

HOW TO NOT GET SAND IN YOUR EYES WHEN USING PLAYGROUNDS

by Felipe Espinoza



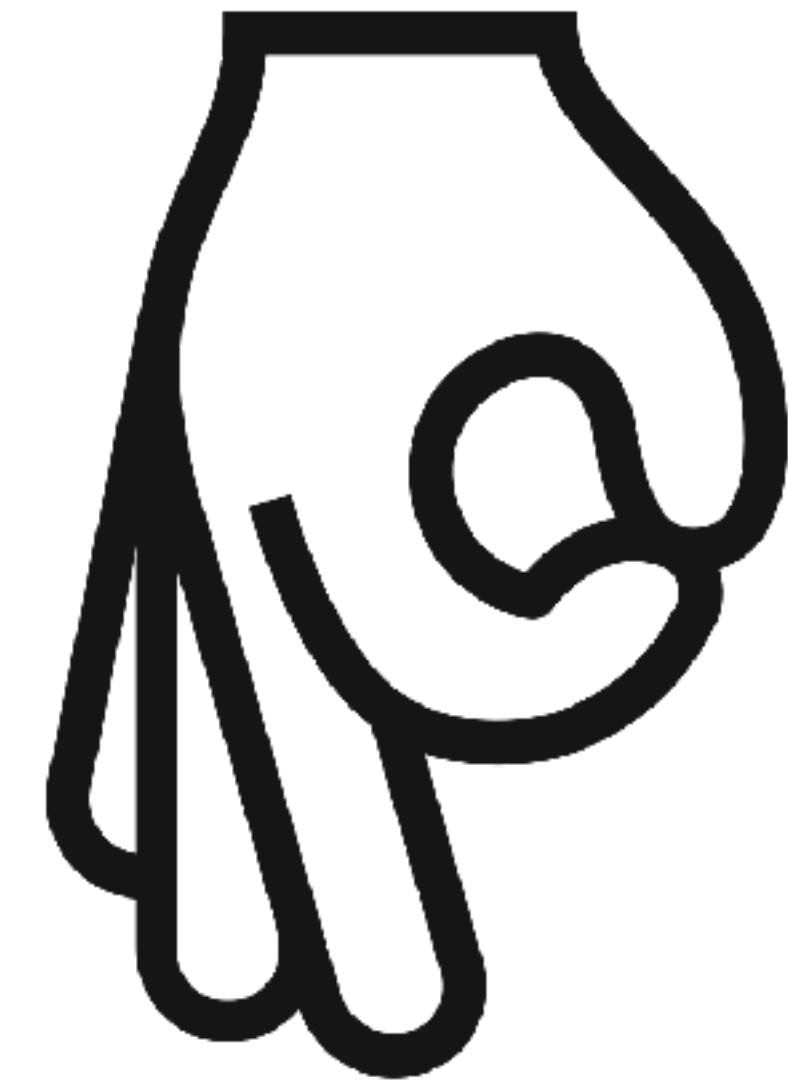
@fespinoza



@fespinozecast



SCHIBSTED
MEDIA GROUP



Peil

fra VG

■ ■ ■

09:41

* 100 %

DAGENS PEIL



OJ

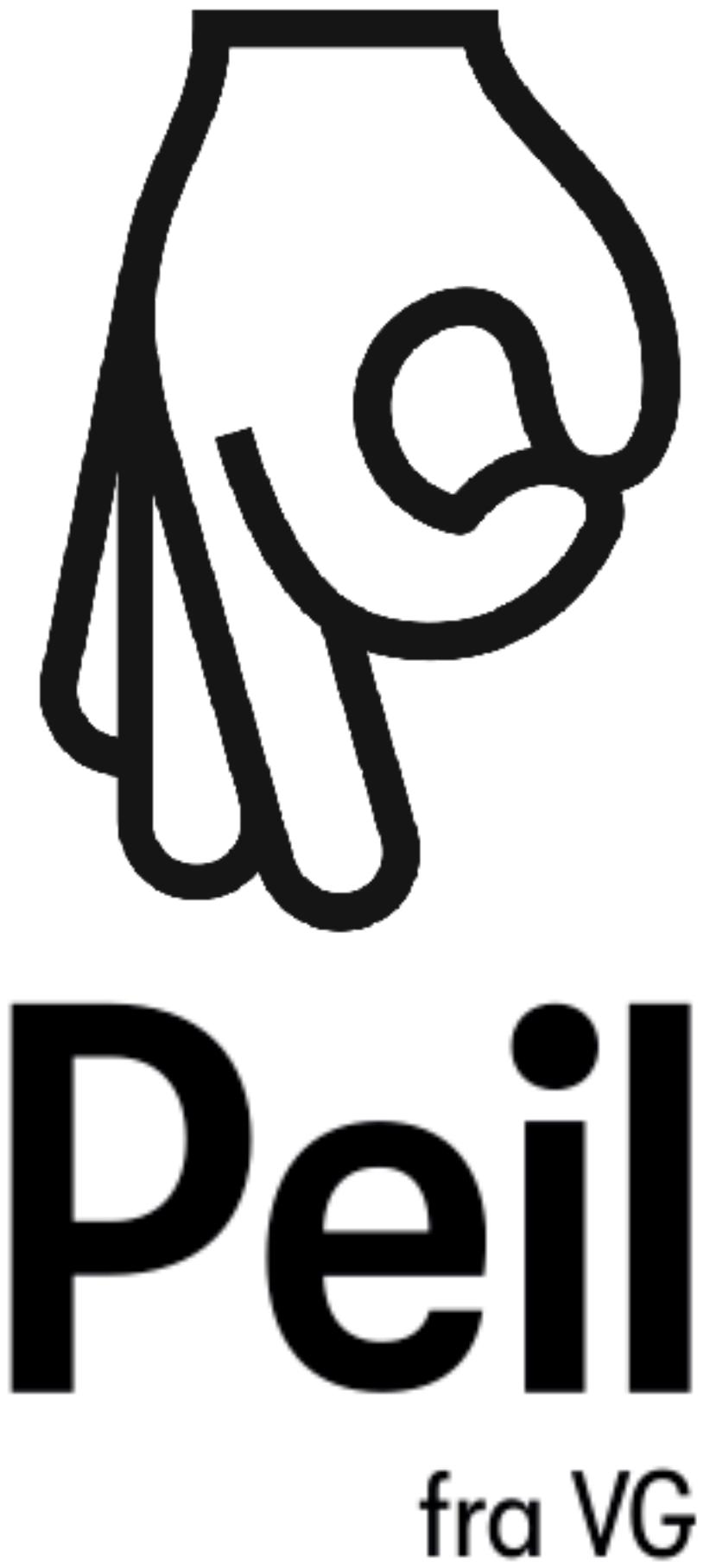
Nyhetsoppsummering

- Ber om livstid for Stockholm-terroristen.
- Partitopper samlet til metoo-møte.
- PST: Tolv norske fremmedkrigere etterlyst.

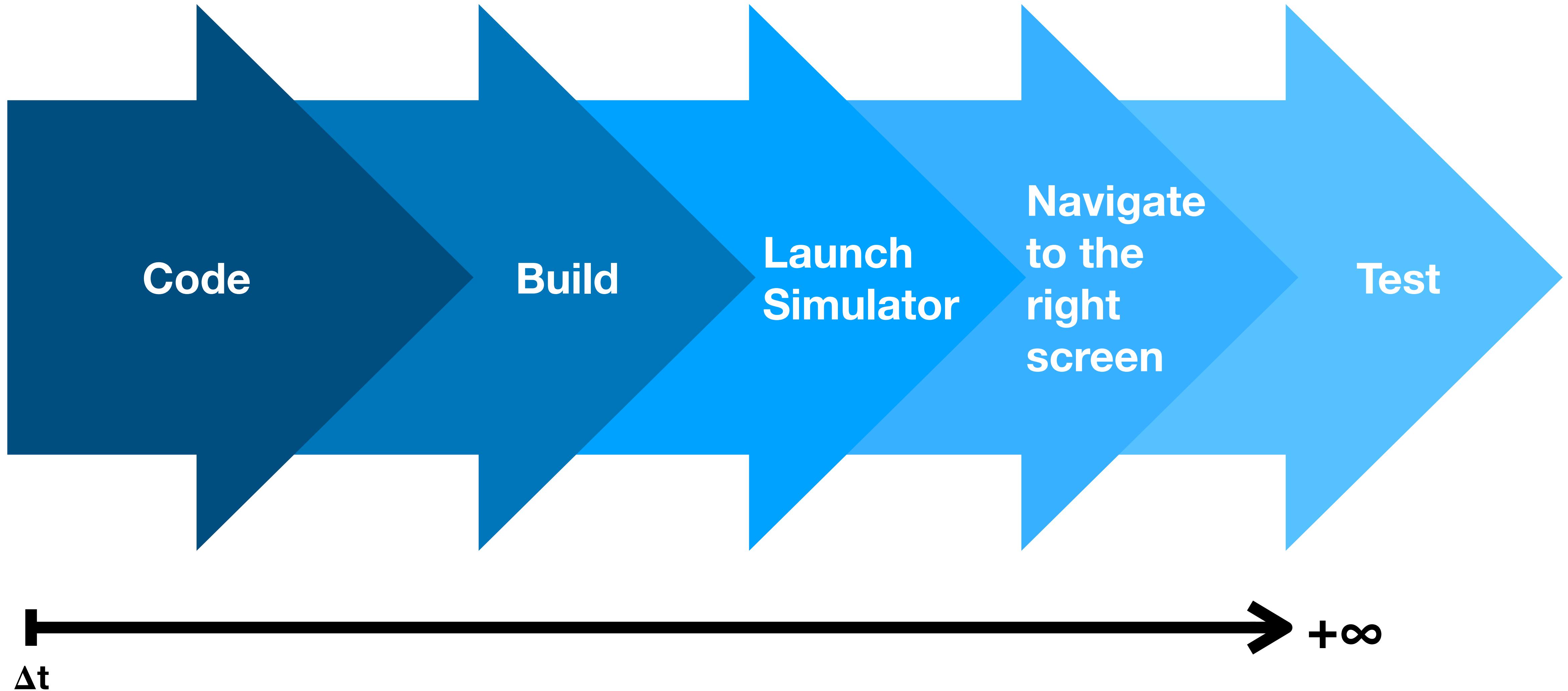
Privat

30. jan. TERROR METOO PST

CAMILLA
18 MINUTTER SIDEN



The problem with iOS 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
```

Clear boundaries

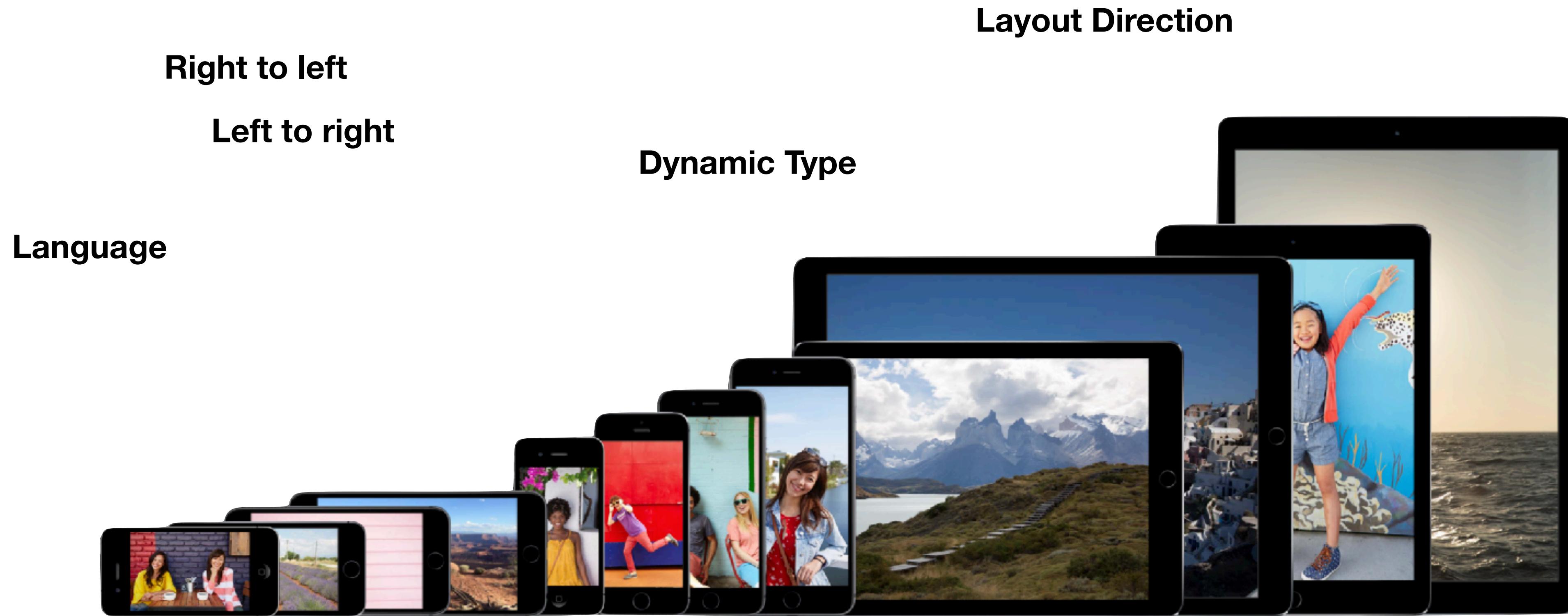
view(state) -> rendered screen

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?



Quick testing device + traits



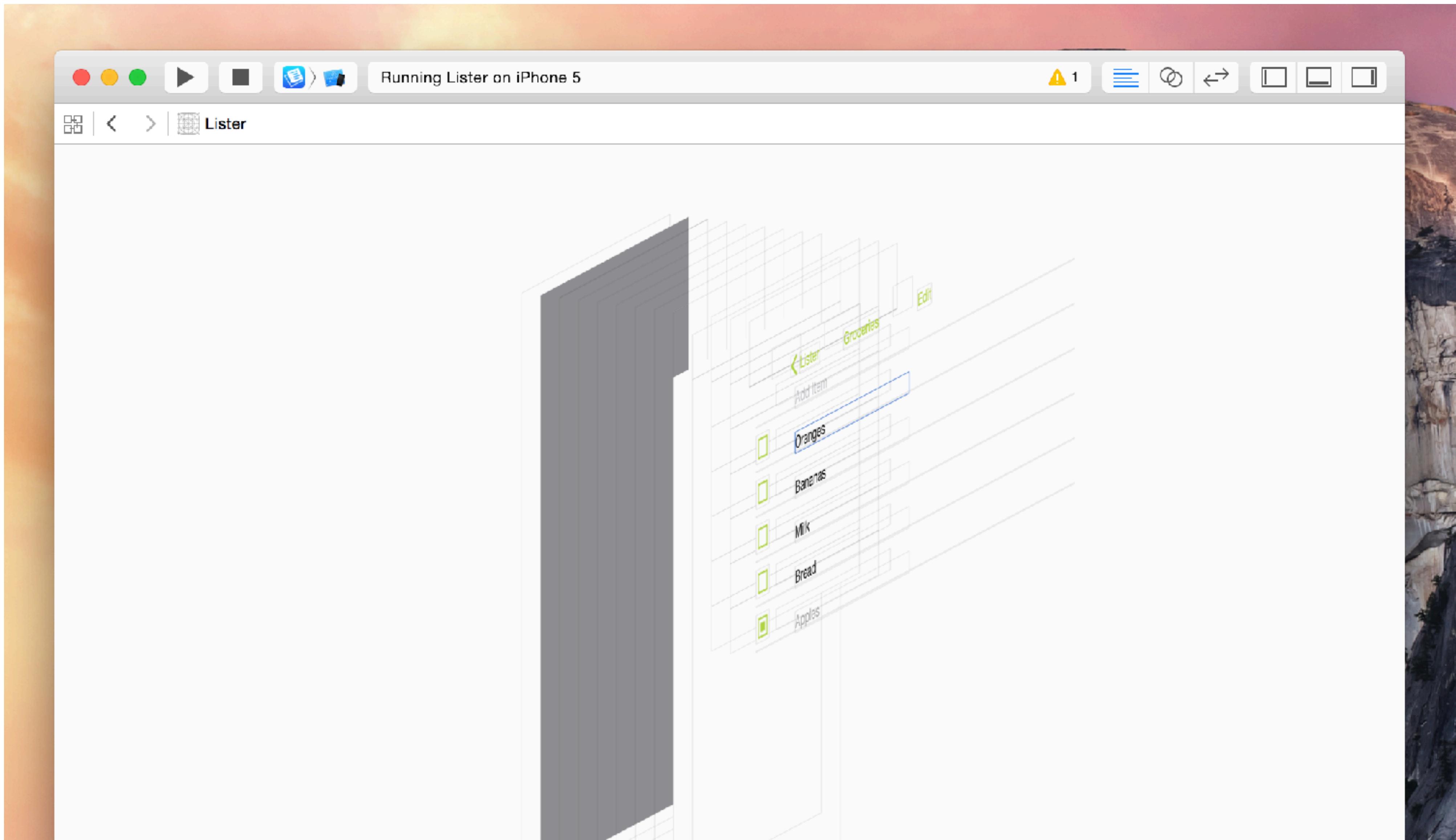
[Kickstarter Playground Helpers](#)

Why **not** to use playgrounds?

Device specific capabilities



Debugger ability



Other reasons

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



- lottie-ios-test.xcworkspace
- Playground.playground
- lottie-test.xcworkspace
- LinkedIdeas.xcworkspace
- MonkeyBusiness.xcodeproj
- 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



How to use them in your
project

Creating a framework



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

Demo

Recommendations

- beware of `translateAutoResizingMasksIntoConstraints`
- requires bundle management, not just `UIImage(named:)`
- cautious use of `@testable`

Conclusion

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





MAY THE ODDS BE EVER IN YOUR FAVOR

Thanks!



@fespinoza



@fespinozecast

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
- How to playground
- Kickstarter's playground helpers
- Point free's testing helpers
- @johnsundell's playground script

Appendix 2: playground-error dictionary

Problem

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

Solution

Clear derived data

Problem

Playground execution failed:

```
error: Couldn't lookup symbols:  
__T013PeilFramework13VGServiceViewCACSC6CGRectV5frame_tcfC  
__T013PeilFramework13VGServiceViewCMa
```

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>

Problem

```
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.

Solution

There is an unhandled NSError to fix

Problem

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

Solution

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