

Playground-Driven Development

by Felipe Espinoza



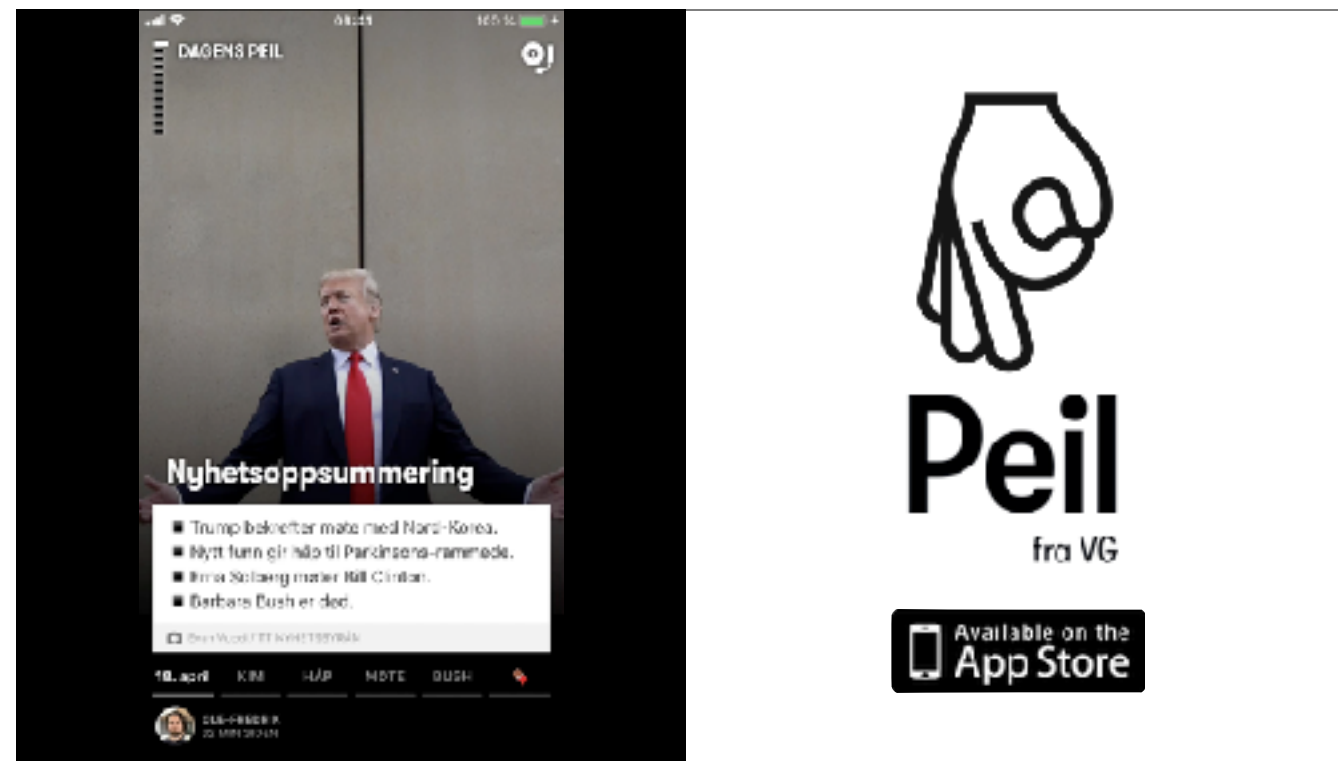
@fespinoza



@fespinozacast

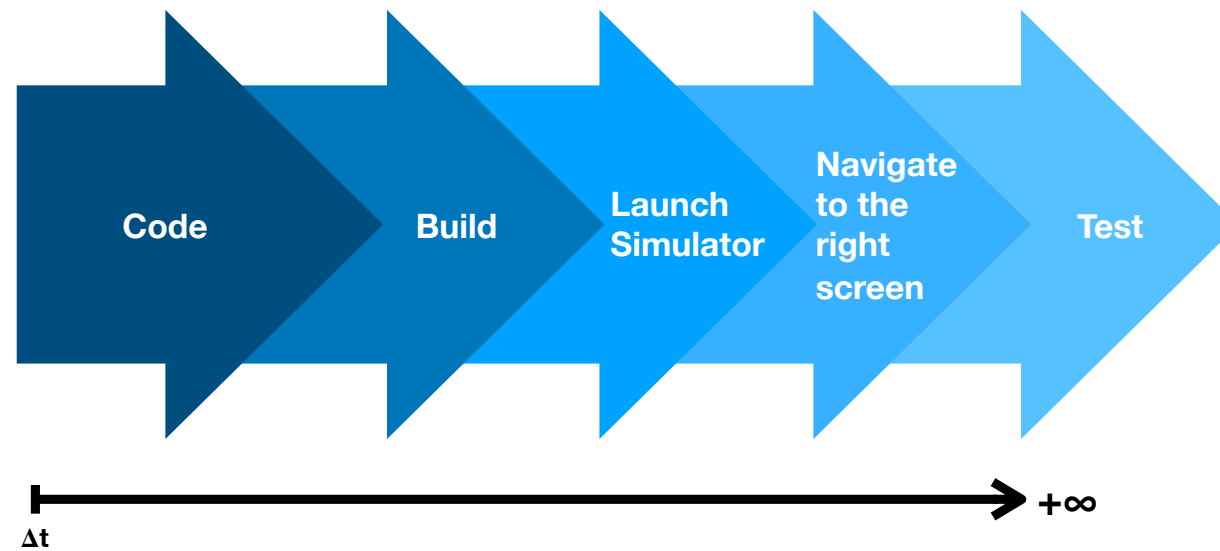
some recommendations of how to use playgrounds





<https://itunes.apple.com/no/app/peil-nyheter-på-en-ny-måte/id1300541549?l=nb&mt=8>

The problem with iOS development



the feedback loop takes too much time!
=> less developer happiness



as a new iOS developer, coming from backend where the feedback loop is smaller
i set myself an objective -> master playground driven development

Why use playgrounds?

Great for experimenting

- spikes, throw away code
- create new controllers/views
- test new pods

<https://github.com/johnsundell/playground>

```
$ playground -t ~/Desktop/lab/lottie-ios-test/lottie-test -d ../lottie-ios/  
Lottie.xcodeproj
```

new pods
new controllers/views
throw away code (spikes)

`view(state) -> rendered screen`

playgrounds are like unit-test for view controllers

Clear boundaries

- For this particular screen:
 - what dependencies do i need?
 - what app context does it need?
 - what objects are called inside?
 - how should i invoke this view controller?



figure out dependencies
side effects
more testable code

Quick testing device + traits



[Kickstarter Playground Helpers](#)

<https://github.com/kickstarter/ios-oss/blob/master/Kickstarter-iOS.playground/Sources/playgroundController.swift>

Why **not** to use playgrounds?

Device specific capabilities



Debugger ability

```
48
49 extension SomeClass {
50
51     static func determineActivity(array1: [Double], array2: [Double], hz: Double) -> Bool {
52
53         let distance = hz * 8.7344
54         let peaks1 = findpeaks(array1, minDistance: distance, minHeight: 2000)
55         let peaks2 = findpeaks(array2, minDistance: distance, minHeight: 800)
56
57         return peaks1.count >= 2 && peaks2.count >= 2
58     }
59 }
60
```

Thread 2: breakpoint 21

Thread 2 | 0 static SomeClass.determineActivity(array1:array2:hz:)

array1 = 300 values
array2 = 300 values
hz = 100
self = 0x0000000000000000
distance = 23.140000000000001
peaks1 = 0 values
peaks2 = 0 values

(lldb) p distance
error: couldn't apply expression side effects: couldn't get the data for variable self
(lldb) po distance
error: couldn't apply expression side effects: couldn't get the data for variable self
(lldb)

None | Filter
All Output | Filter

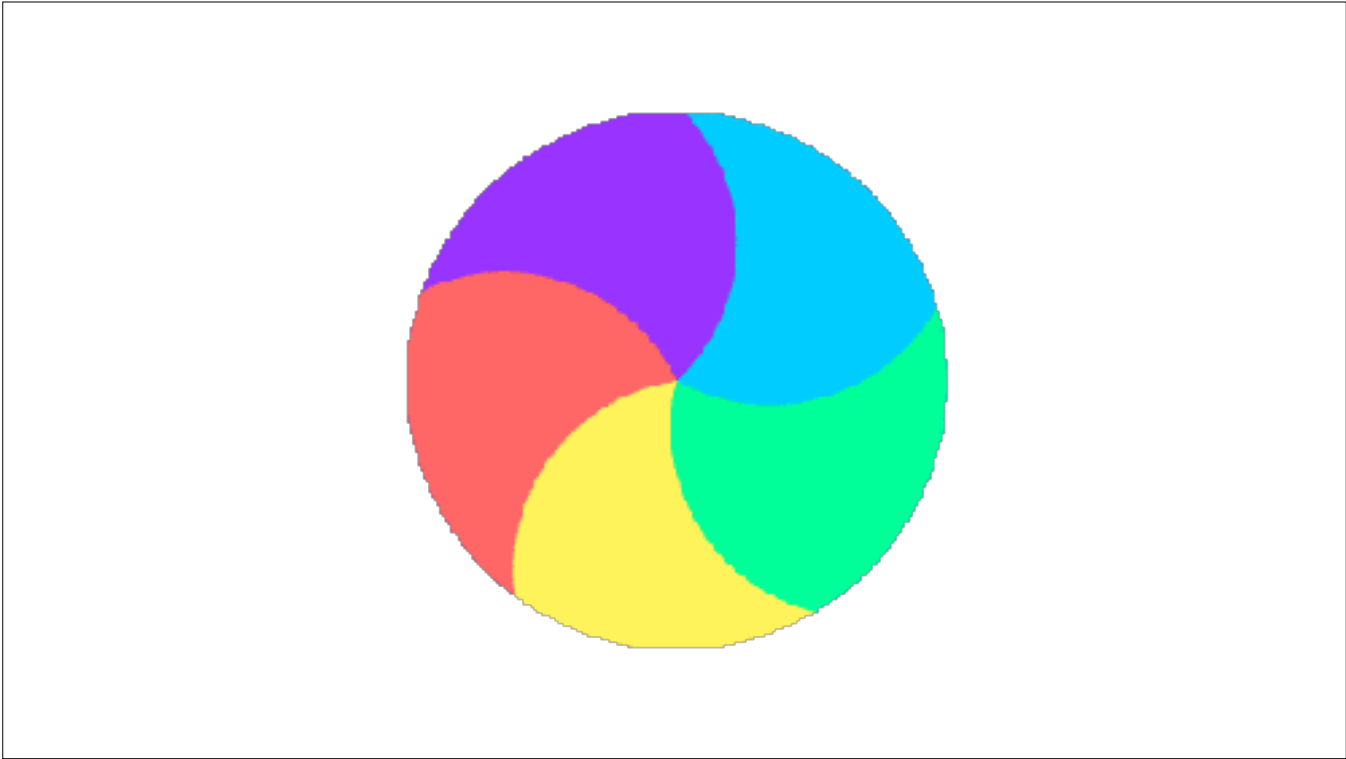
Other reasons






- complex gestures to be tested
- presence of strong side effects
- the effort of adopting them
- etc

managing the
bundle

Network
Requests

like POST or
DELETE



-  Playground-iOS.playground
-  MonkeyBusiness.xcworkspace
-  learning-video-on-ios.playground
-  VGNext.xcworkspace
-  peil-experiments.playground

Application Not Responding

Options



Show All Windows

Hide

Force Quit

BUT

Speed of development/test



Ability to prototype + compose

Work with views and state in **isolation**

How to use them in **your** project

Creating a framework



1. create a new framework target
2. add swift files to the new target*
3. add pods to the new target
4. create playground
5. add playground to the workspace
6. add helper files to playground
7. build the framework
8. profit

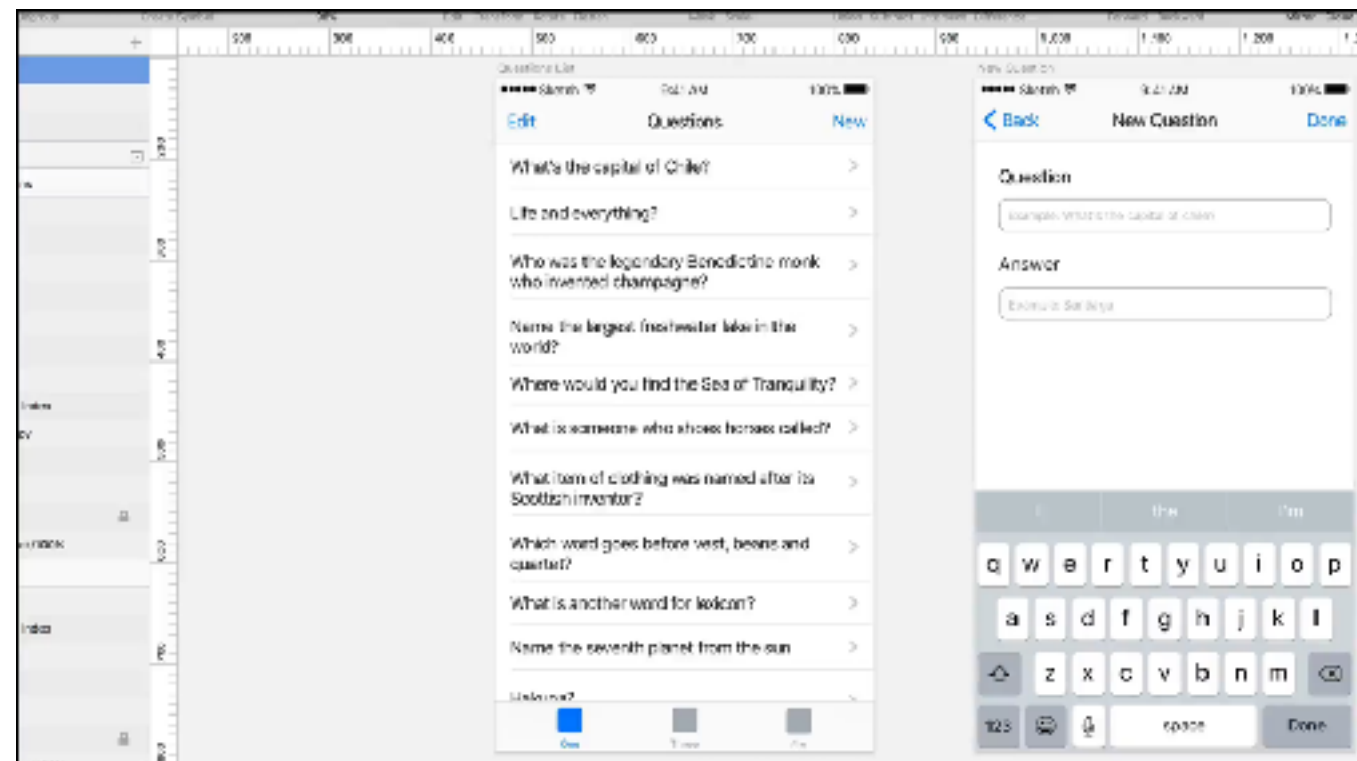
<https://medium.com/flawless-app-stories/playground-driven-development-in-swift-cf167489fe7b>

*This can be done in two ways:

1. every single swift file will be in both **the original target** and **the new framework target**, this is the easier approach for existing code, but it's more difficult to maintain
2. move all the swift files in your project to the **framework target** and work on making the appropriate files **public** to then import them in the **AppDelegate of the original target**. This approach is cleaner and makes explicit the separation of what is public facing code and internal implementation.

Demo

1. show the design of the app
2. show work in progress in the simulator
3. show work in progress in playgrounds
4. style better
5. show issue with border for the long text labels
 - 5.1. how to fix it
6. render app in different device/traits combinations
 - 6.1. fix dynamic font
6. copy code to project, show it in the simulator



1920x1080

Recommendations

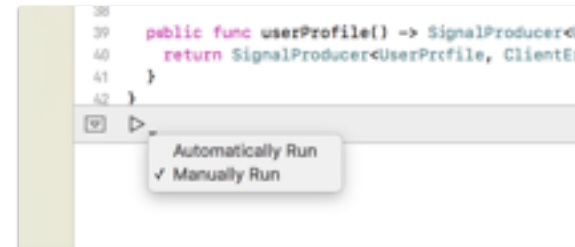
- if you declared a view, added it to a superview, added constraints to it and still doesn't layout appropriately, make sure you set

```
view.translatesAutoresizingMaskIntoConstraints = false
```

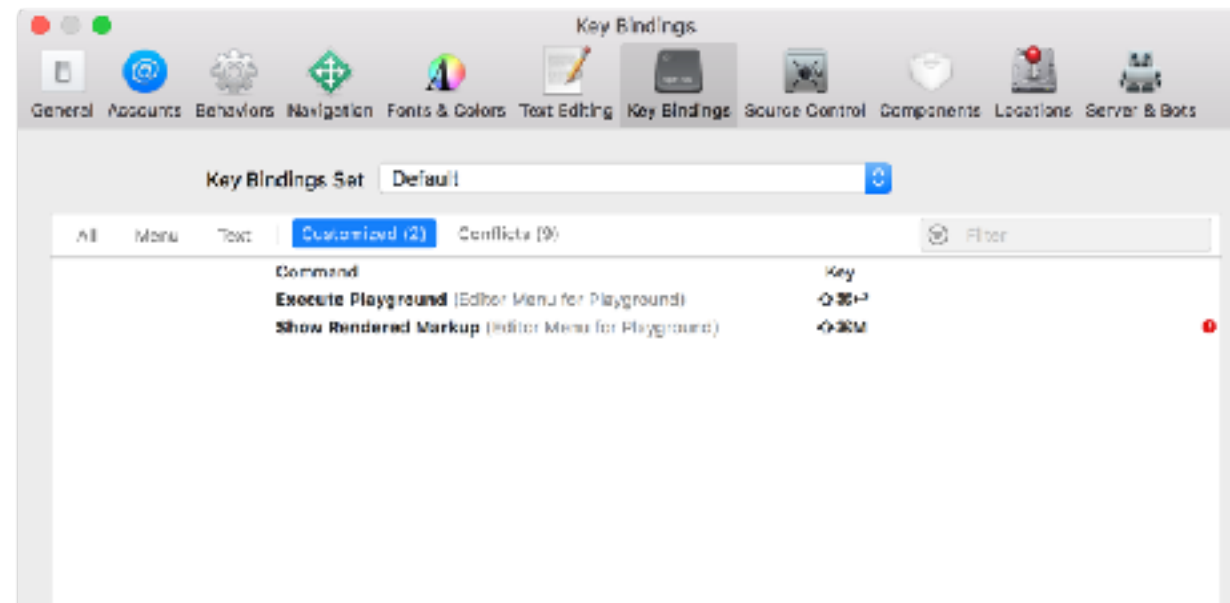
- cautious use of `@testable`

Manually Run Playgrounds

- prevents crashes, makes playground more stable
- easy to add a keyboard shortcut



Useful Key Bindings



Using images

```
UIImage(named: "example")
```

Before

After

```
let myBundle = Bundle(for: MyViewController.self)
UIImage(named: "example", in: myBundle, compatibleWith: self.traitCollection))
```

<https://github.schibsted.io/spt-nextgen-vgnext/ios/blob/master/VGNext/App/ImageScope.swift#L22>

Localized text

```
NSString("example", comment: "An example string")
```

Before

After

```
let bundle = Bundle(for: PeilApplication.self)
NSString(
    "example", tableName: nil, bundle: bundle, value: "\(self) localization not found", comment: ""
)
```

<https://github.schibsted.io/spt-nextgen-vgnext/ios/blob/master/VGNext/Utils/String+Localized.swift#L17>

Using Fonts

To be able to use custom fonts in playgrounds, you need to register them, even if you already did on the application

```
@discardableResult
public func registerFont(withName fontName: String) -> Bool {
    let bundle = Bundle(for: PeilApplication.self)
    let fontURL = bundle.url(forResource: fontName, withExtension: "otf")
    return CTFontManagerRegisterFontsForURL(fontURL! as CFURL, .process, nil)
}

public func registerPeilFonts() {
    registerFont(withName: "GT-Walsheim-Condensed-Bold")
    registerFont(withName: "GT-Walsheim-Condensed-Regular")
}
```

<https://github.schibsted.io/spt-nextgen-vgnext/ios/blob/master/Playground-iOS.playground/Sources/PeilConfiguration.swift#L55-L60>

Conclusions

Conclusions

- ✓ it requires a bit of work but forces you to write less effect dependent code
- ✓ increases velocity of iteration and **can be used within your project**
- ✓ easier test of different traits combinations => better support for them
- ⚠ there is a way to go regarding Xcode playground's stability

if you make state more explicit and easy to set, it's easier to quickly develop considering the most cases



Main request for wwdc: more stable playgrounds!



end

Appendix 1: Resources

Resources

- [Playground Driven Development Talk by @mbrandronw](#)
- [Everyone is an API designer by @johnsundell](#)
- [Adding playgrounds to your Xcode Project by @onmyway133](#)
- [Playground Driven Development in Swift by @onmyway133](#)
- [How to playground by @fespinoza](#)
- [Kickstarter's playground helpers](#)

Resources (even more)

- [Point free's testing helpers](#)
- [@johnsundell's playground script](#)
- [Use a custom framework in a playground](#)
- [Peil's source code](#)
- [Writing unit tests in Swift playgrounds](#) by @johnsundell
- [Test-Driven Reactive Programming](#) from [objc.io](#)

Appendix 2: playground-error dictionary

error: module compiled with Swift 4.0 cannot be imported in Swift 4.0.3

Problem

Solution

Clear derived data

Playground execution failed:

error: Couldn't lookup symbols:

```
__T013PeilFramework13VGServiceViewCACSC6CGRectV5frame_tcfC  
__T013PeilFramework13VGServiceViewCMA
```

Problem

Solution

Playgrounds regression between CocoaPods 1.4.0 and CocoaPods 1.5.0

Playground execution failed:

error: Couldn't lookup symbols:

__T013PeilFramework13VGServiceViewCACSC6CGRectV5frame_tcfC
__T013PeilFramework13VGServiceViewCMA

Problem

Solution

```
use_frameworks!

target 'DateToolsPlayground' do
  pod 'DateTools'
end

post_install do |installer|
  installer.pods_project.targets.each do |target|
    target.build_configurations.each do |config|
      config.build_settings['CONFIGURATION_BUILD_DIR'] = '$PODS_CONFIGURATION_BUILD_DIR'
    end
  end
end
```

<https://github.com/CocoaPods/CocoaPods/issues/5334#issuecomment-223444937>

error: Playground execution aborted: error: Execution was interrupted, reason: EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0).
The process has been left at the point where it was interrupted, use "thread return -x" to return to the state before expression evaluation.

Problem

Solution

There is an unhandled NSError to fix


```
error: no such module 'Playground_iOS_Sources'
```

Problem

Solution

Force a recompilation of the sources by adding a comment or anything



end