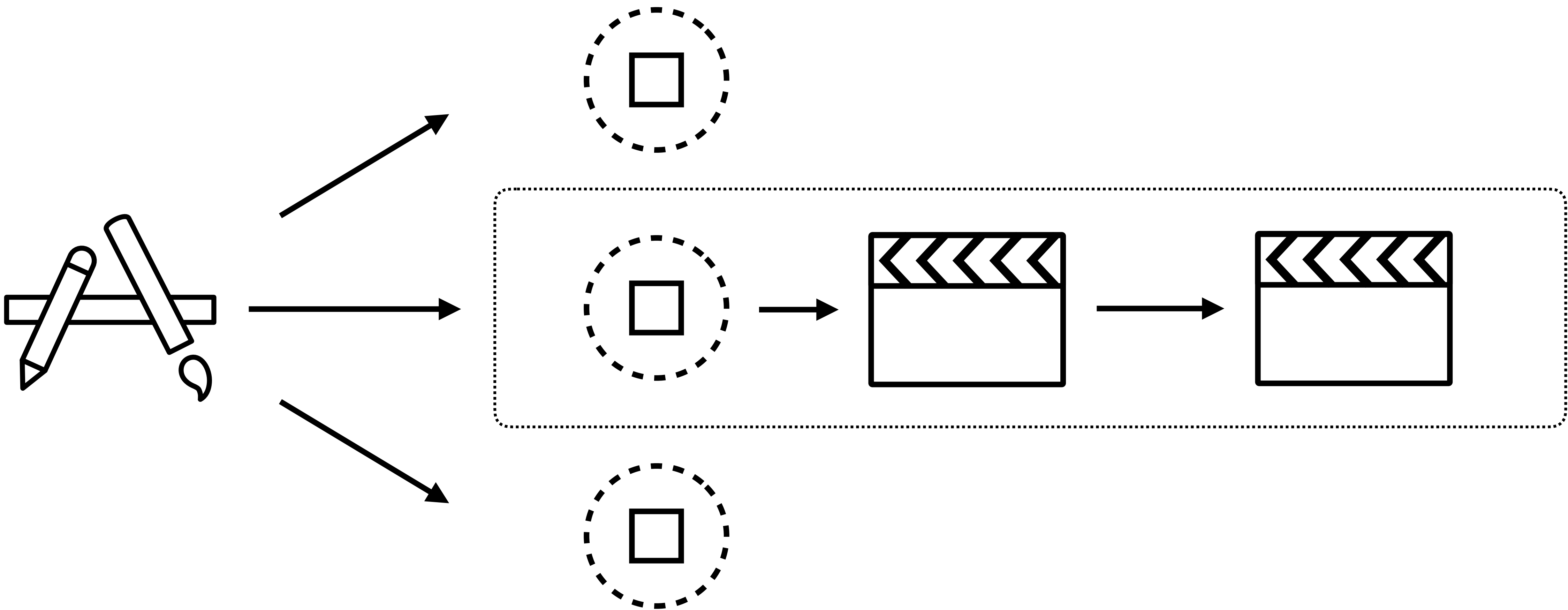
MyAppUITests



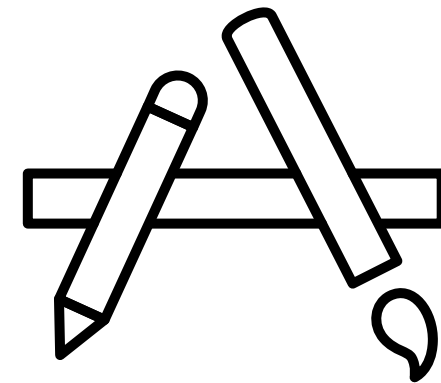
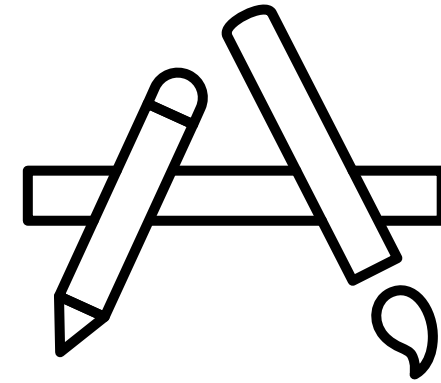
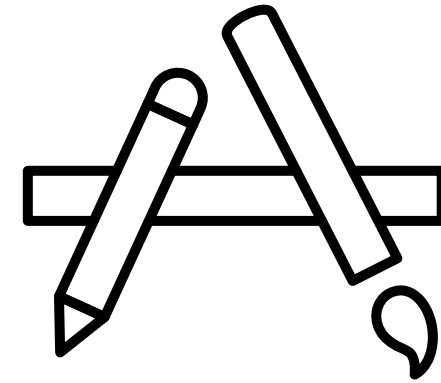
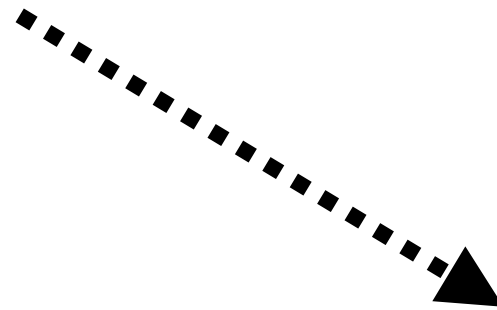
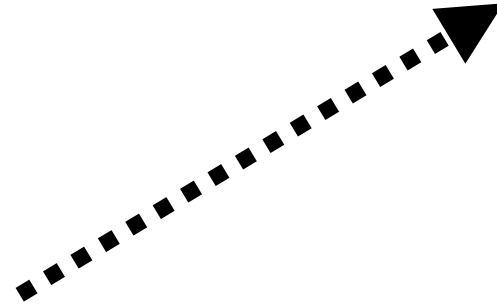
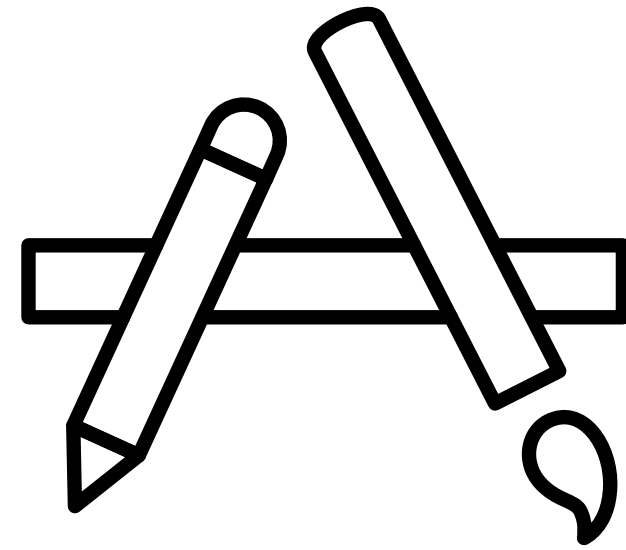
SomePackage

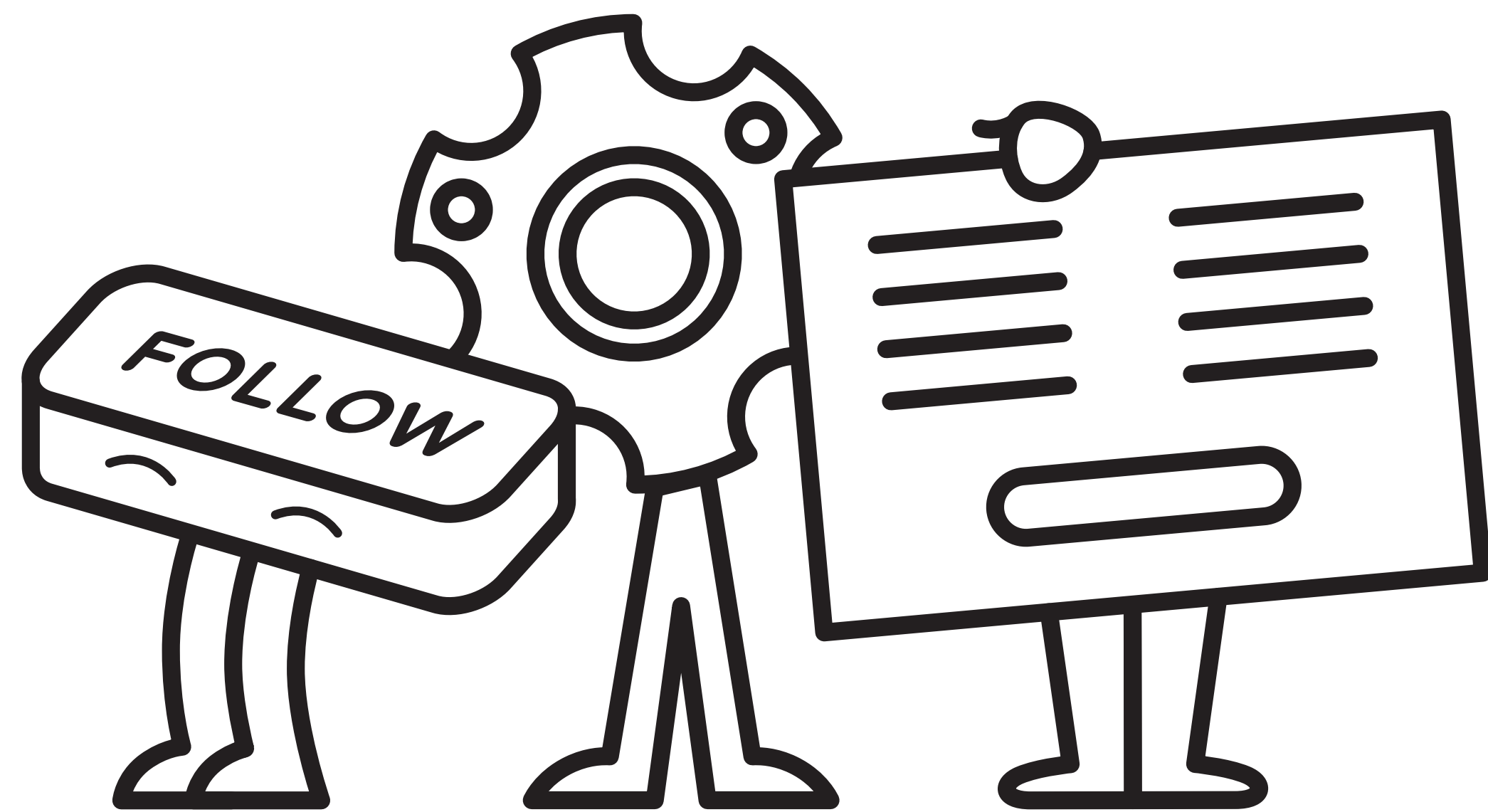
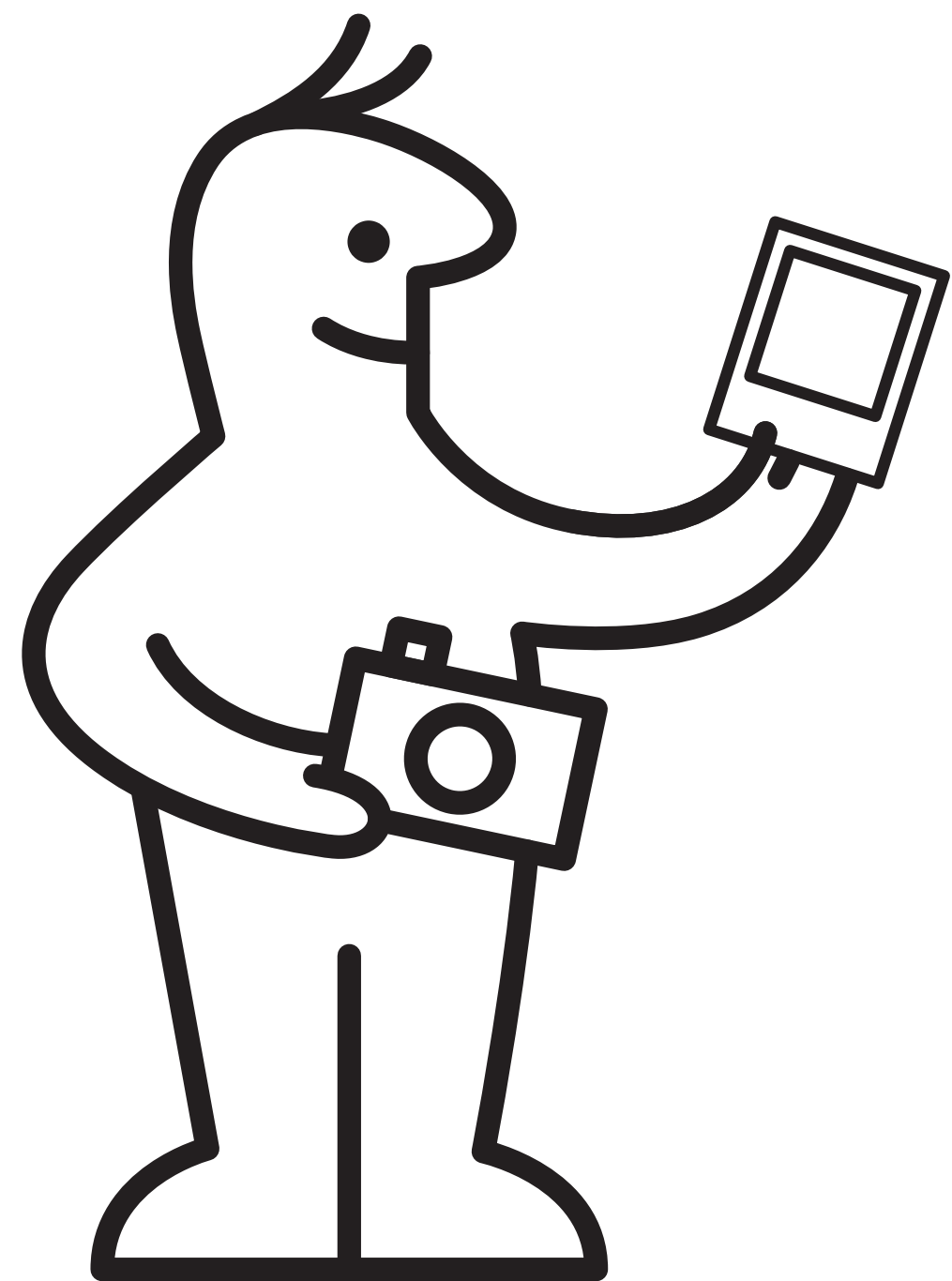


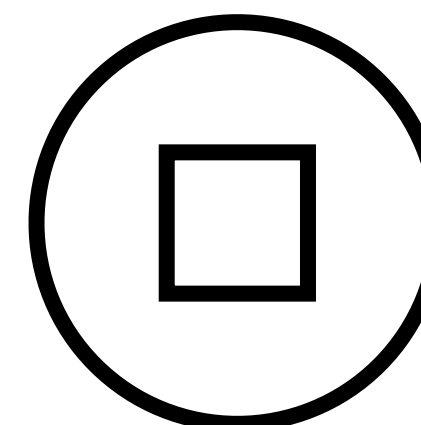
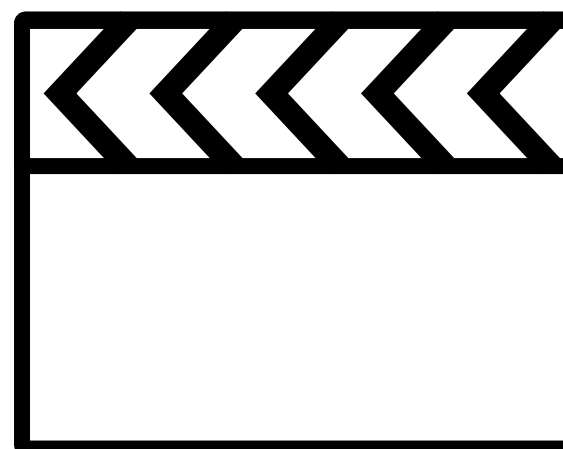
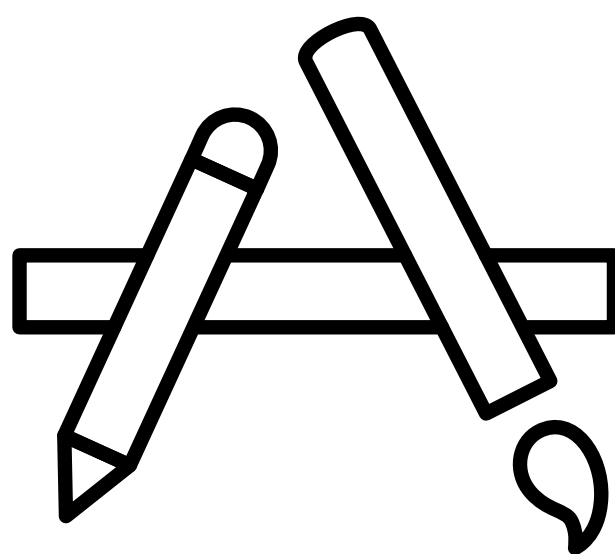


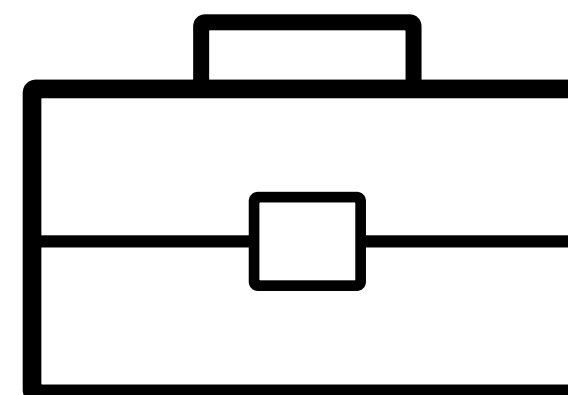
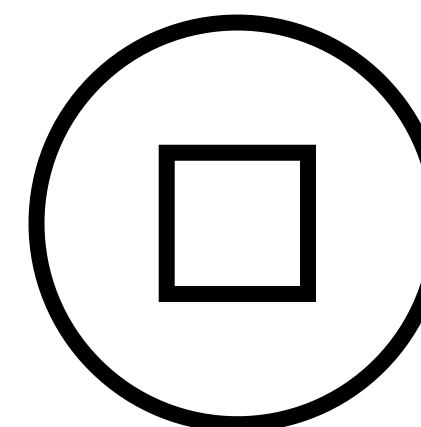
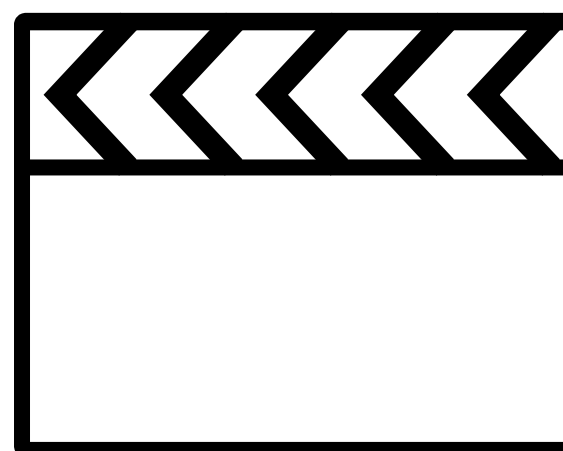
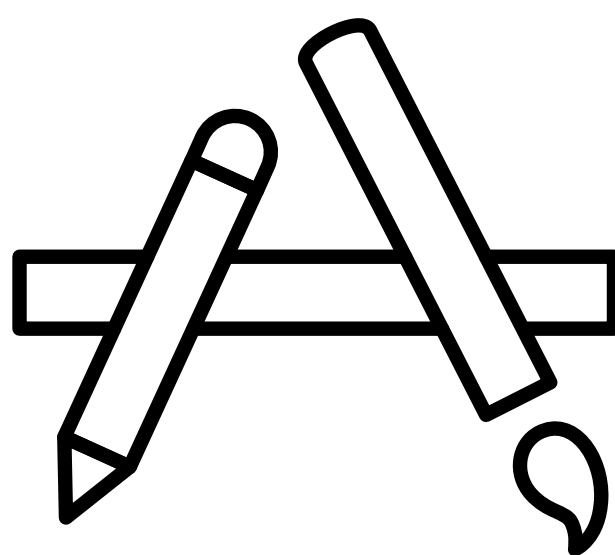


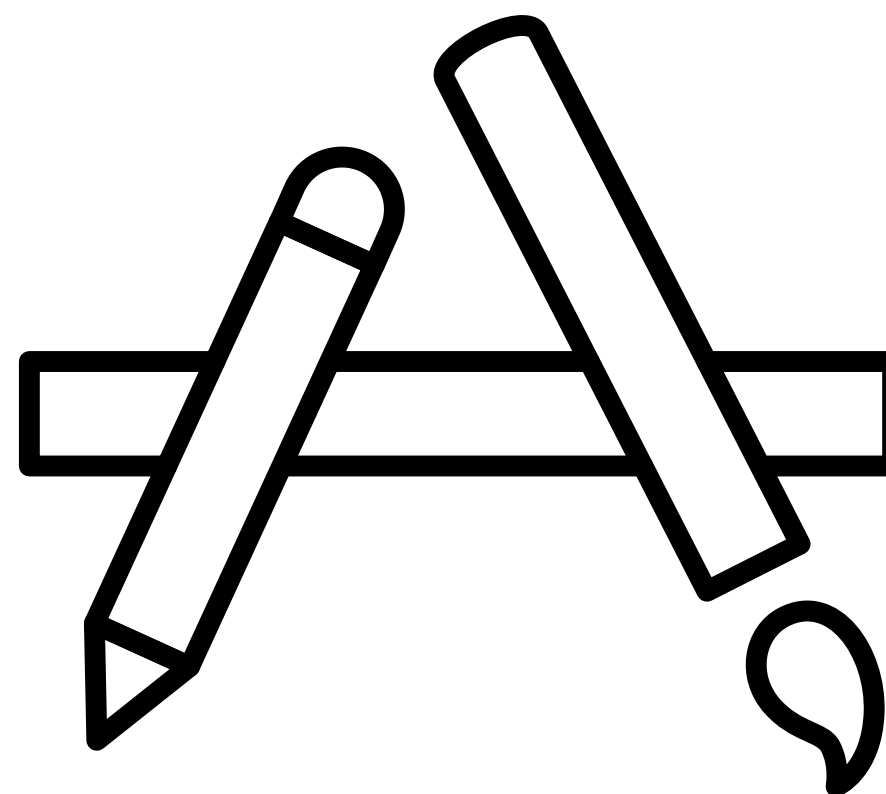
?



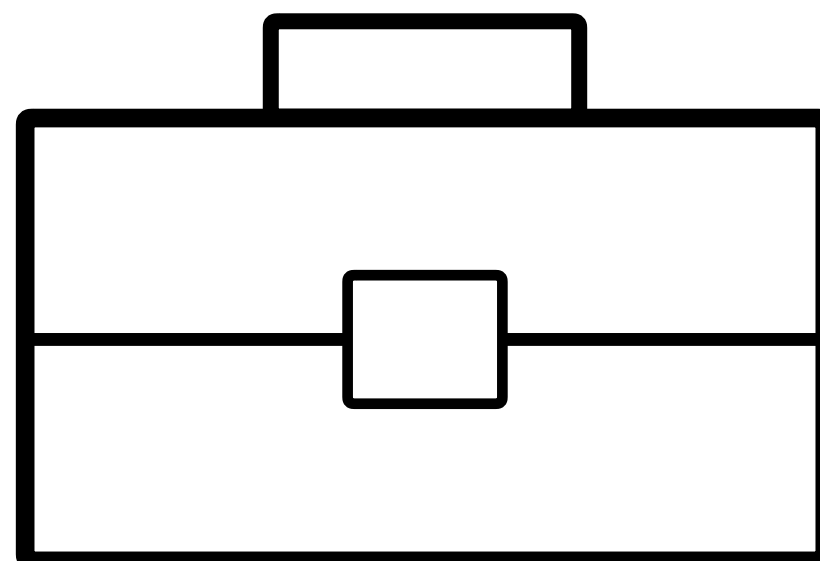




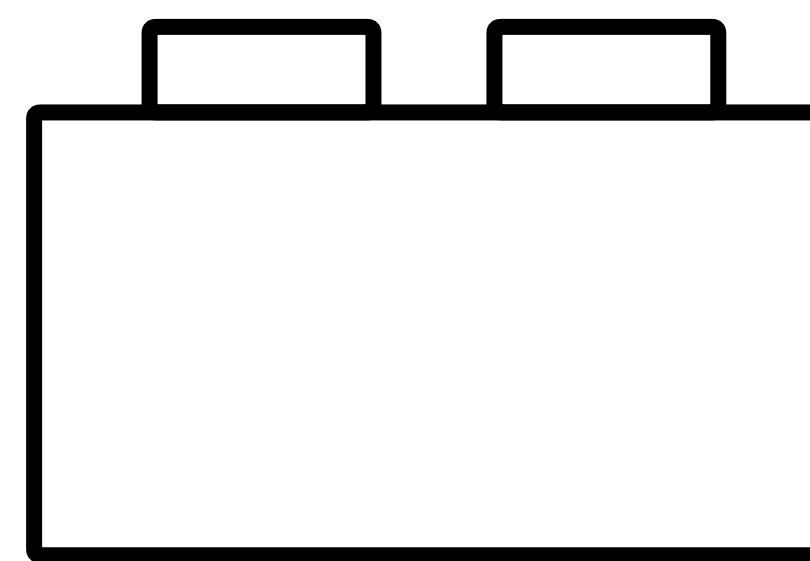




App



AppUI

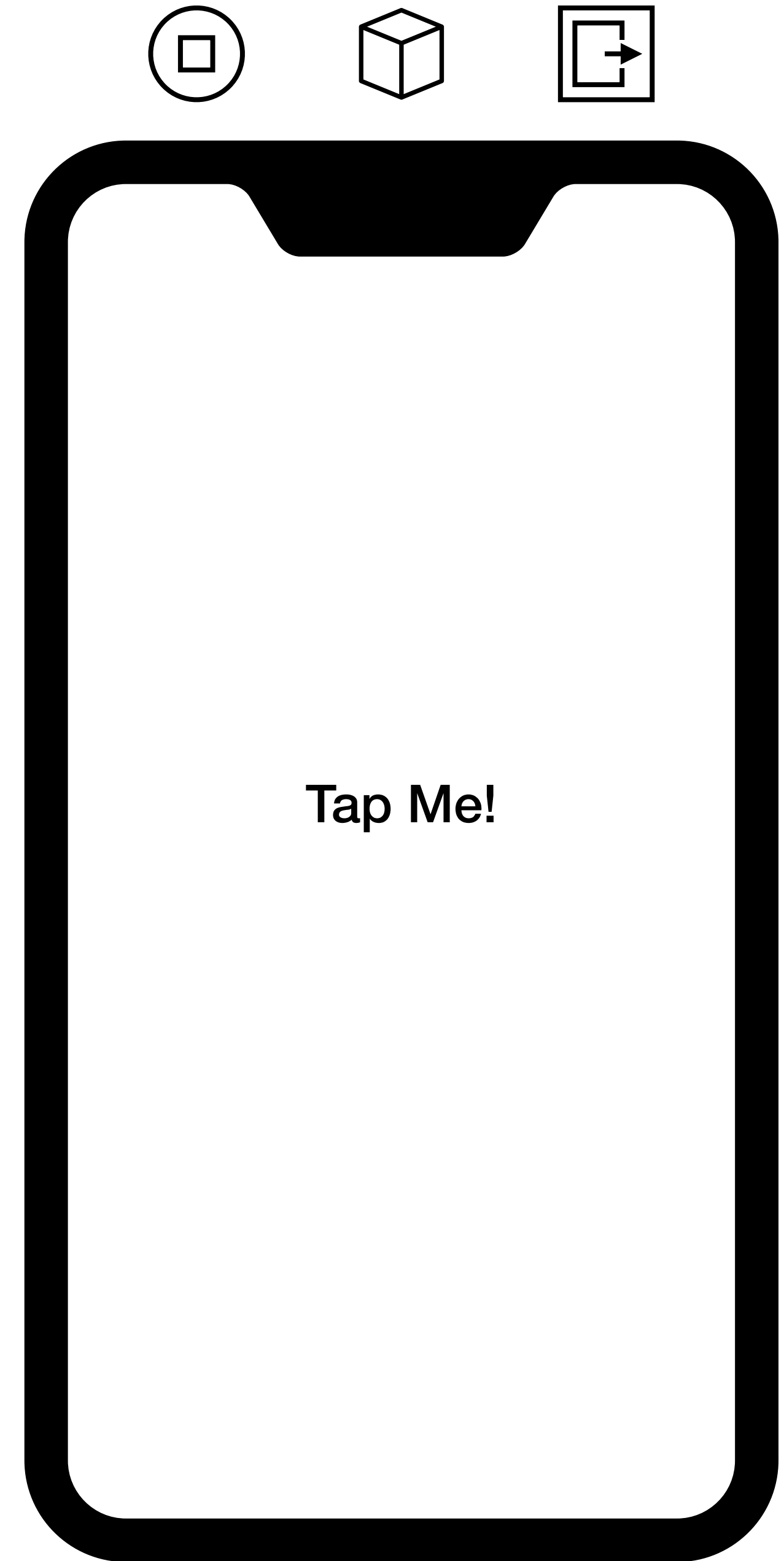


AppUITests

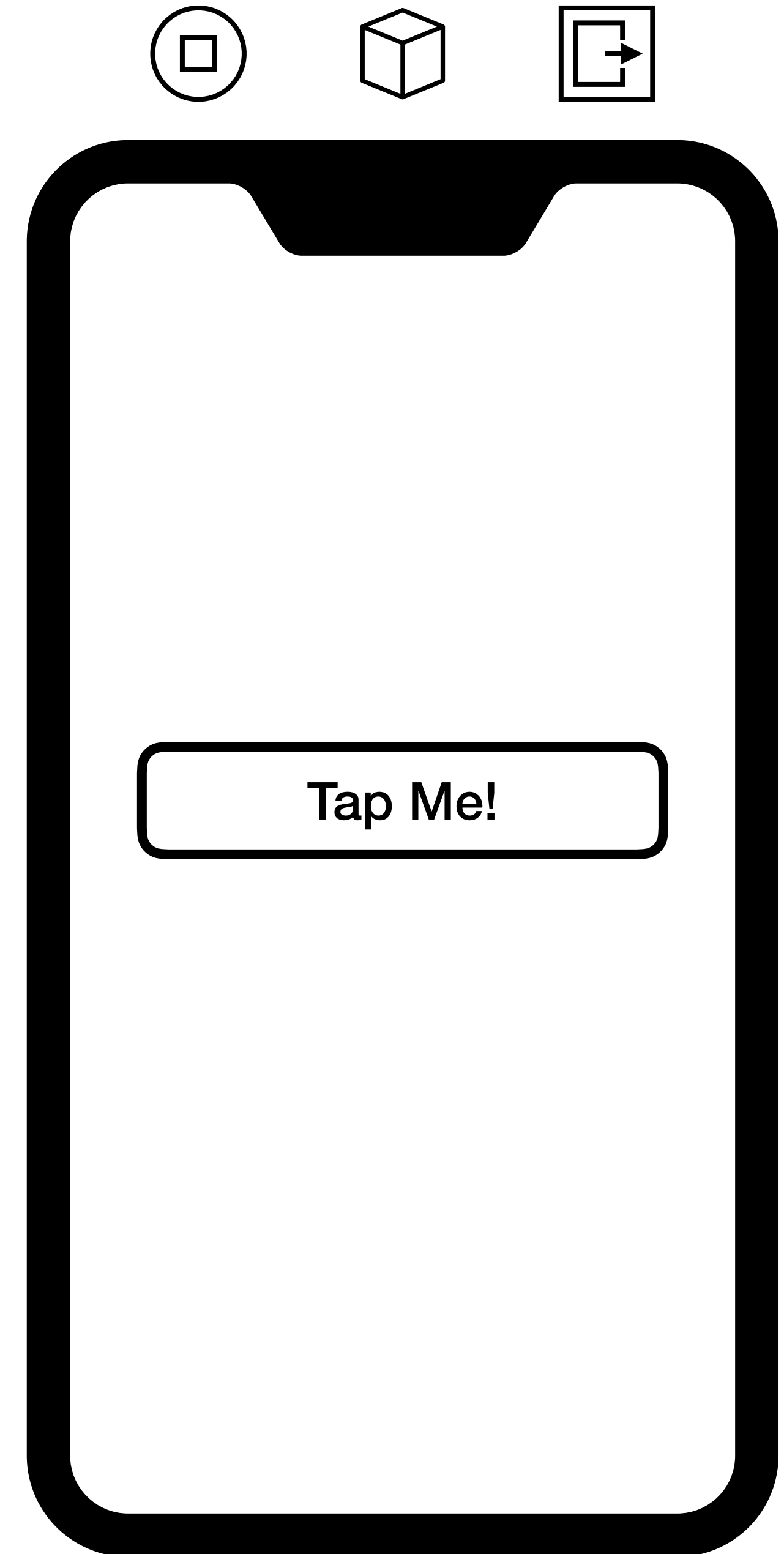
```
class BorderedButton: UIButton {  
    var cornerRadius: CGFloat  
    var borderWidth: CGFloat  
    var borderColor: UIColor?  
}
```



```
class BorderedButton: UIButton {  
    var cornerRadius: CGFloat  
    var borderWidth: CGFloat  
    var borderColor: UIColor?  
}
```




```
@IBDesignable  
class BorderedButton: UIButton {  
  
    @IBInspectable  
    var cornerRadius: CGFloat  
  
    @IBInspectable  
    var borderWidth: CGFloat  
  
    @IBInspectable  
    var borderColor: UIColor?  
  
}
```



```
#if canImport(SwiftUI) && DEBUG
import SwiftUI

@available(iOS 13.0, *)
struct BorderedButton_Preview: PreviewProvider {
    static var previews: some View {
        UIViewPreview {
            let button = BorderedButton(frame: .zero)
            button.setTitle("Tap Me!", for: .normal)
            return button
        }.padding(10)
    }
}
#endif
```

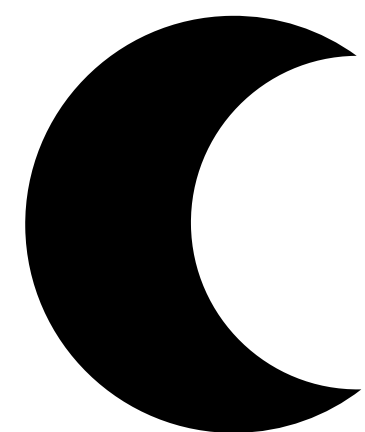


```
#if canImport(SwiftUI) && DEBUG
import SwiftUI

@available(iOS 13.0, *)
struct BorderedButton_Preview: PreviewProvider {
    static var previews: some View {
        Group {
            UIViewPreview {
                let button = BorderedButton()
                button.setTitle("Tap Me!", for: .normal)
                return button
            }
            UIViewPreview {
                let button = BorderedButton()
                button.setTitle("Toccami!", for: .normal)
                return button
            }
        }.padding(10)
        .previewLayout(.fixed(width: 200, height: 100))
    }
}
#endif
```

Tap Me!

Toccami!



Tap Me!

Tap Me!

Aあ

Tap Me!

Toccami!

これをタップ！

A_A

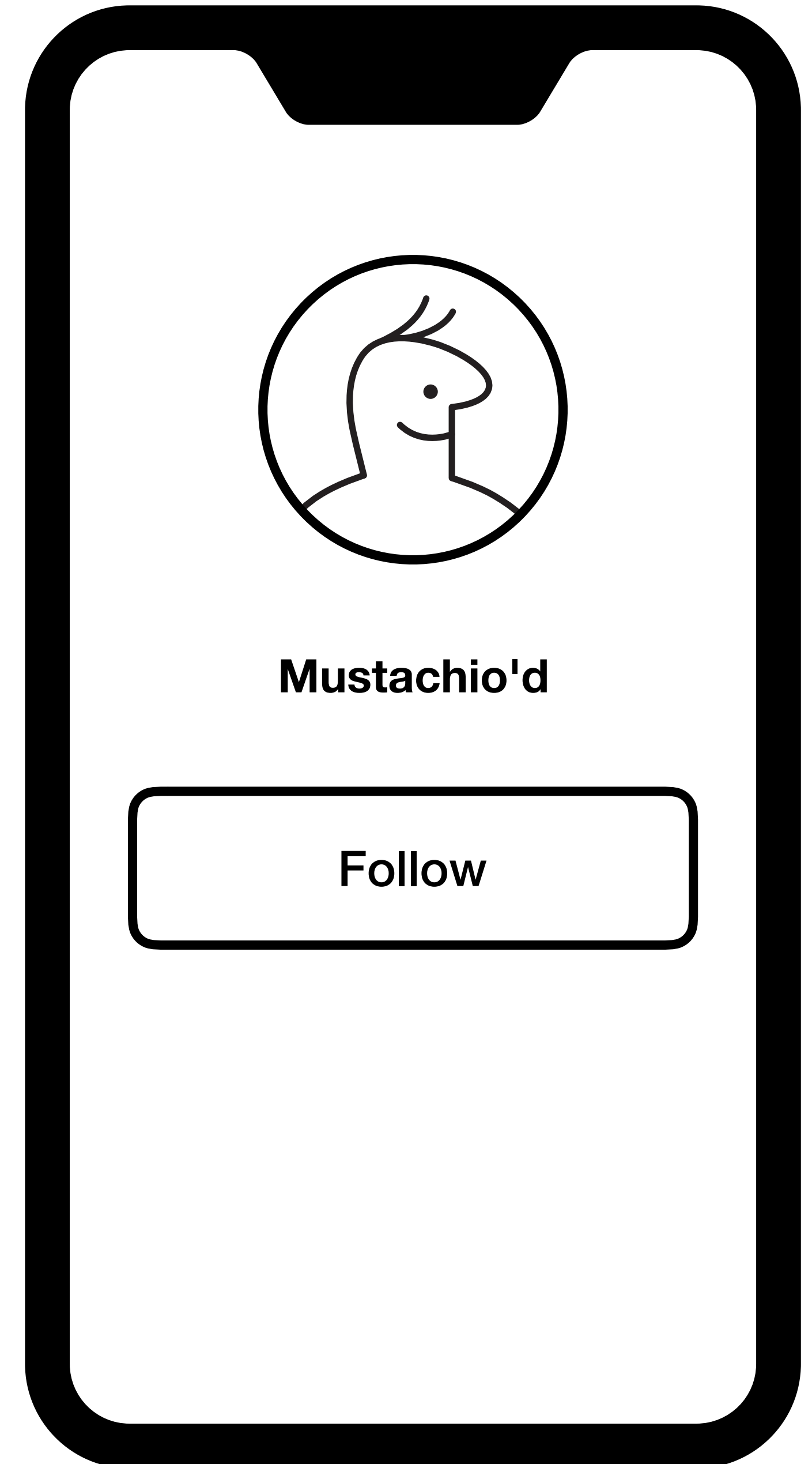
Tap Me!

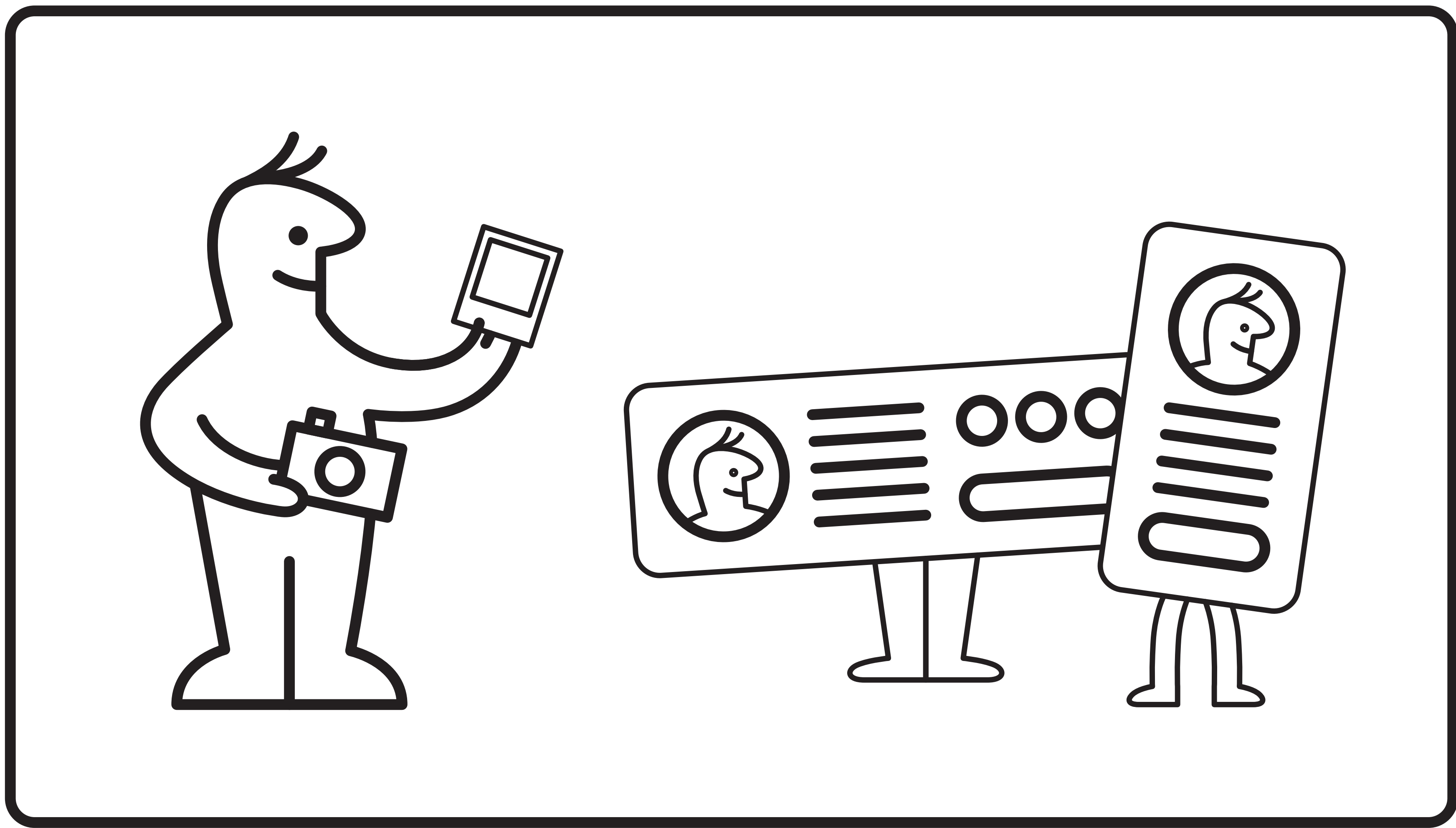
Tap Me!

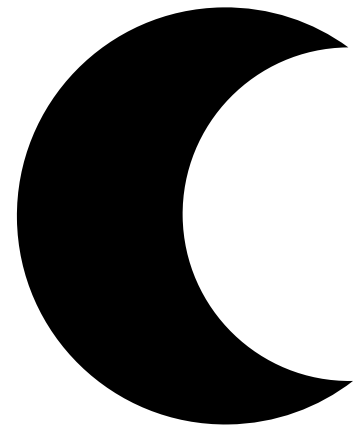
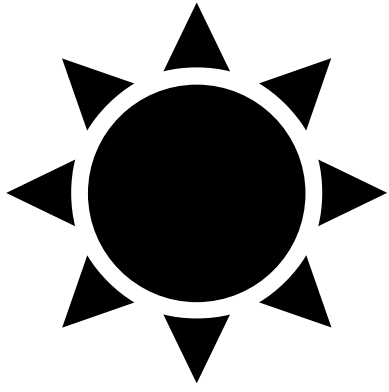
Tap Me!

```
#if canImport(SwiftUI) && DEBUG
import SwiftUI

@available(iOS 13.0, *)
struct ProfileViewController_Preview: PreviewProvider {
    static var previews: some View {
        let storyboard =
            UIStoryboard(name: "ProfileViewController",
                          bundle: nil)
        storyboard.instantiateInitialViewController { decoder in
            ProfileViewController(
                coder: decoder,
                name: "Johnny Appleseed",
                avatarImageURL: nil
            )
        }
    }
}
#endif
```



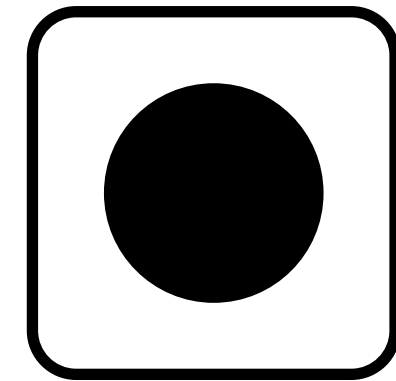
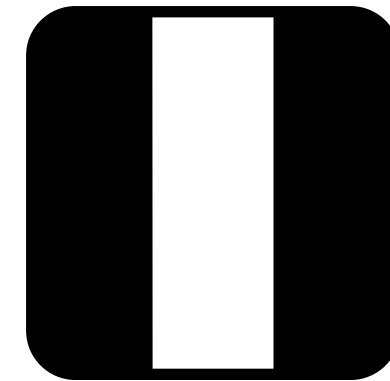
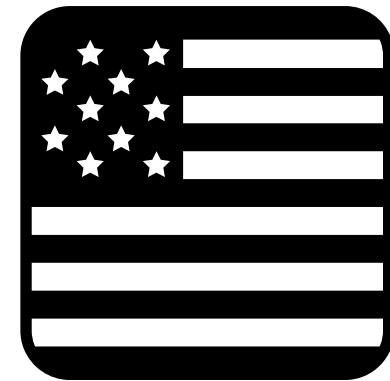
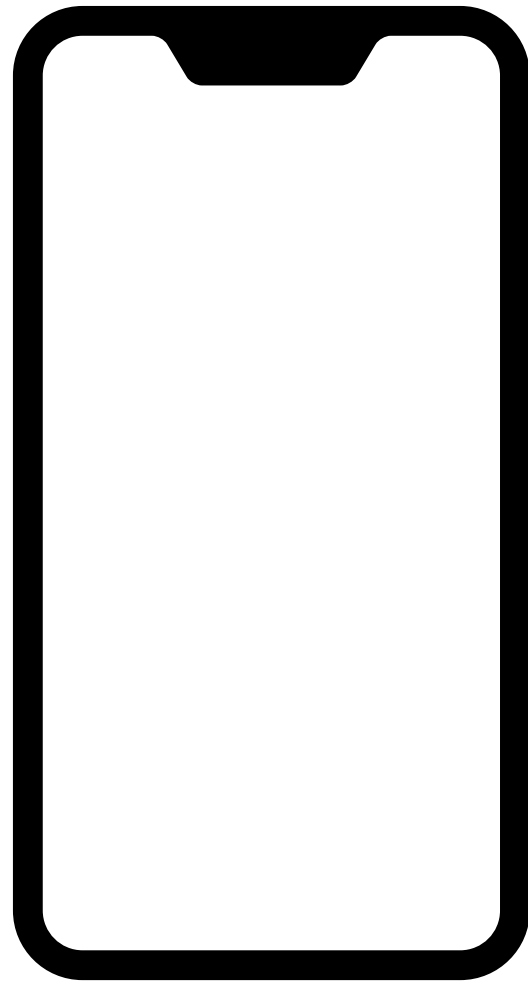
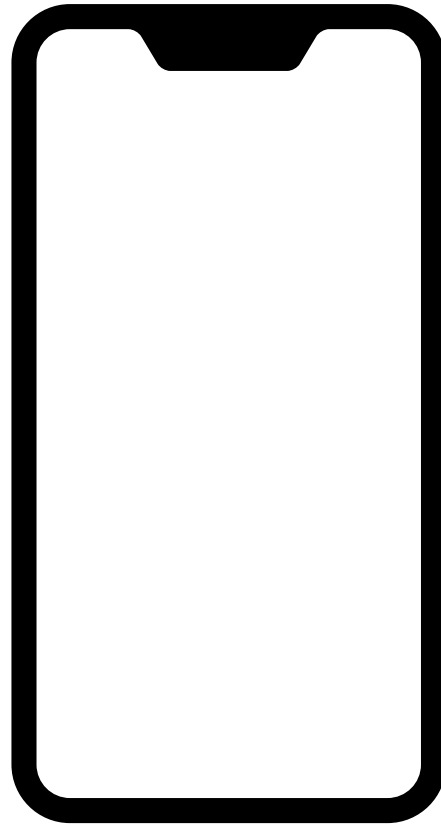
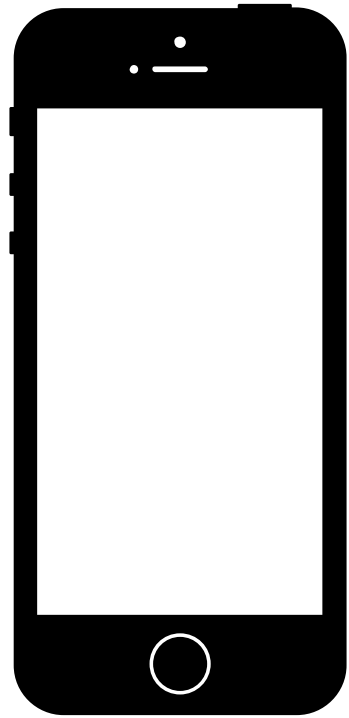




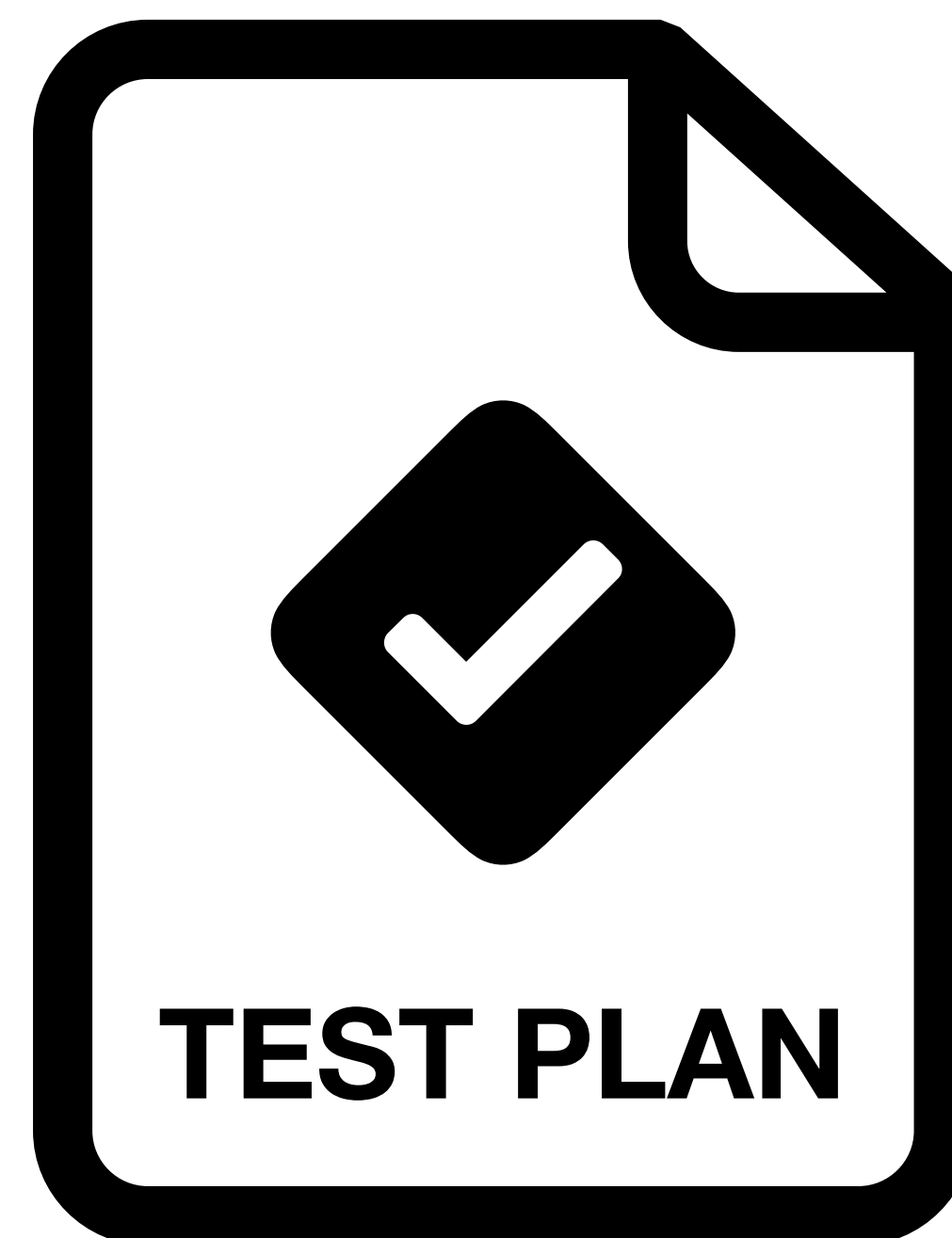
S

L

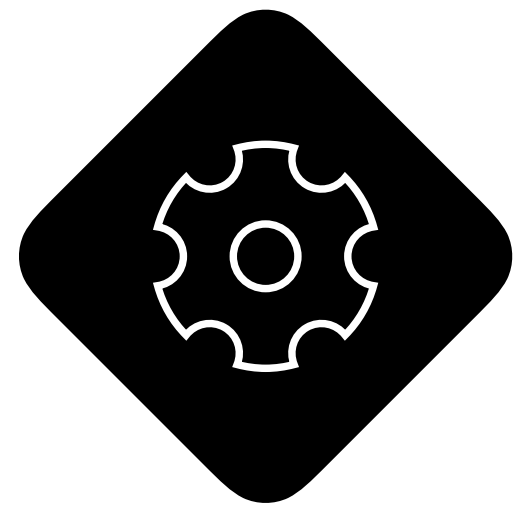
XXL



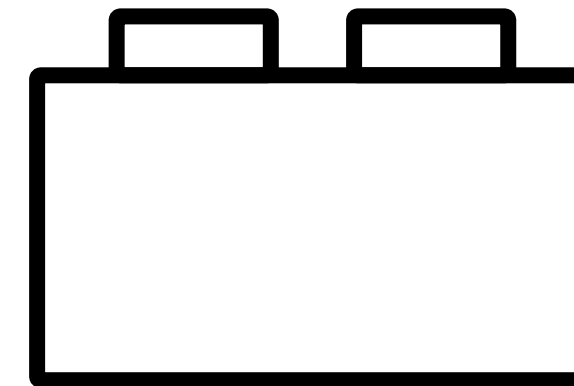




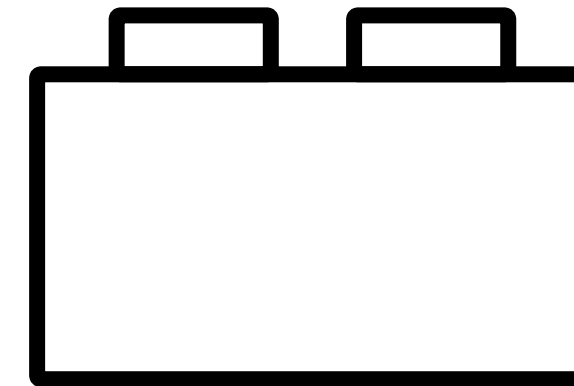
Integration.xctestplan



Default

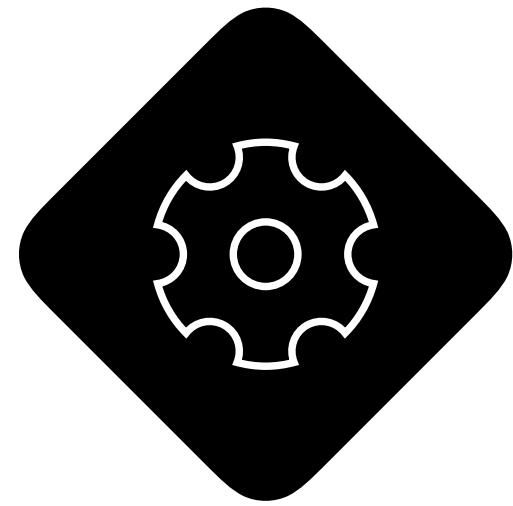


MyAppTests

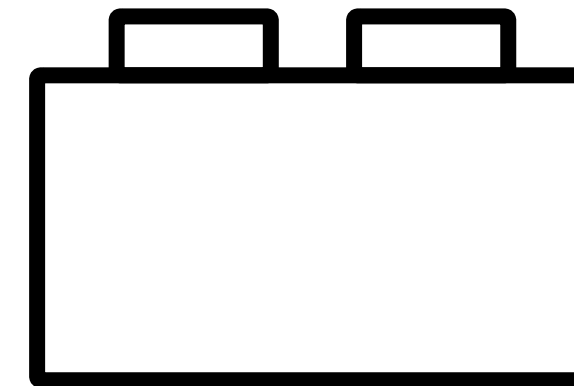


MyAppUITests

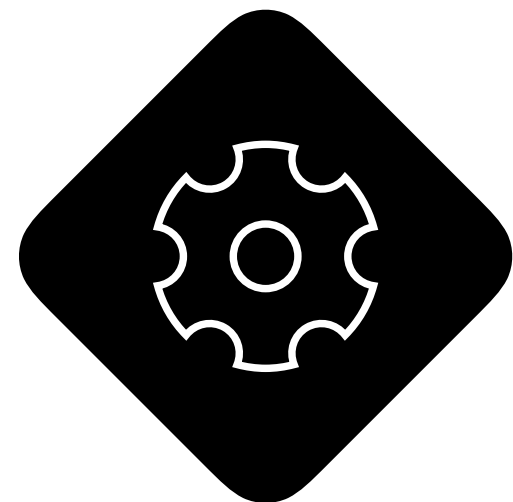
Accessibility.xctestplan



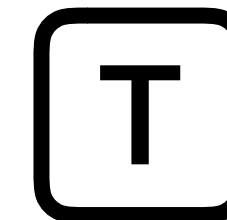
Small Type (S)



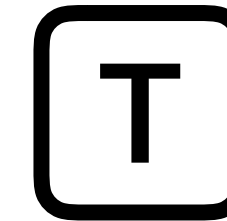
MyAppUITests



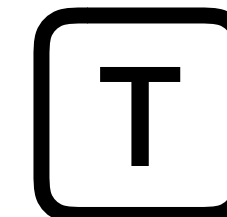
Extra-Large Type (XL)



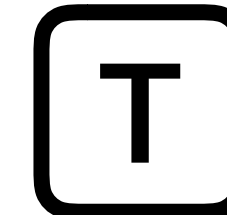
LoginTests



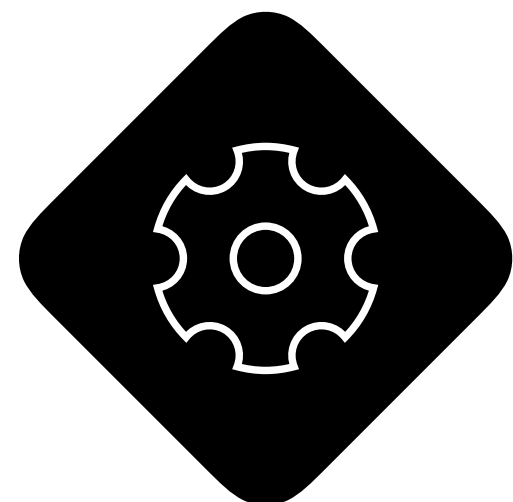
SignUpTests



ProfileTests

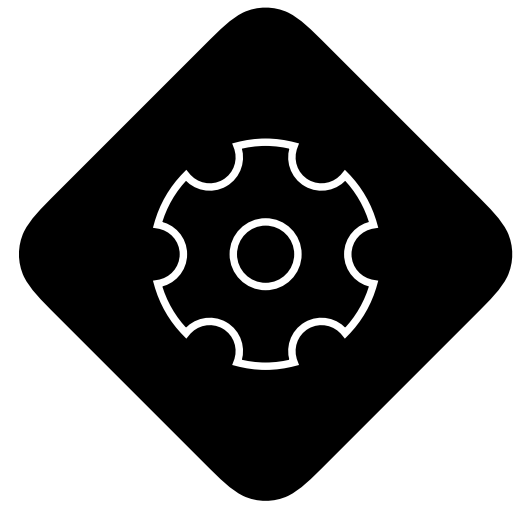


AccessibilityTests

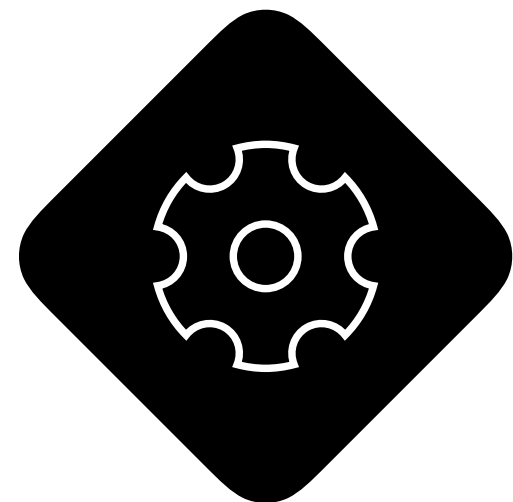


High Contrast

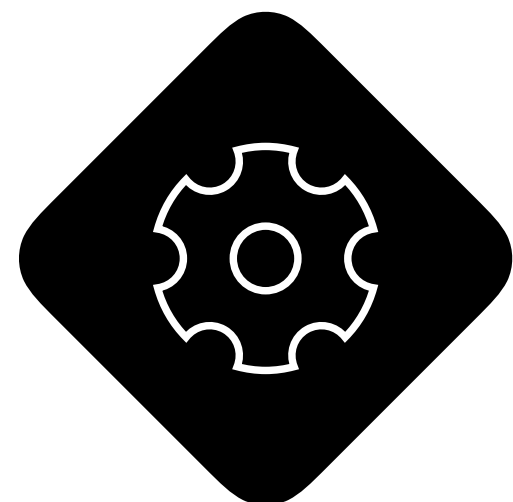
Localization.xctestplan



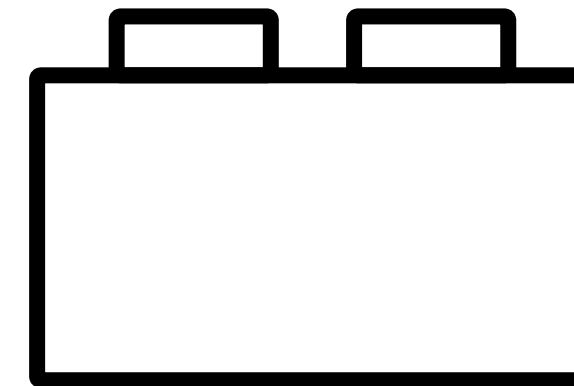
English (en-US)



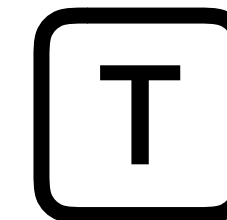
Italian (it-IT)



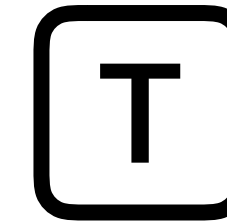
Japanese (ja-JP)



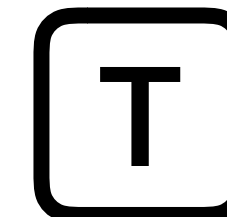
MyAppUITests



LoginTests



SignUpTests

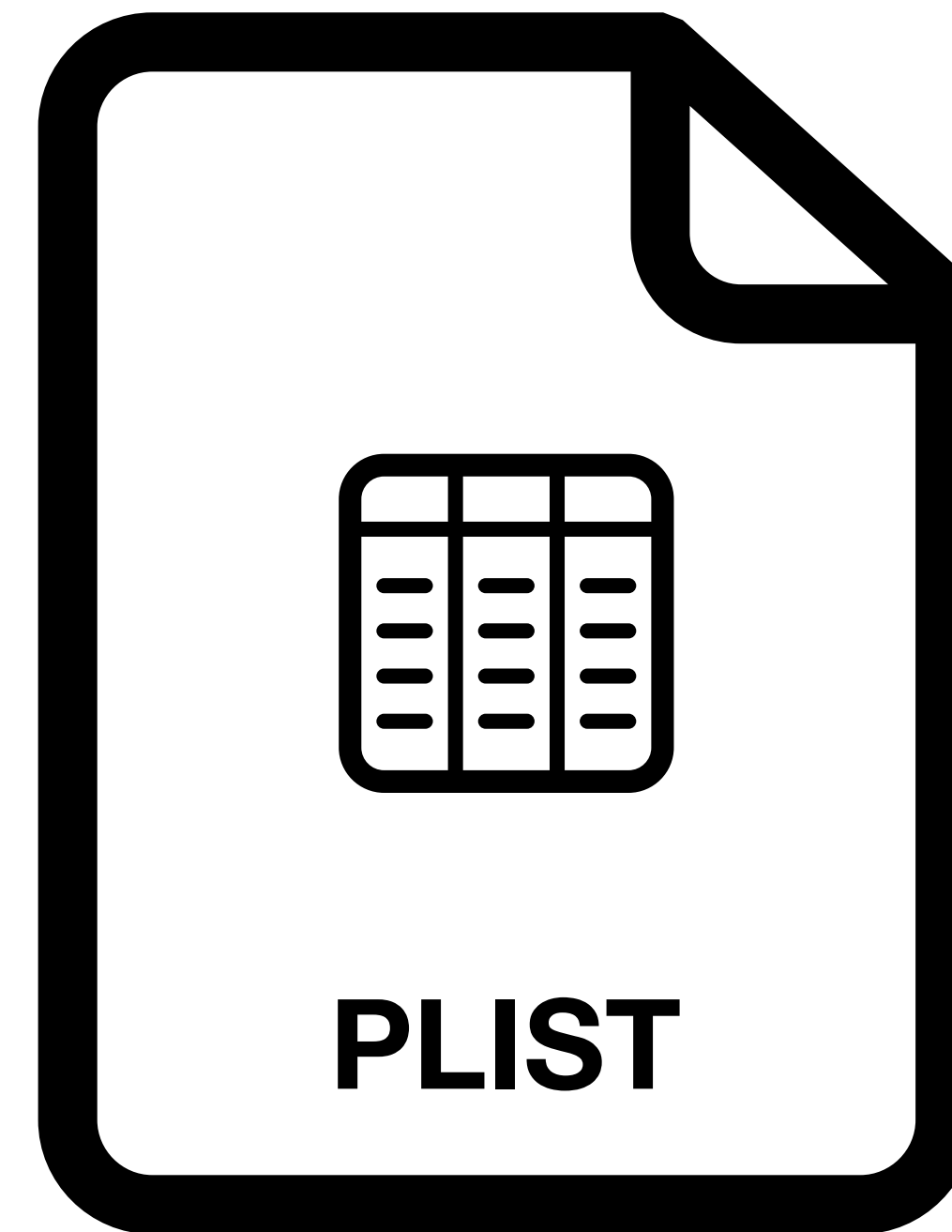
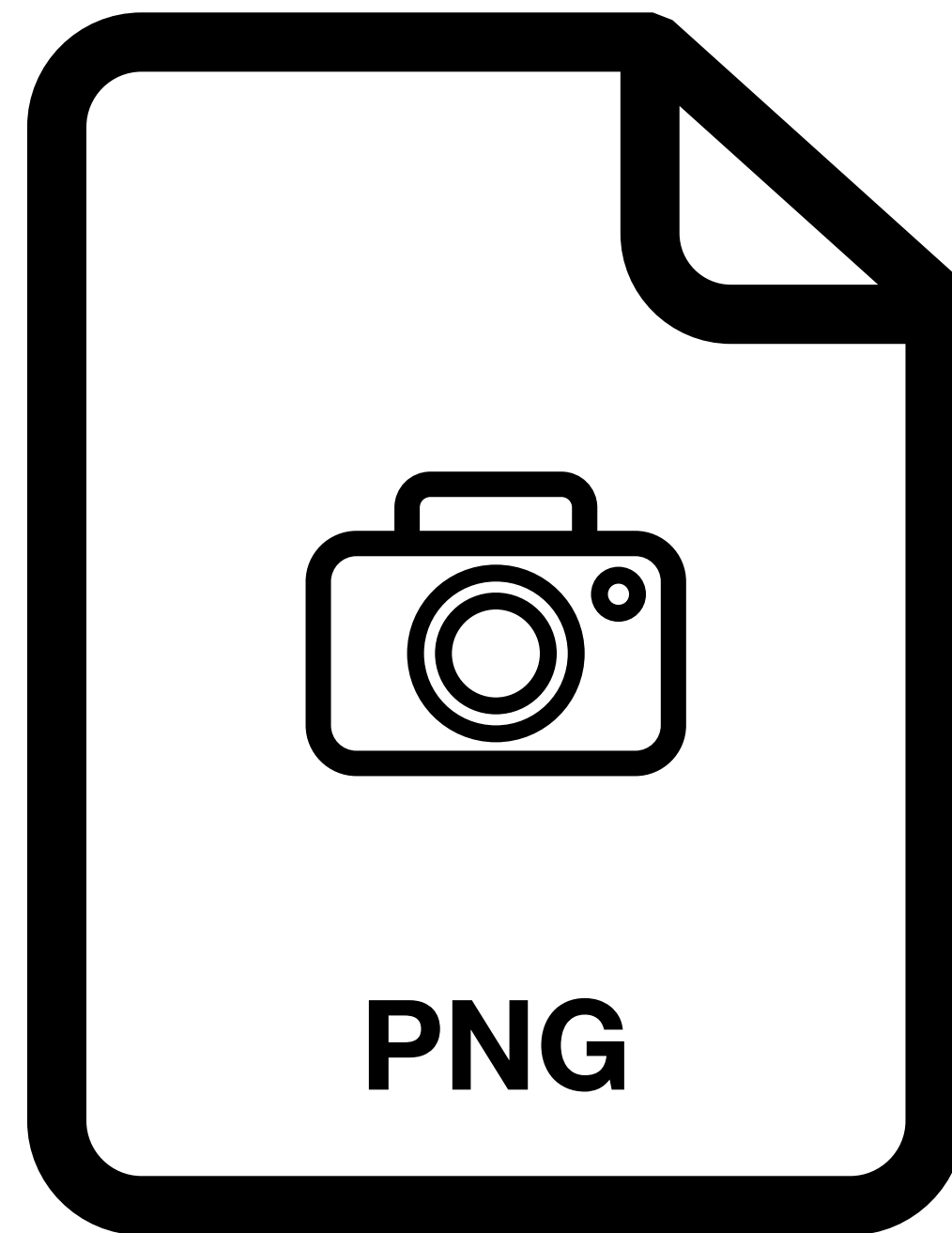


ProfileTests

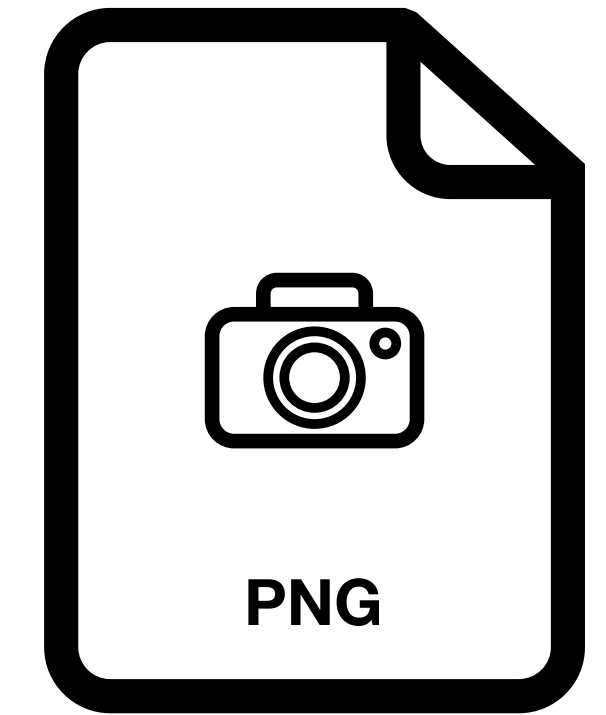
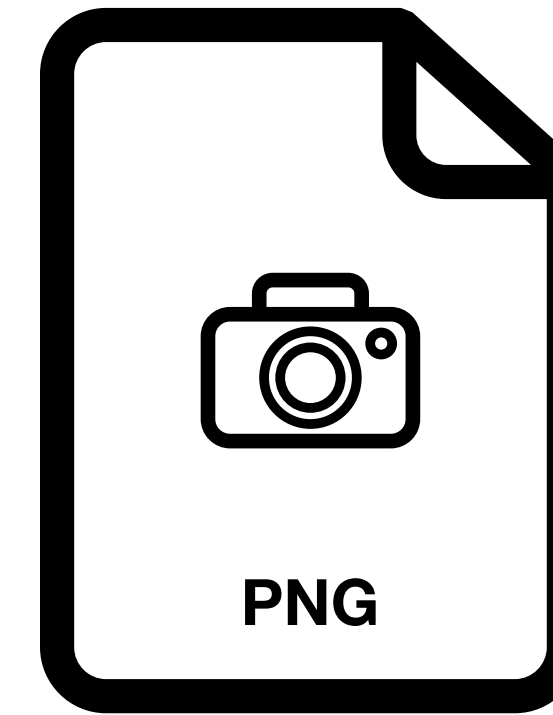
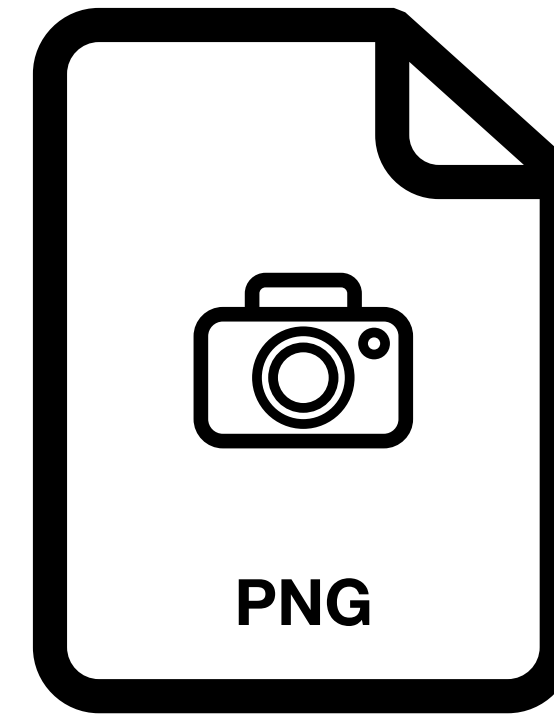
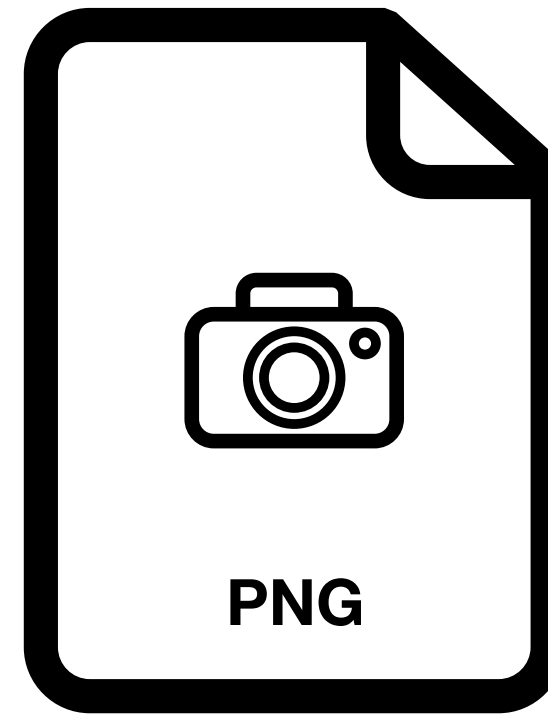
XCTSnapshot + XCTAttachment

```
let screenshot = app.windows.firstMatch.screenshot()  
let attachment = XCTAttachment(screenshot: screenshot)  
attachment.lifetime = .keepAlways  
add(attachment)
```

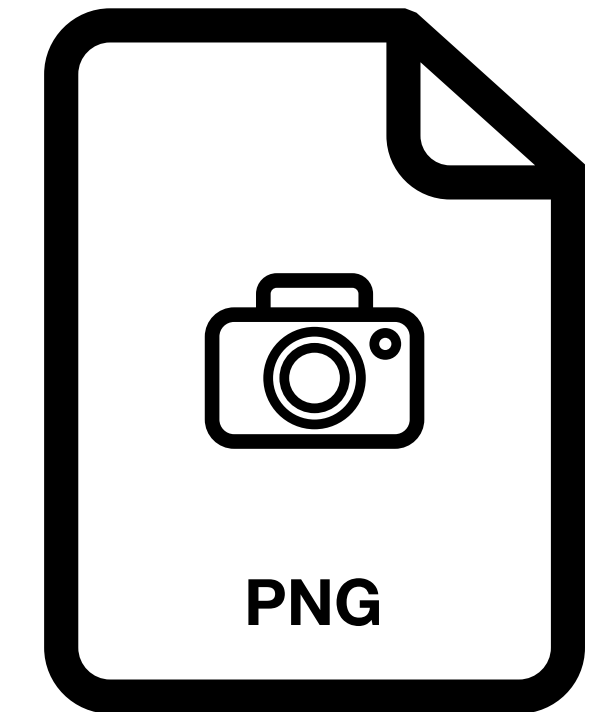
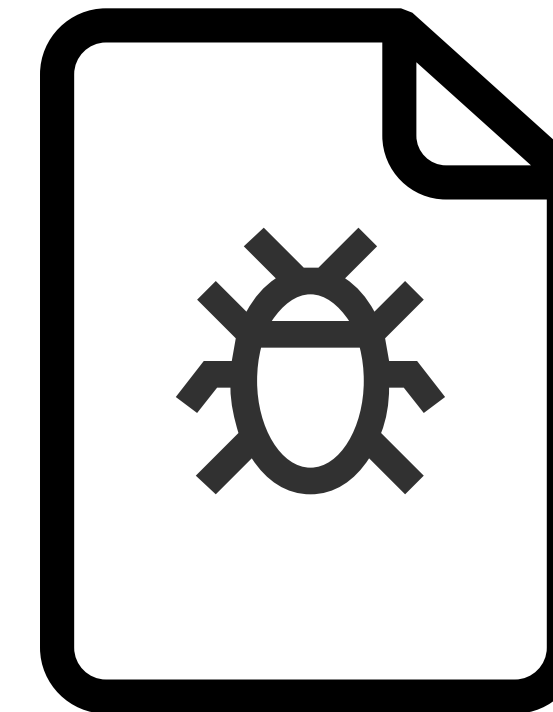
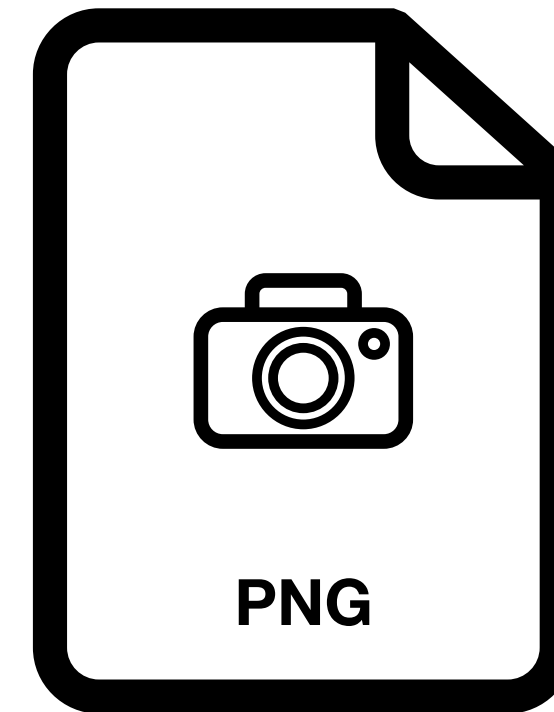
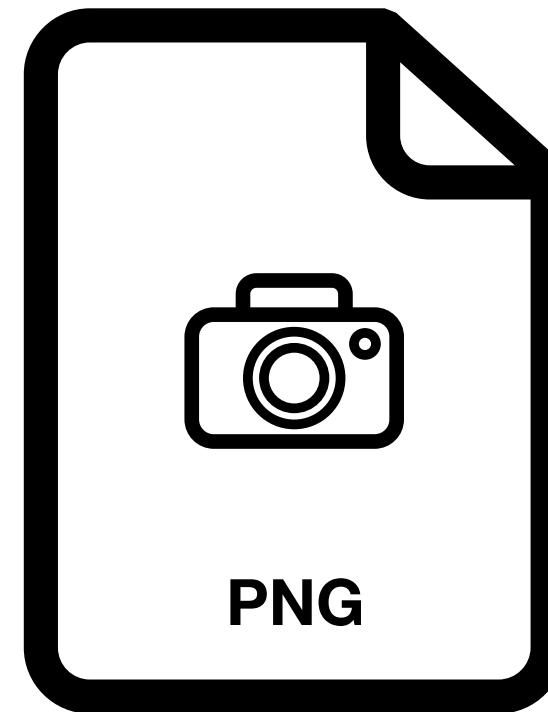
xcparse



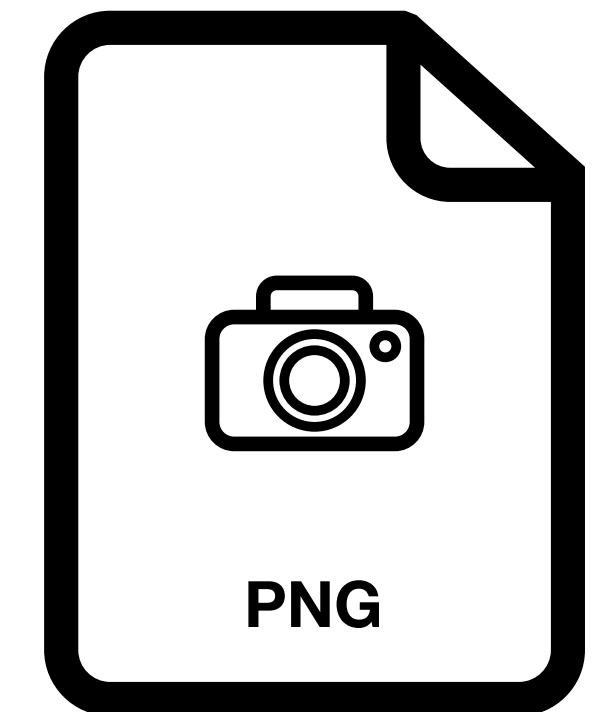
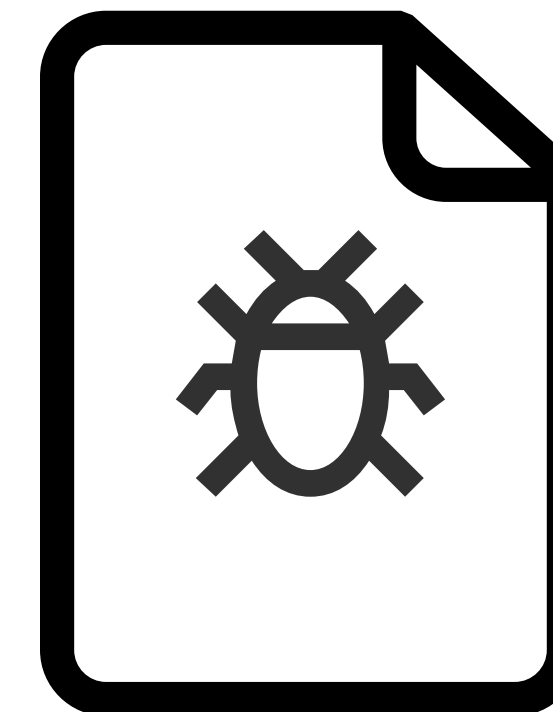
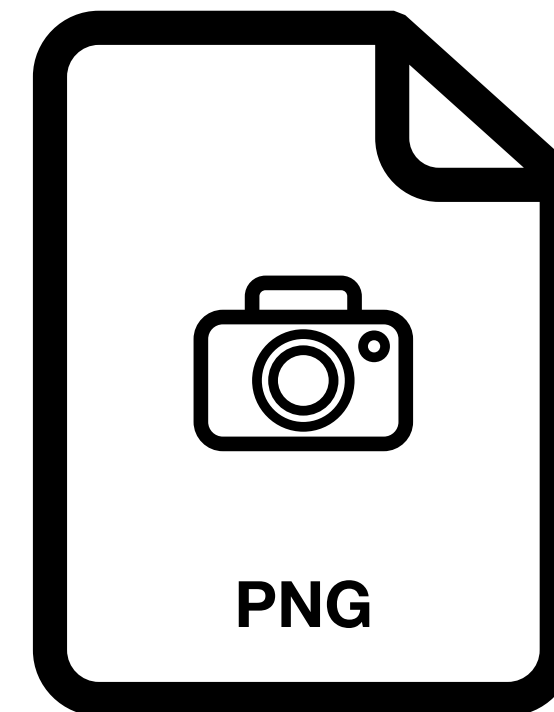
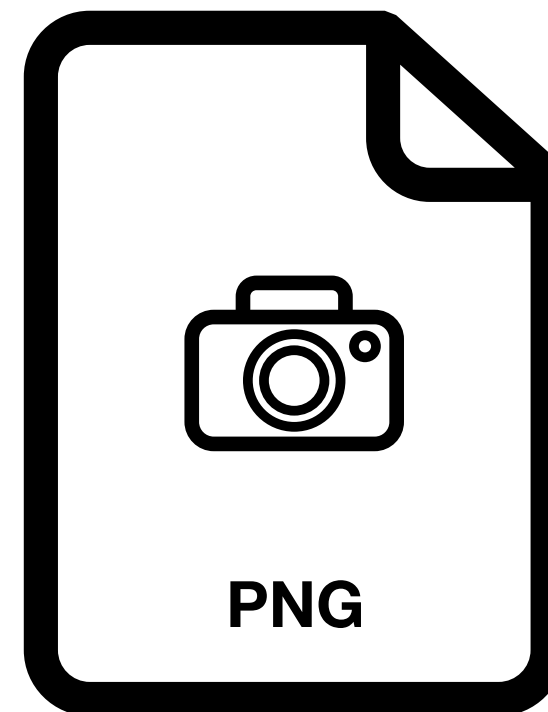
Lunedì



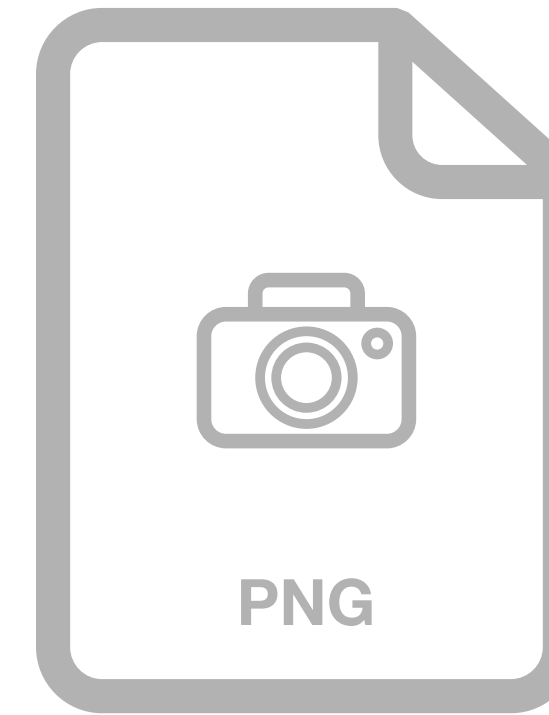
Martedì



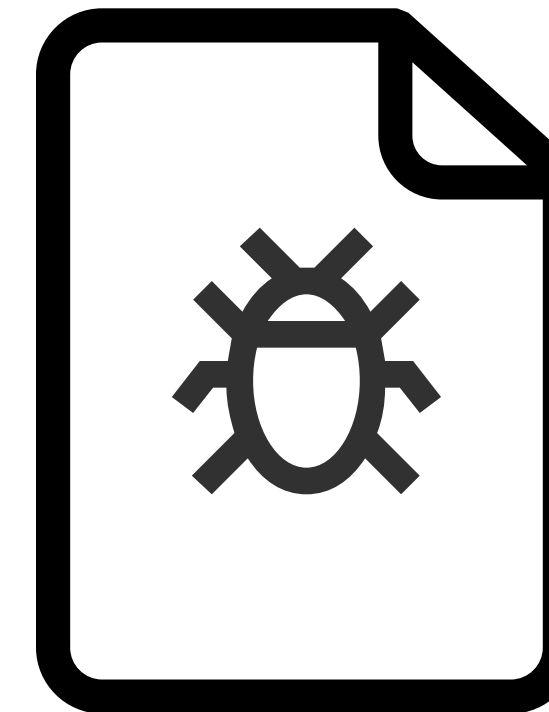
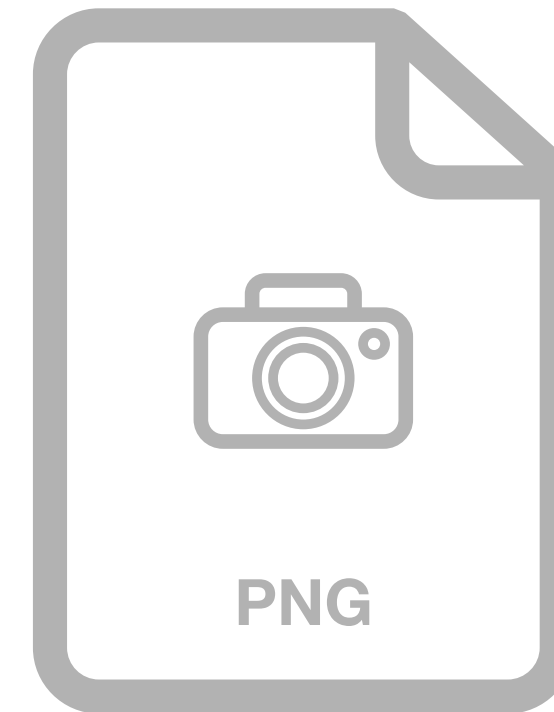
Mercoledì



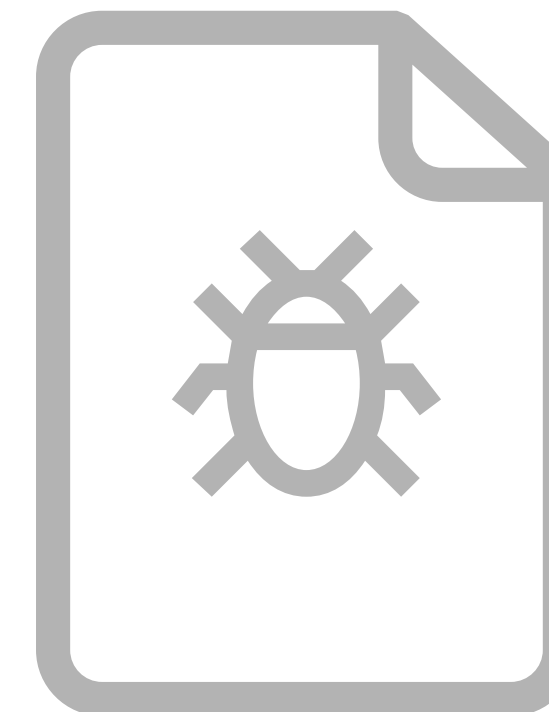
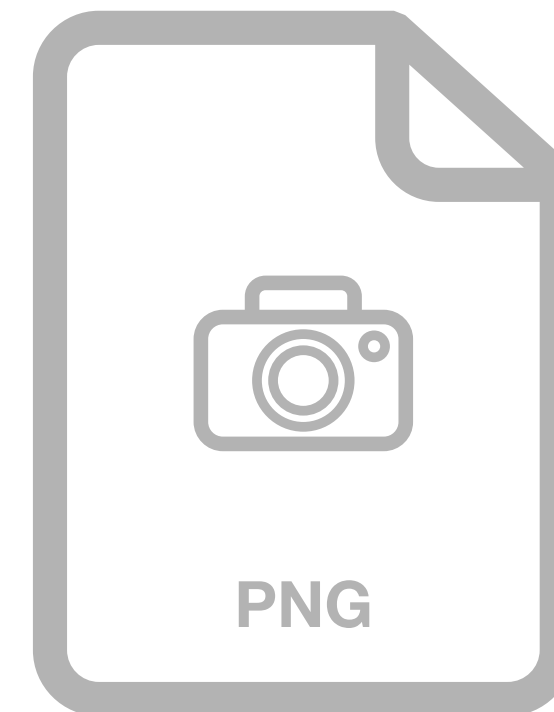
Lunedì



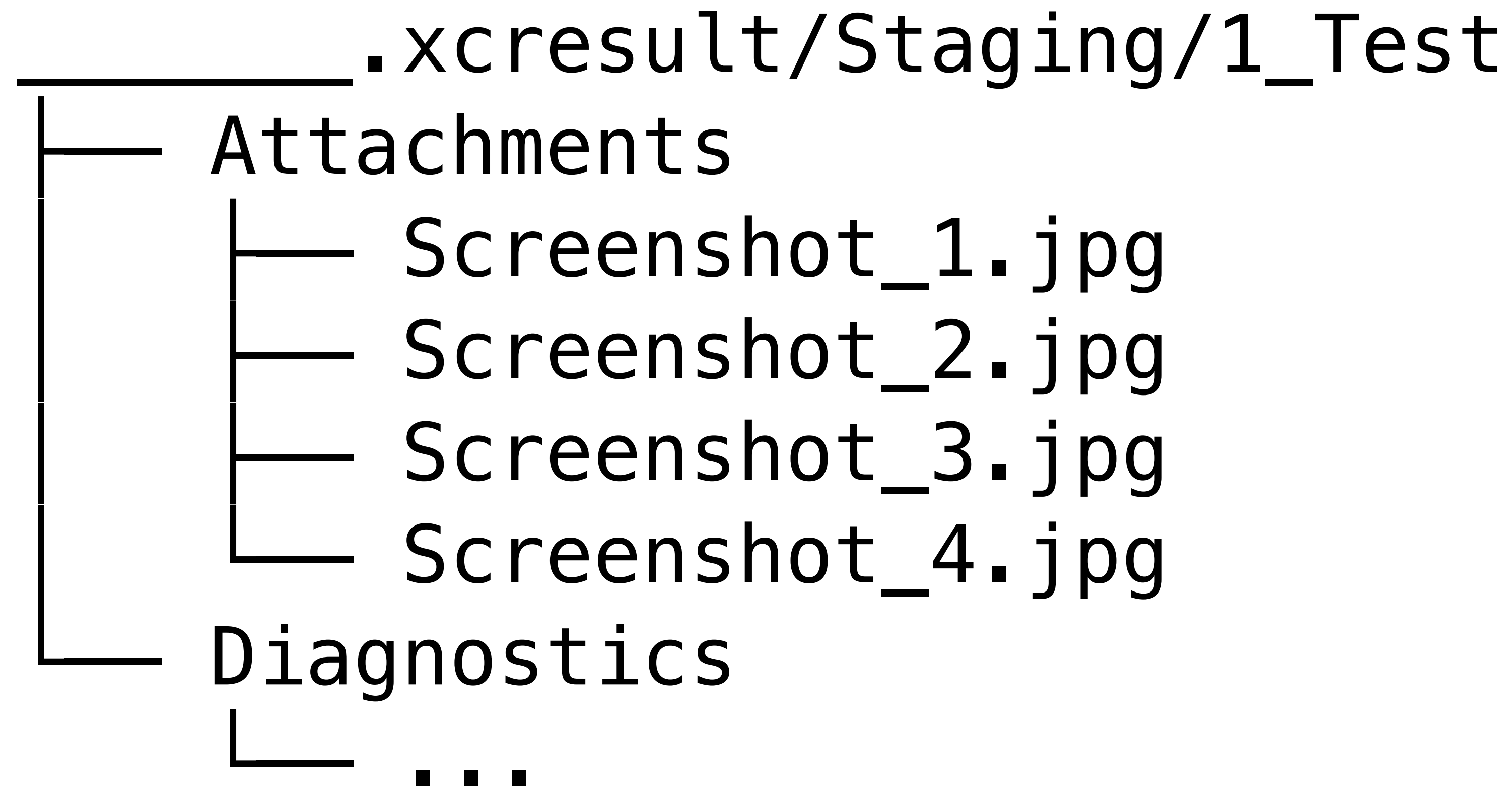
Martedì



Mercoledì



XCTSnapshot + XCTAttachment



XCTSnapshot + XCTAttachment

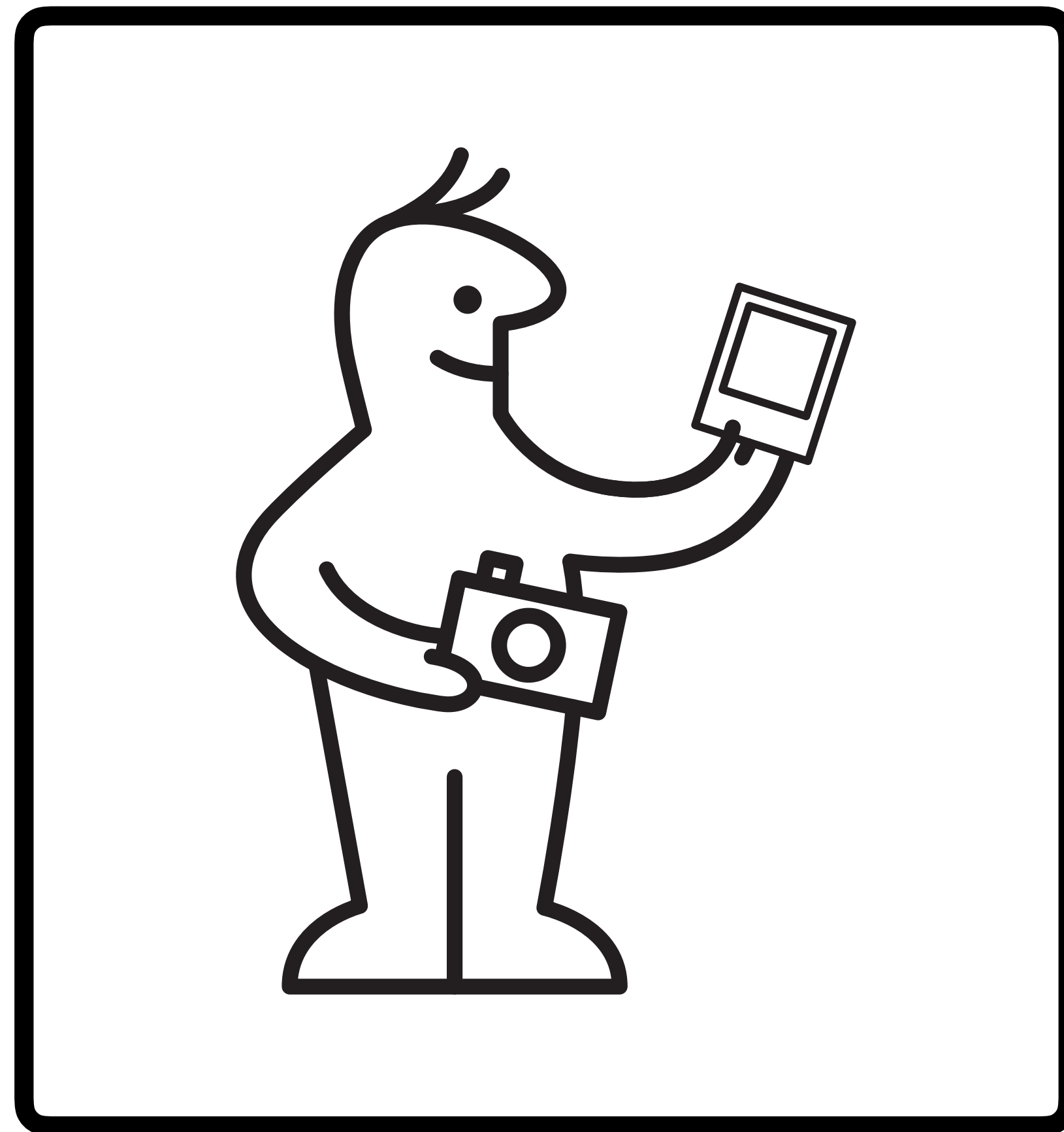
└── .xcresult
 └── Data

- └── data.0~00oA6ZYLWVgTgowtmxfnQp5IL5fuLGFy0i_4_VIB0MfR98gLVPf08i6xhVdRvCgrG9MXogo5pmU3jisYjKB1Ww==
- └── data.0~0CiGEuyJeMF58YgEx7no2BtJyxU_noXKX0b0bJ1AD934J4Gd9L0tQjIyB6j9ZmghlI0YiXng_o_7KL0XQmzQUA==
- └── data.0~0Db2ZUM80e_-Tk9azQbJNKfSGuV00L0w-0oaQH7q2WNYNv8-lk_oiSawF4l0C071IPSCM0u_TTIJM7D3V7izvA==
- └── data.0~1FBRTgTWx00rxqZq8rM52GAFeea8SEJT4_Tj0on2gC6DdUvZ87JfTcNYCSwvFpW_n60WYlVoM0nDGwC6AIuNaw==

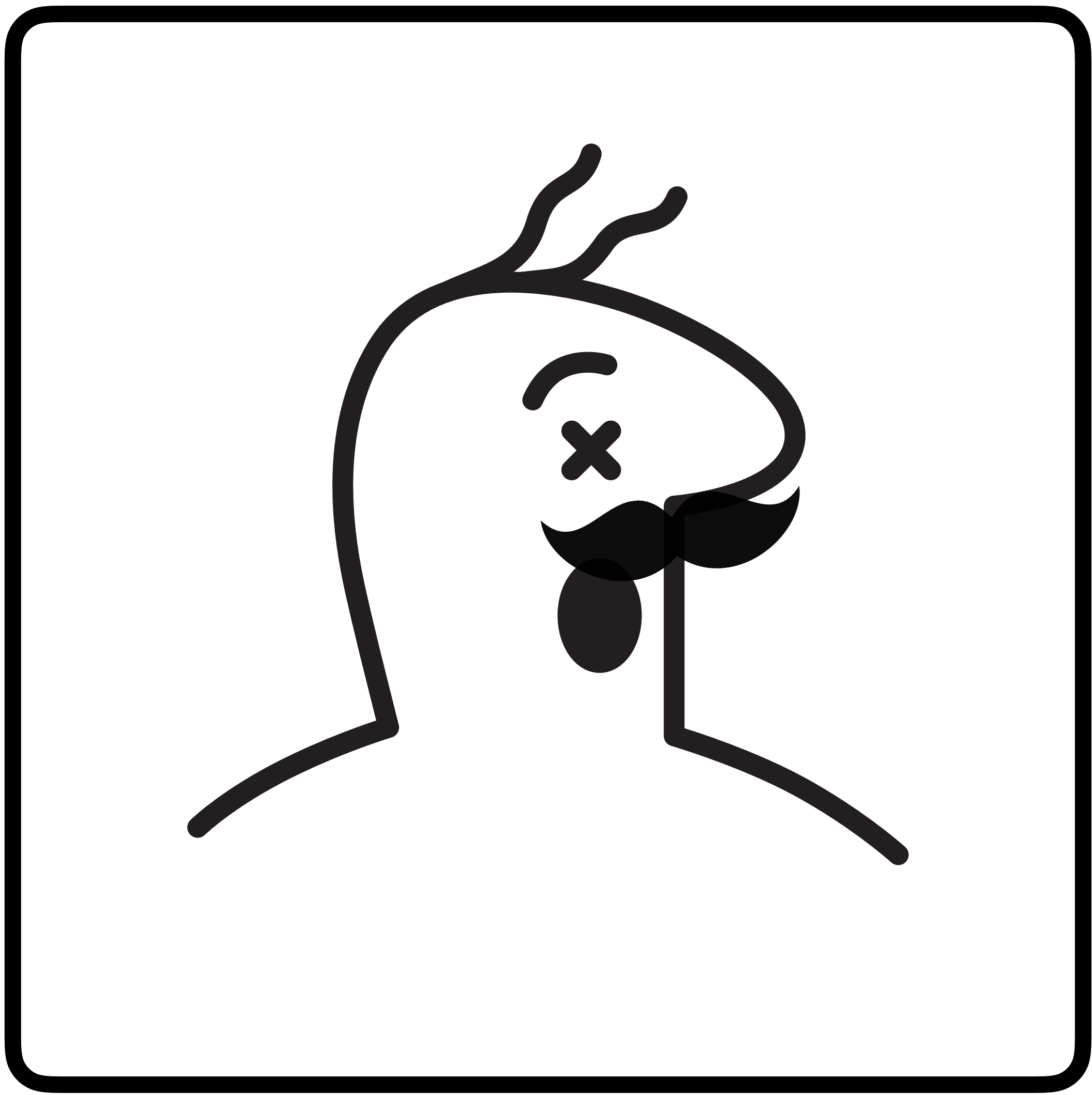
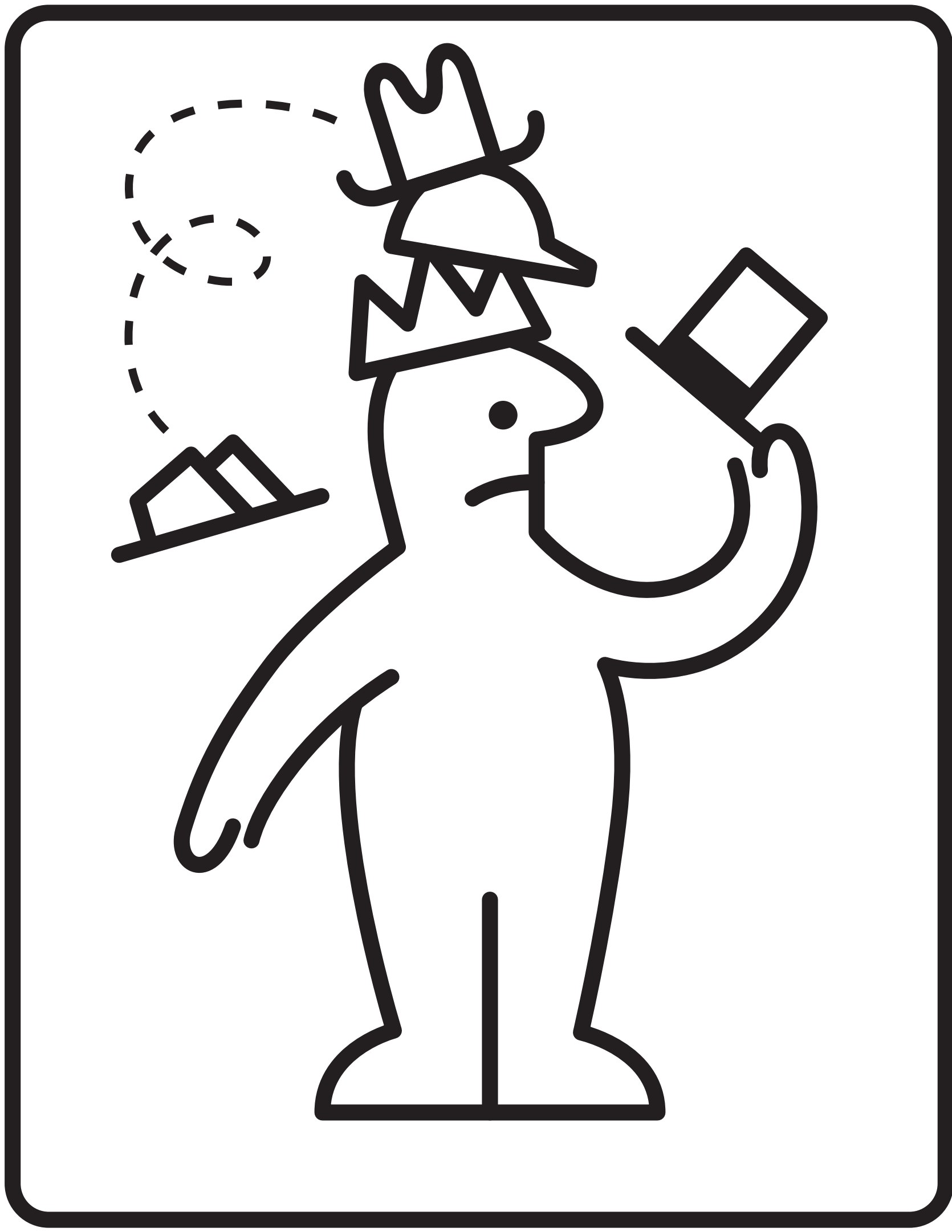
xcparse

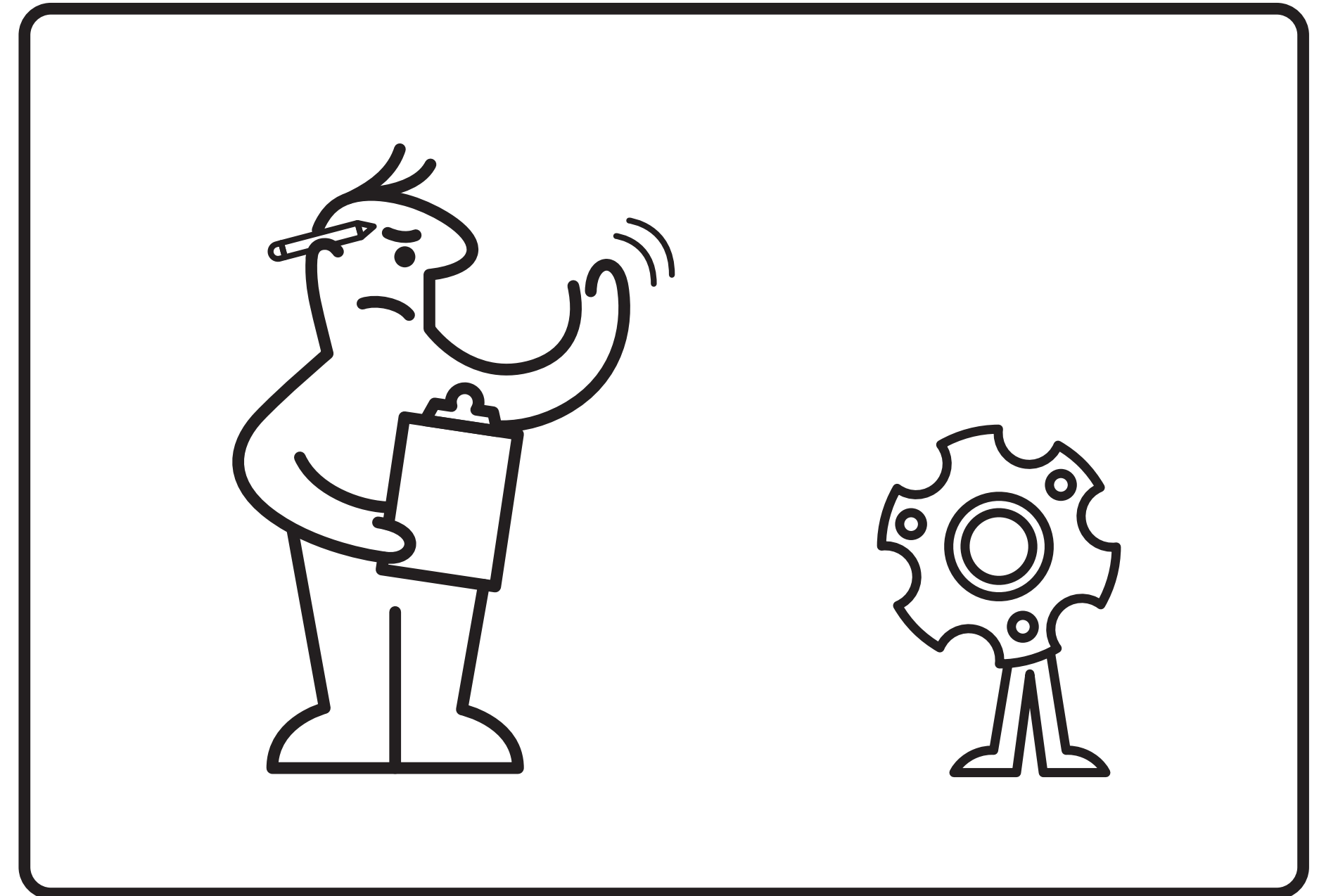
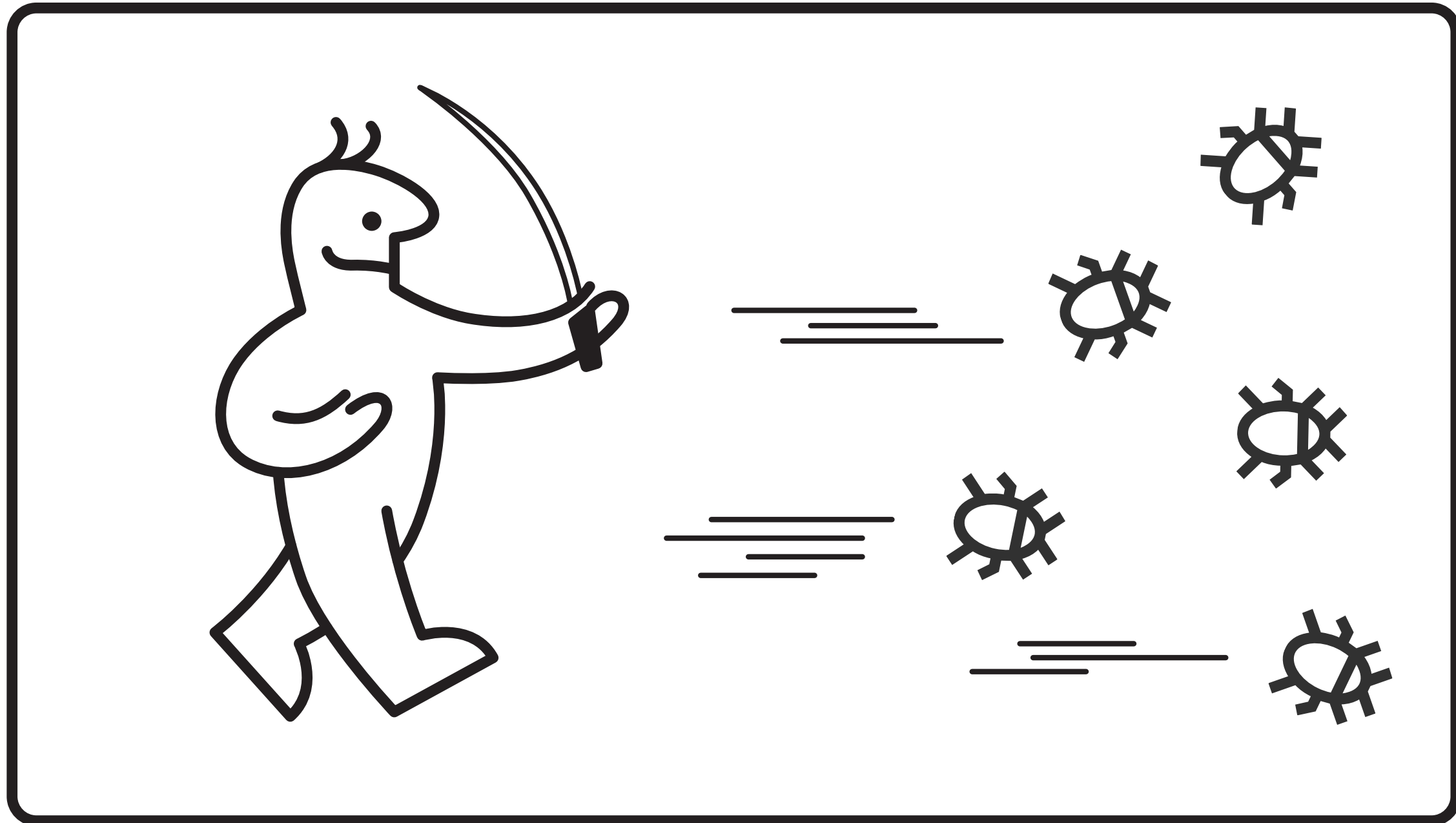
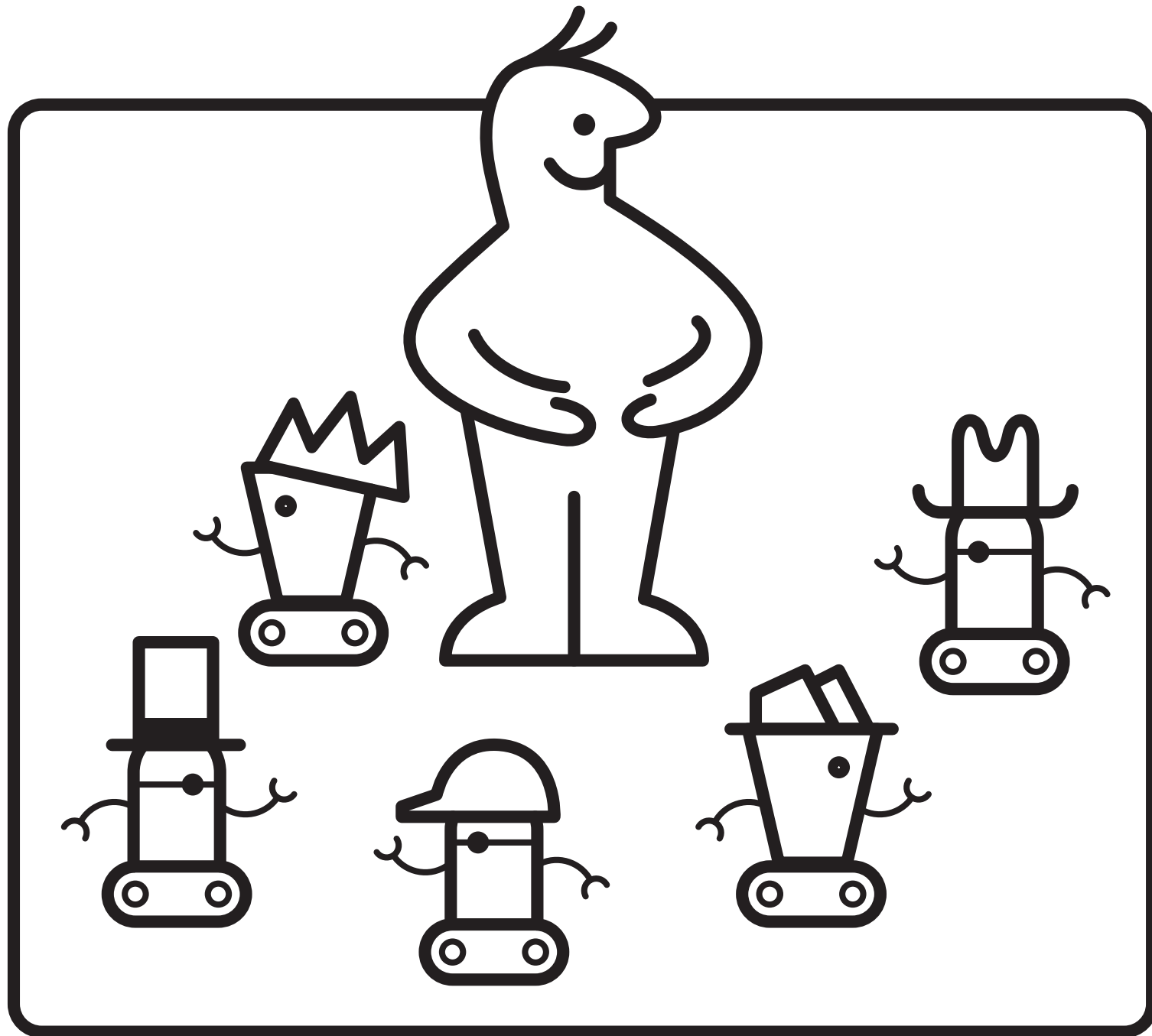
```
$ brew install chargepoint/xcparse/xcparse
```

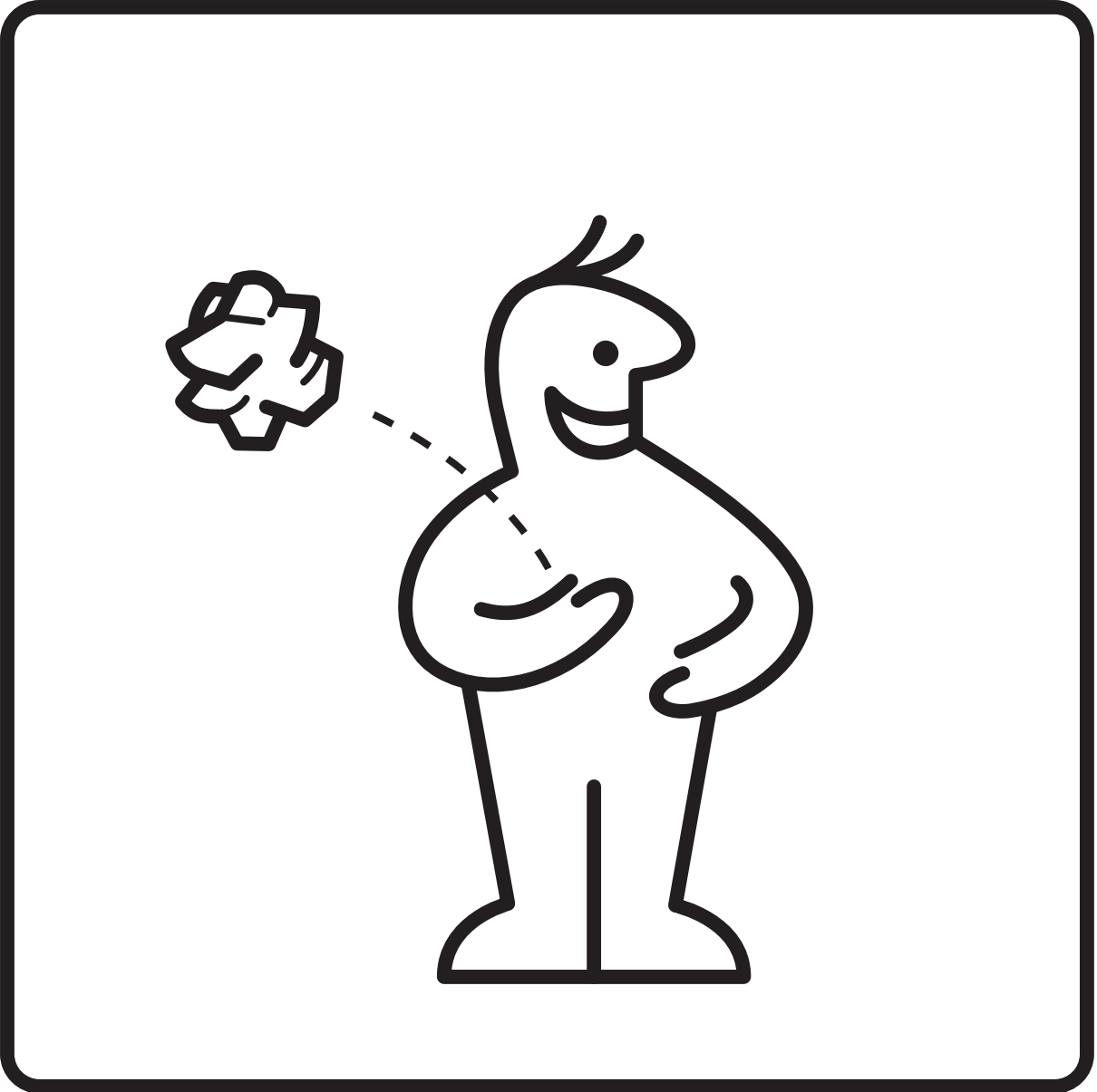
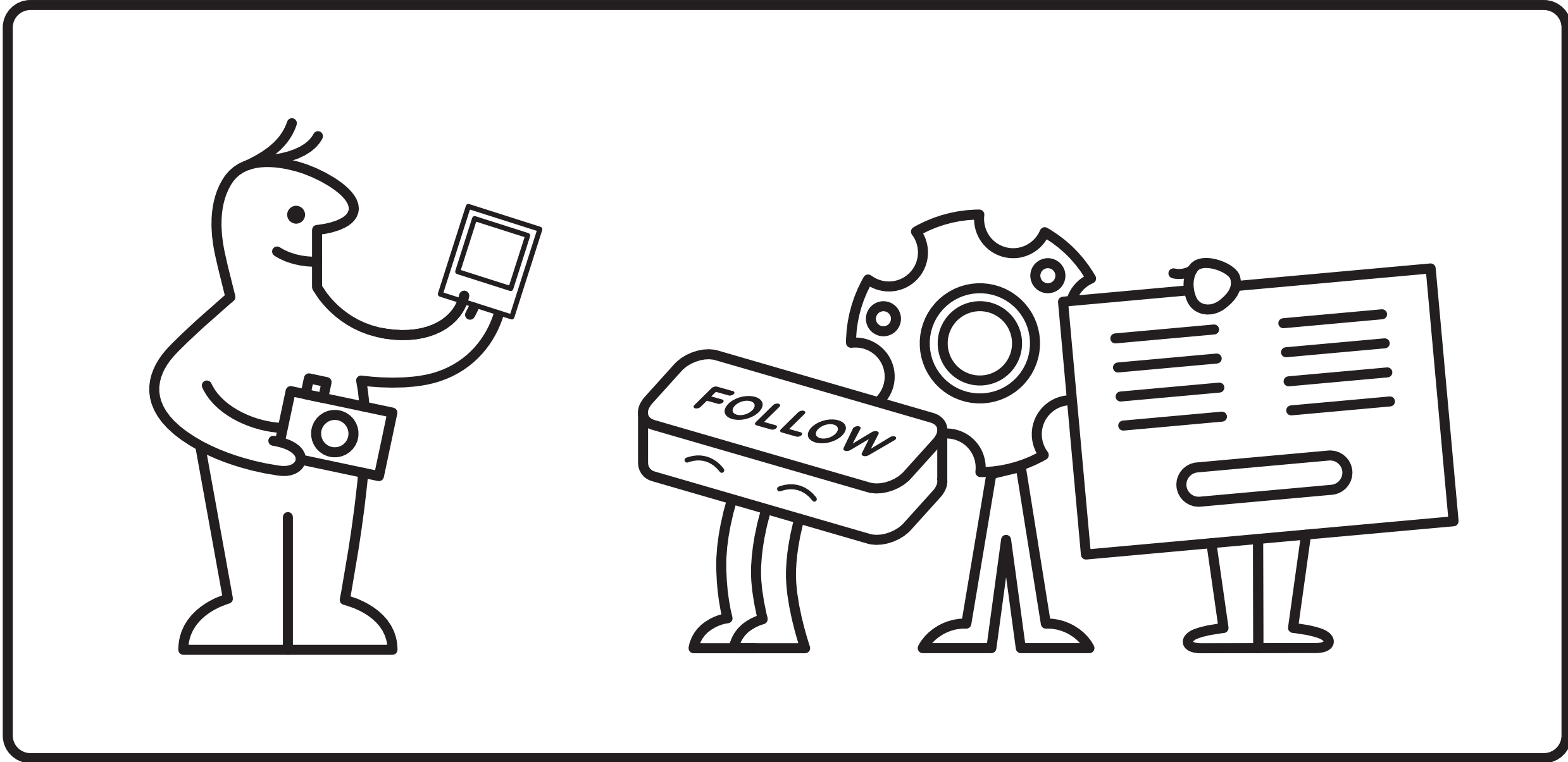
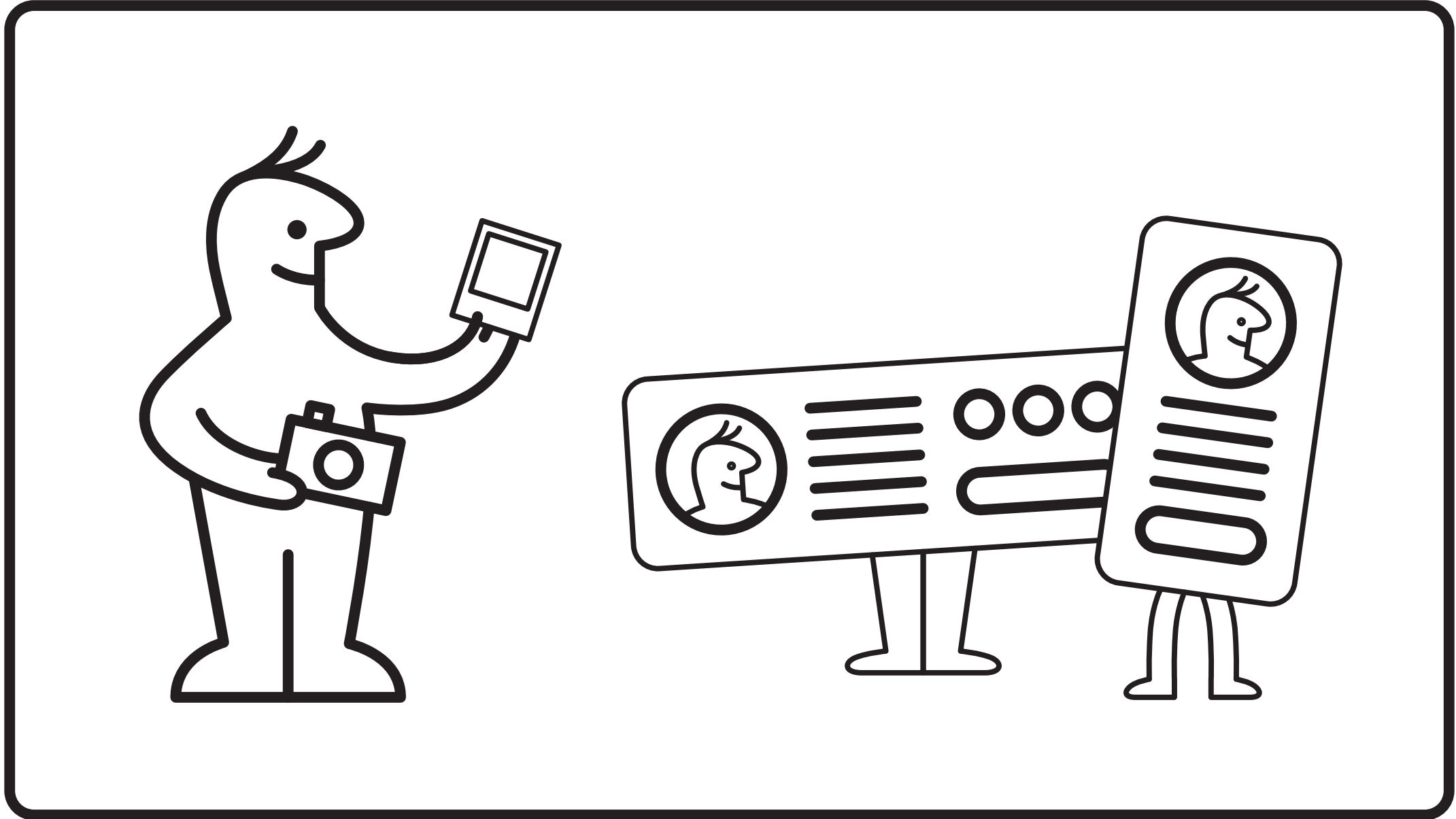
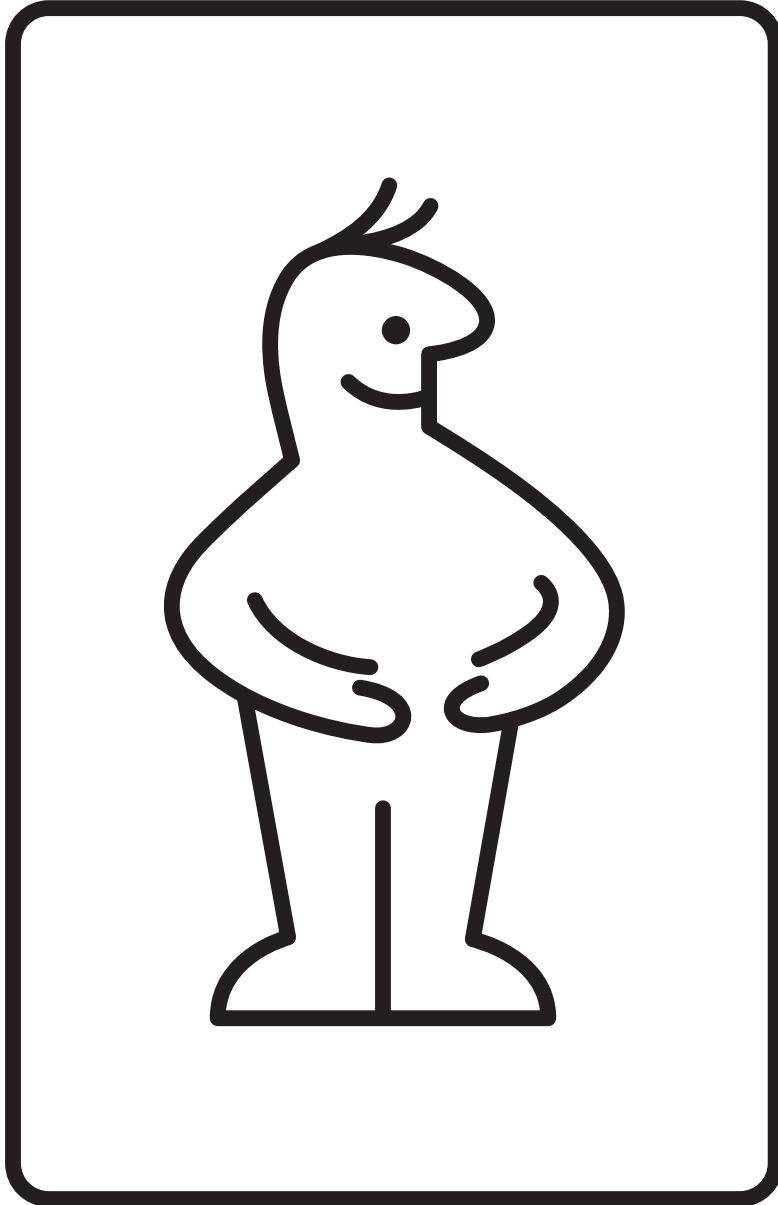
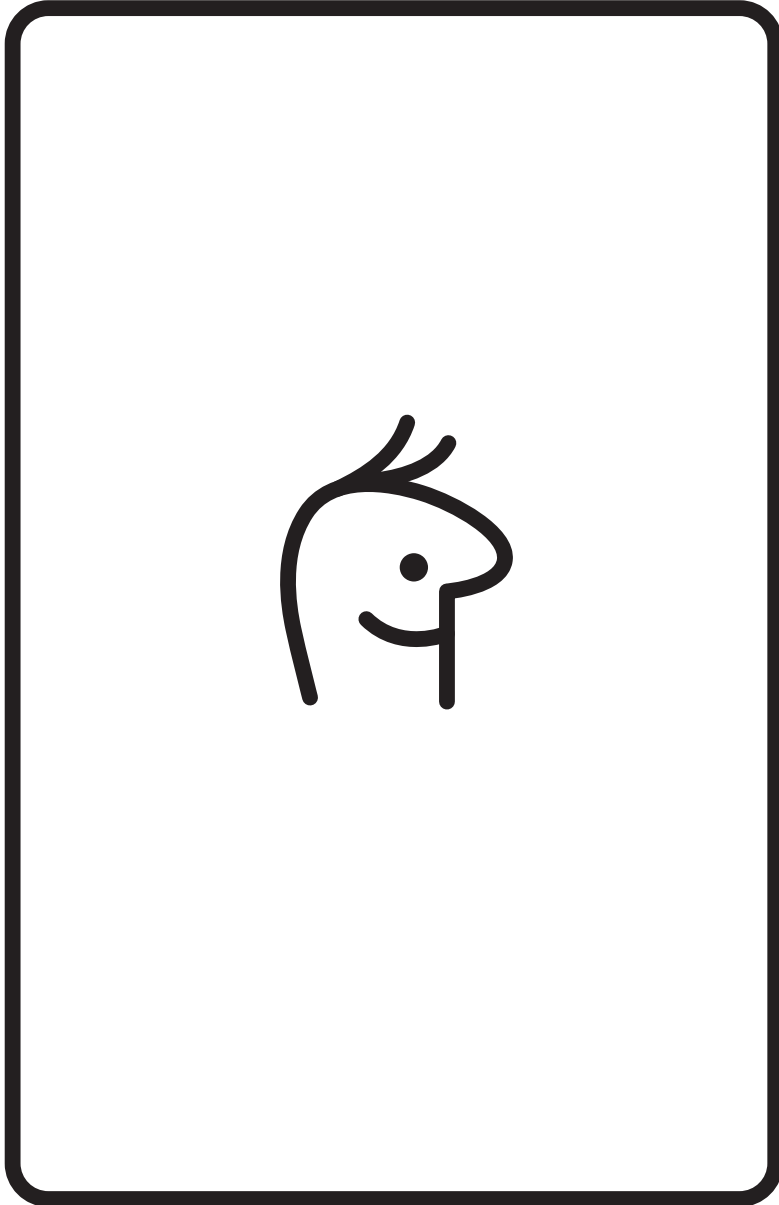
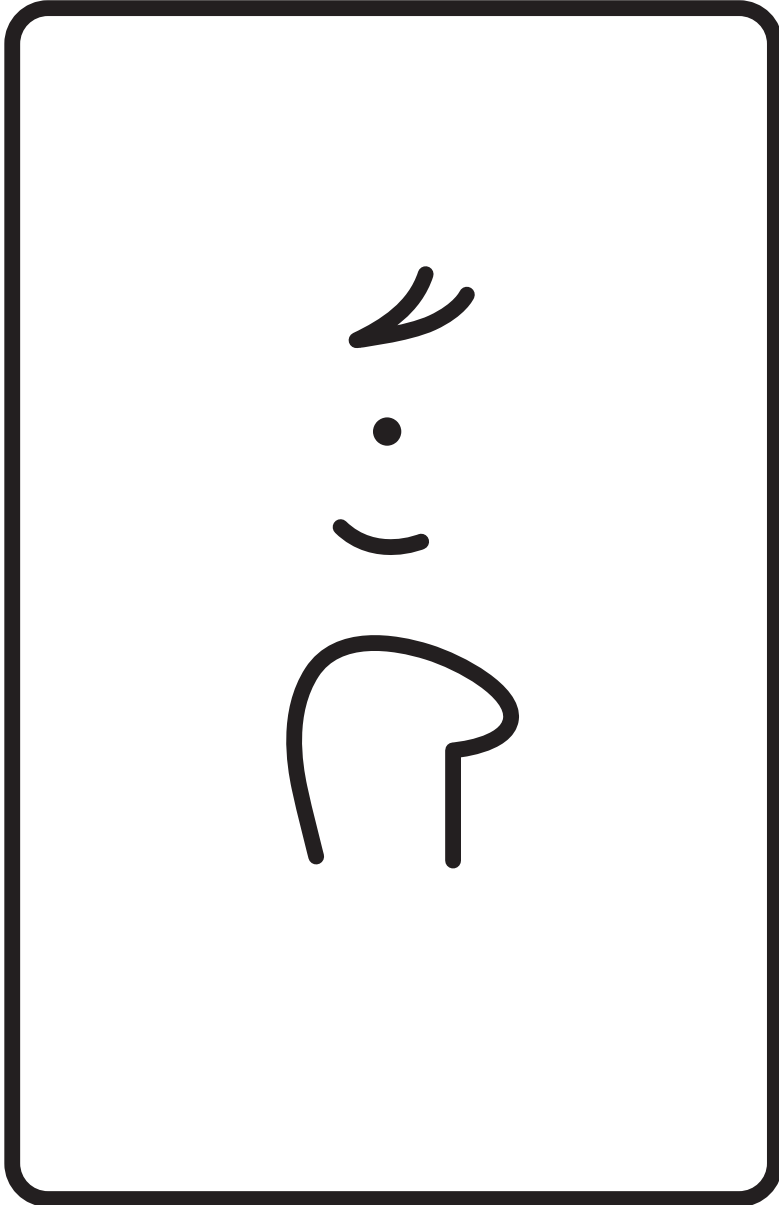
```
$ xcderiveddata | \  
  xargs -I{} find {} -name '*.xcresult' | \  
  xargs -I{} xcparse -s {} ./output
```



Ricapitolare







swift-format	github.com/apple/swift-format	ibcolortool	github.com/NSHipster/ibcolortool
SwiftLint	github.com/realm/SwiftLint	xcperfect	github.com/mkchoi212/xcperfect
xiblint	github.com/lyft/xiblint	SwiftMonkey	github.com/zalando/SwiftMonkey
IBLinter	github.com/IBDecodable/IBLinter	xcprojectconfig	<i>(Coming Soon)</i>
scancode	github.com/nexB/scancode-toolkit	xcprojectlint	github.com/americanexpress/xcprojectlint
Periphery	github.com/peripheryapp/periphery	xcodegen	github.com/yonaskolb/XcodeGen
xcassetlint	<i>(Coming Soon)</i>	xhammer	github.com/pinterest/xhammer
xcassetscop	github.com/LalaInc/xcassetscop	mergepbx	github.com/simonwagner/mergepbx
xcderriveddata	github.com/NSHipster/xcderriveddata		



Grazie mille!