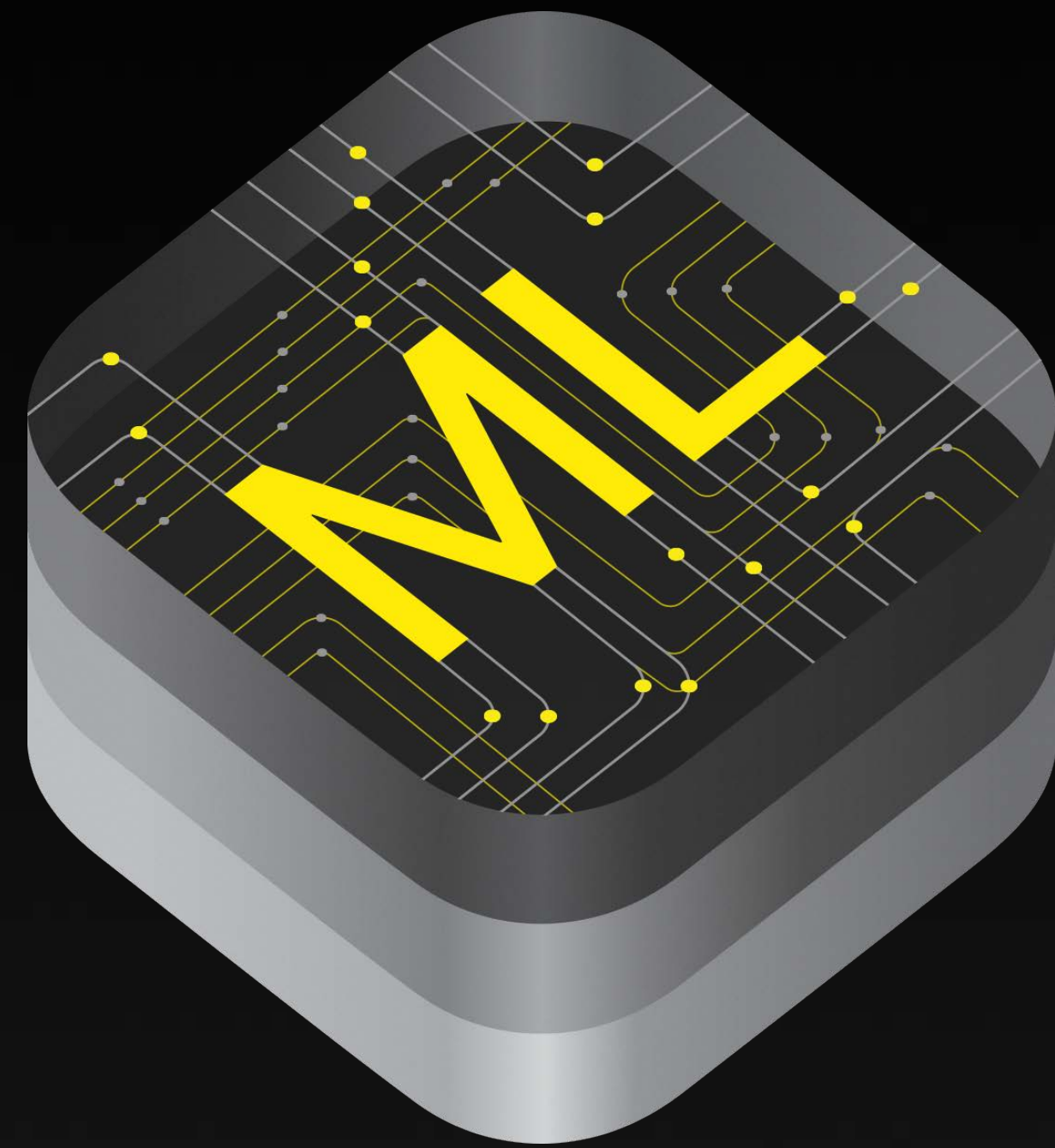


 **WWDC17**

Technology and Refinements





iOS macOS watchOS tvOS



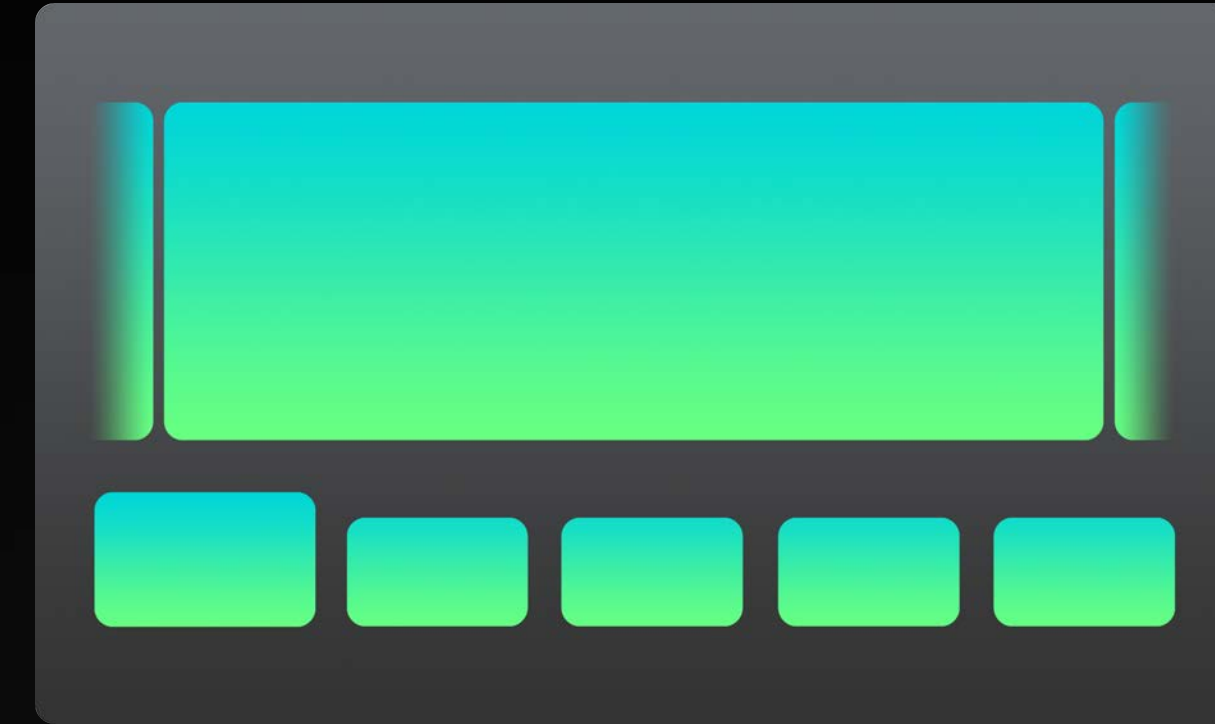
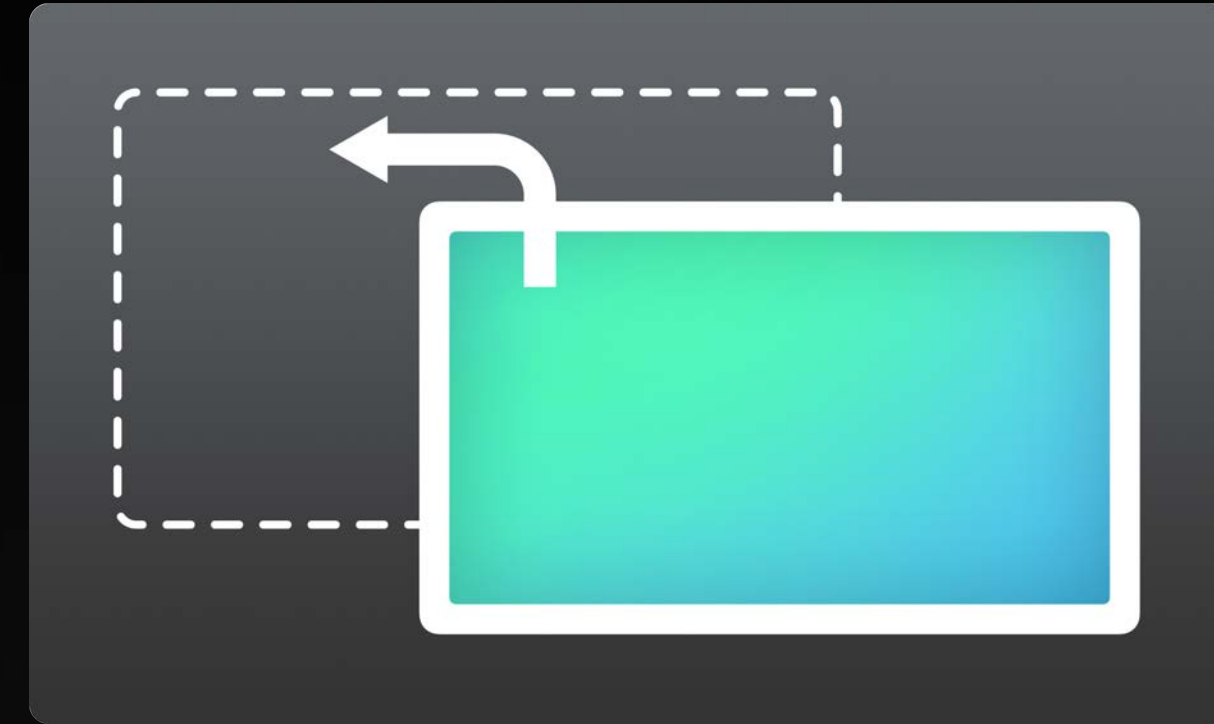
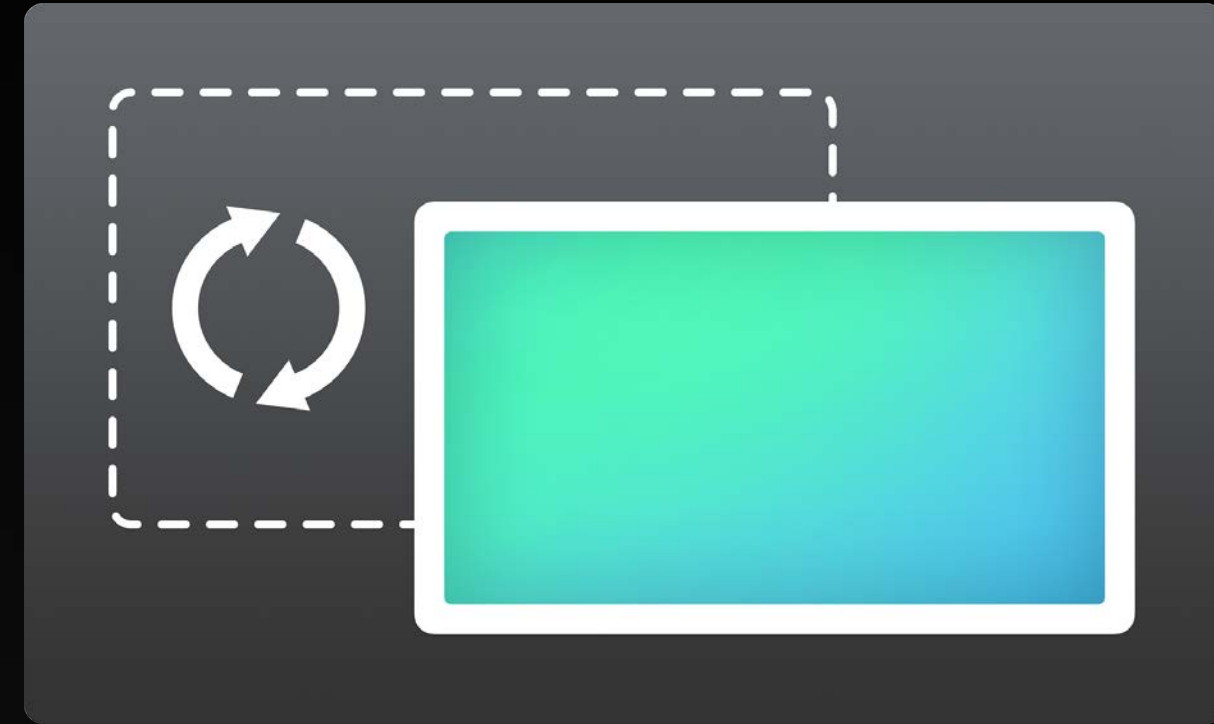
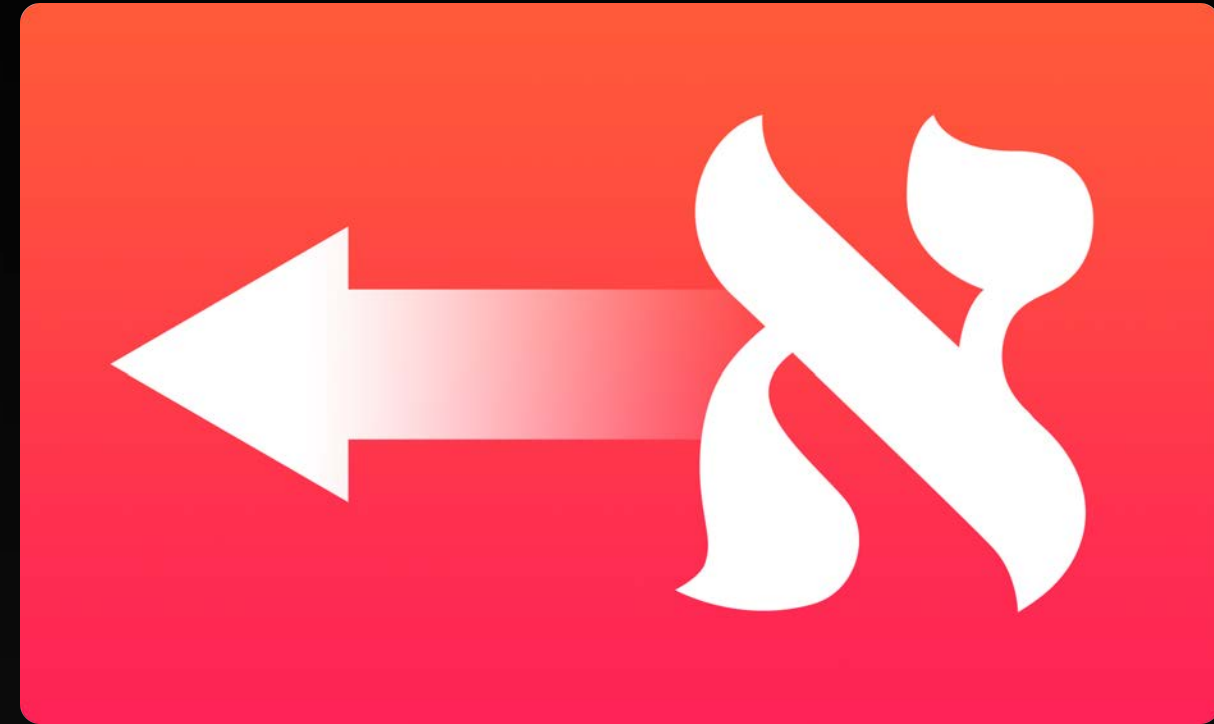
iOS



macOS



watchOS



tvOS

New iOS design resources

App Store cleanup

TestFlight more testers

Localized enrollment

Faster app review

Schedule a call

Responses to reviews

Phased release

Unified Developer Services key

Reviews API

Apple Pay domain autorenewal

TestFlight multiple builds

Xcode automatic distribution signing

Subscriptions reporting

Subscriptions apps transfer

Expanded free trials

Receipt enhancements

64

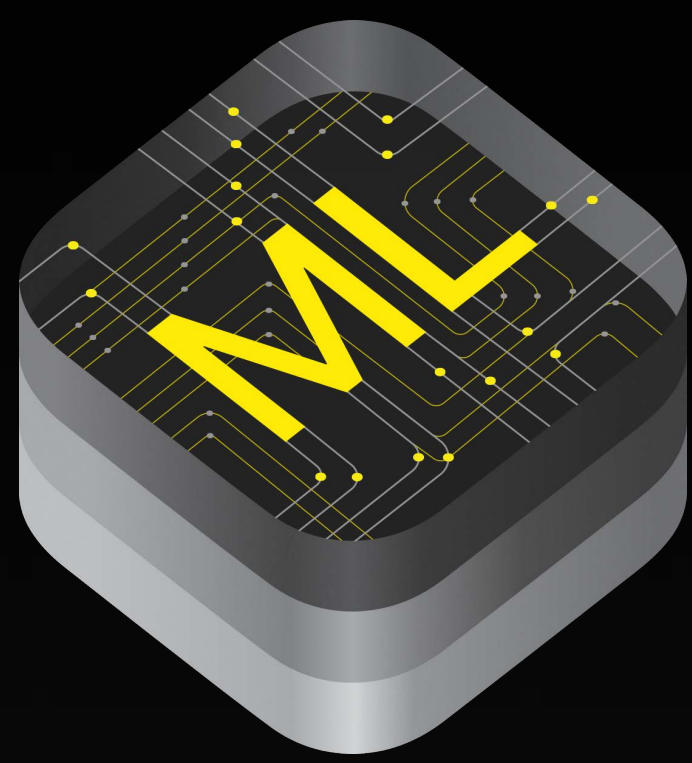
High Sierra will be the last macOS release to support 32-bit apps without compromises



64-bit Requirements

January 2018: New apps

June 2018: All apps and updates





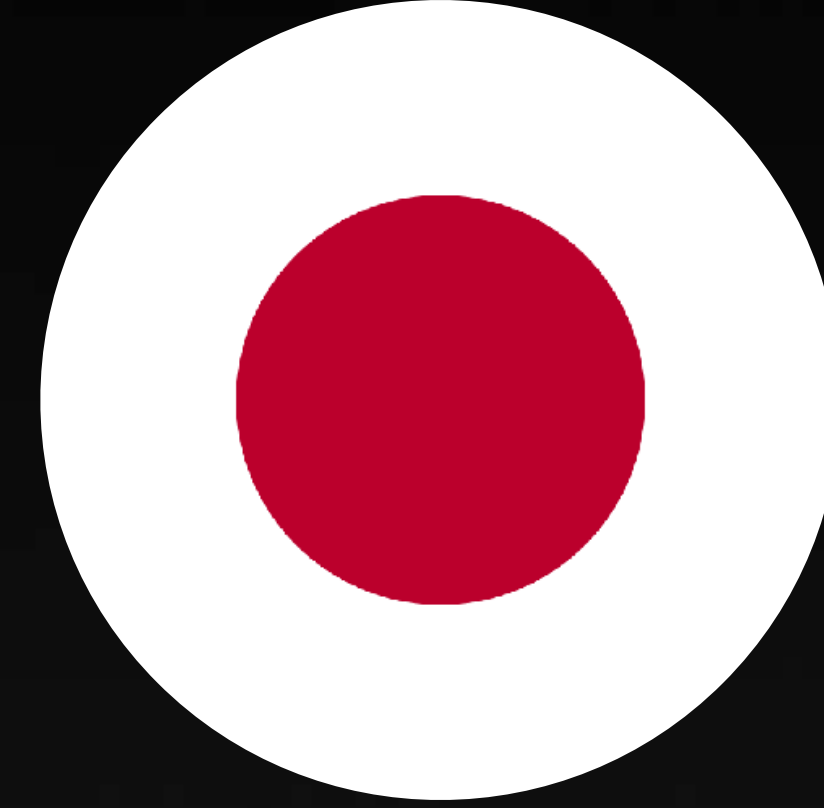
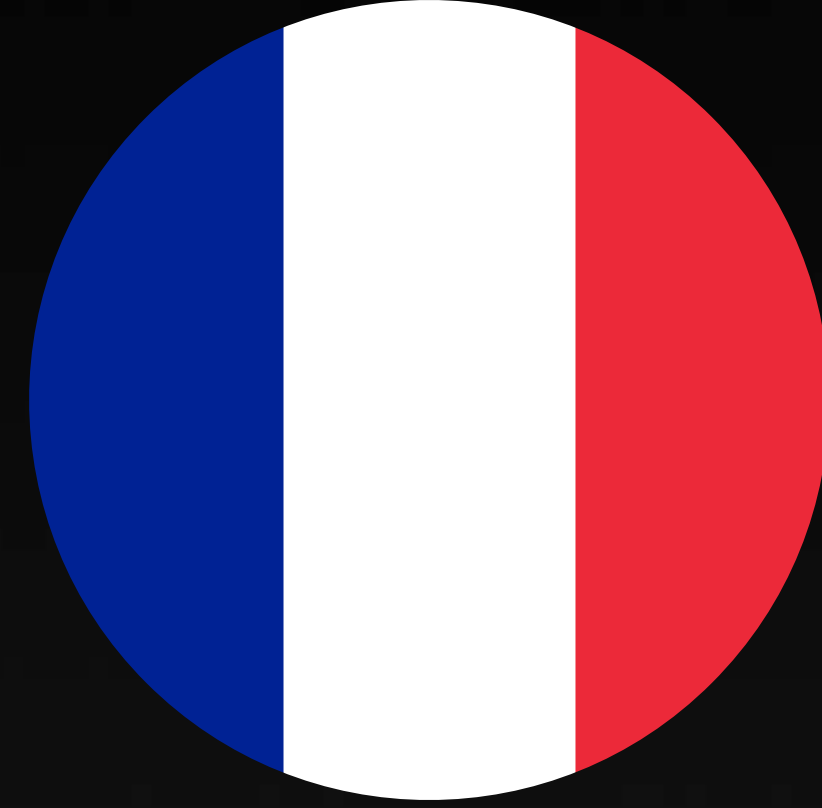


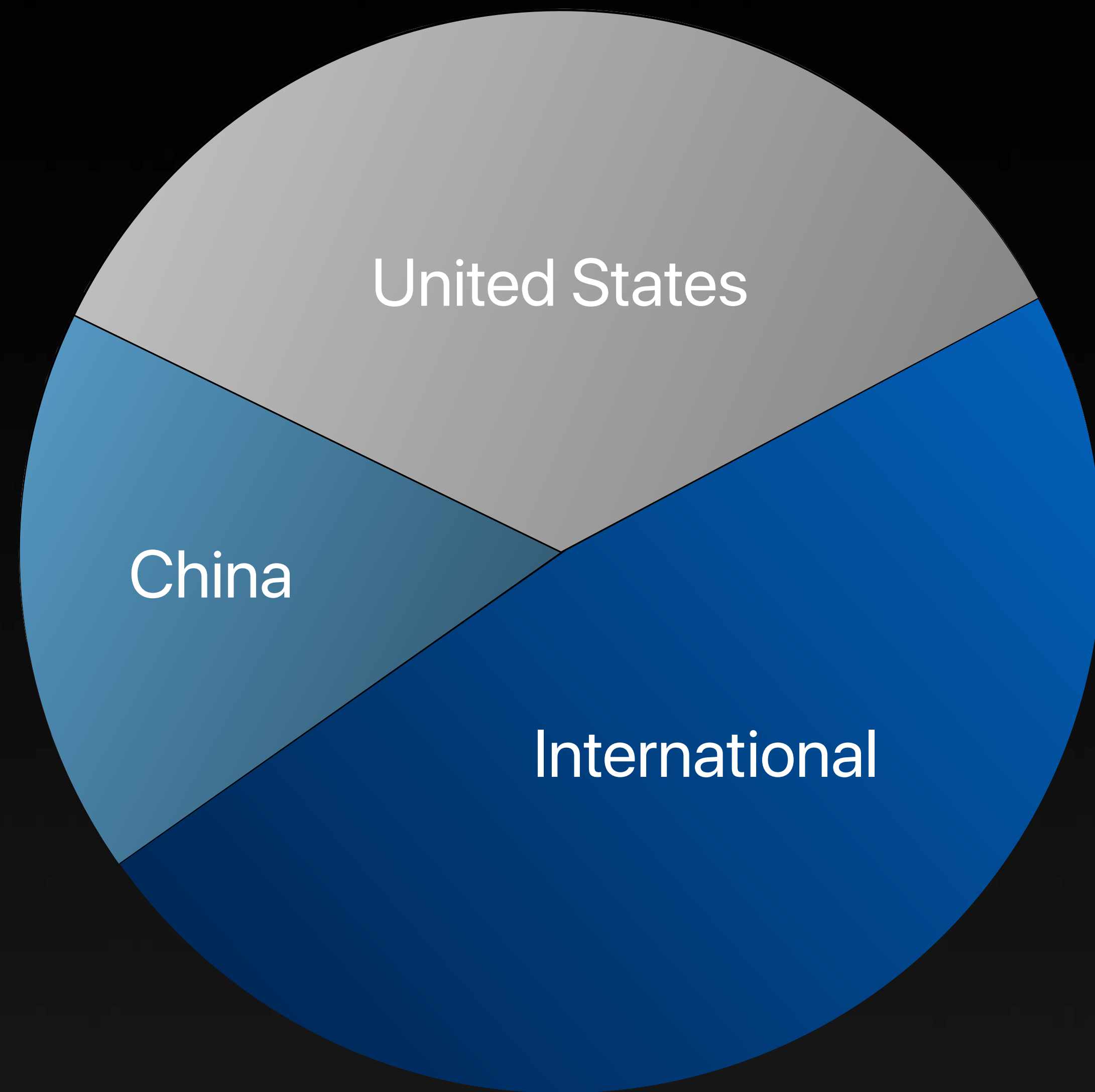
Swift Playgrounds



1 Million

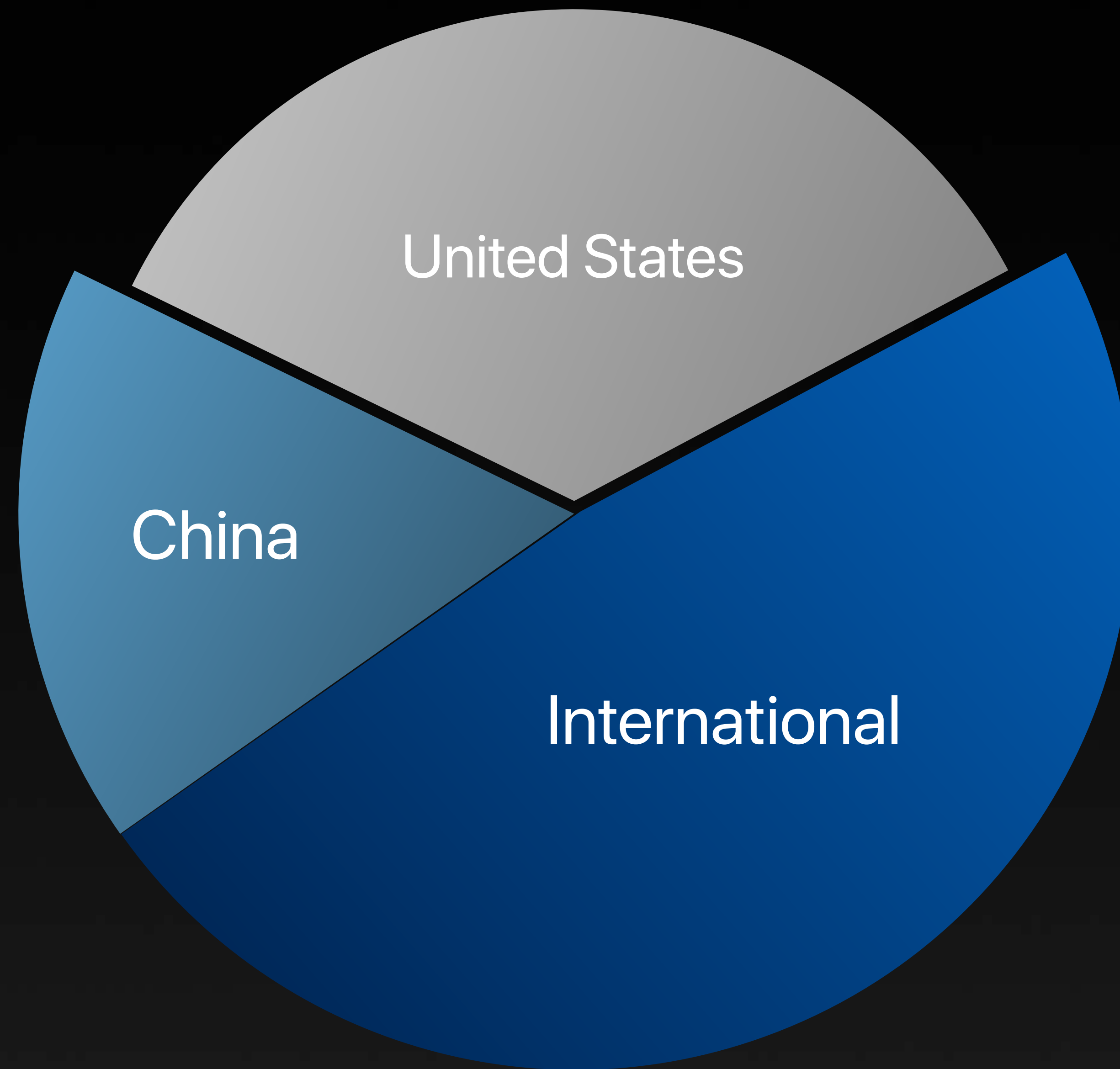
Users of Swift Playgrounds





65%

Users Outside U.S.



65%

Users Outside U.S.

Looping All the Sides


0/4

Goal: Use a for loop to repeat a sequence of commands.

In this puzzle, you must collect four gems that are located in the same relative locations around a square. You'll create a **loop** that repeats the code below for each of the sides to solve the entire puzzle.

- 1 Drag a `for` loop from the code library, then drop it above the existing code.
- 2 Tap the bottom curly brace to select the loop.
- 3 Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

```
for i in 1 ... 4 {  
  moveForward()  
  collectGem()  
  moveForward()  
  moveForward()  
  moveForward()  
  turnRight()  
}
```



for collectGem() moveForward() turnRight()

- Line-by-line highlighting
- Content update notifications
- Add new Playground pages
- Improved error messages
- MapKit support

Looping All the Sides


0/4

Goal: Use a for loop to repeat a sequence of commands.

In this puzzle, you must collect four gems that are located in the same relative locations around a square. You'll create a **loop** that repeats the code below for each of the sides to solve the entire puzzle.

- 1 Drag a `for` loop from the code library, then drop it above the existing code.
- 2 Tap the bottom curly brace to select the loop.
- 3 Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

```
for i in 1 ... 4 {  
  moveForward()  
  collectGem()  
  moveForward()  
  moveForward()  
  moveForward()  
  turnRight()  
}
```

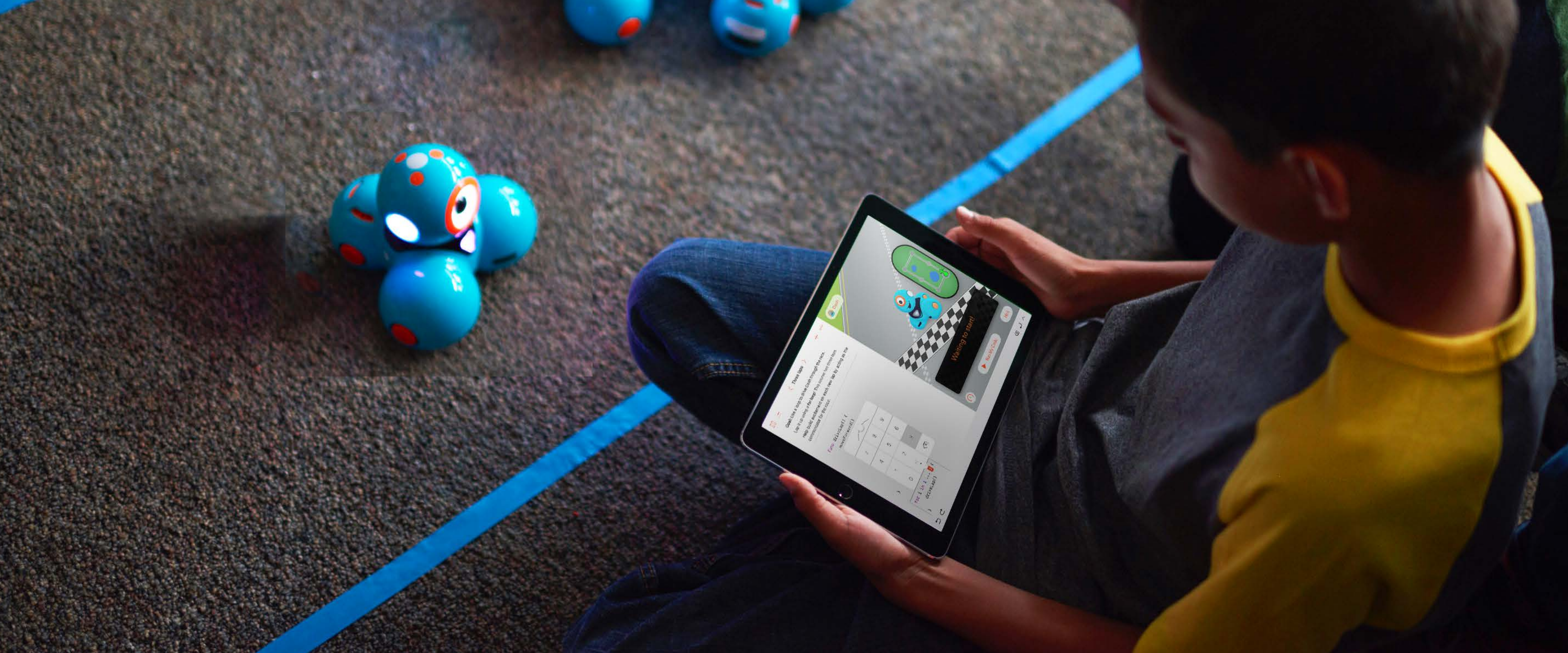


for collectGem() moveForward() turnRight()

- Line-by-line highlighting
- Content update notifications
- Add new Playground pages
- Improved error messages
- MapKit support



Swift Playgrounds 1.5



Three legs

Speak like a frog to make Chalk through the race.
Use if to wait in the loop. This occurs the first time the loop runs and then on each new lap by adding to the counter of 100 ms.

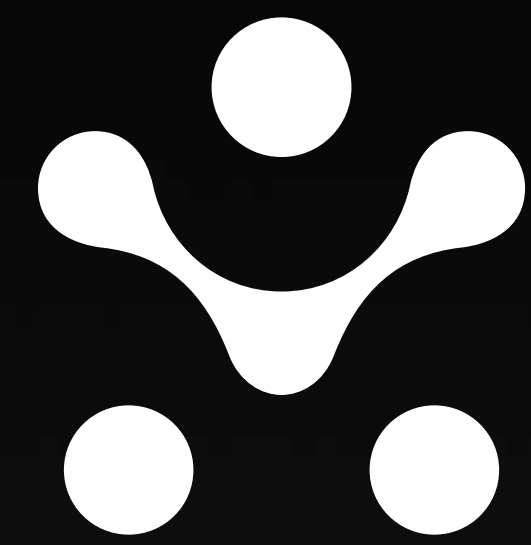
```
func digitalWrite( LED_PIN, HIGH );
```

Waiting to start!



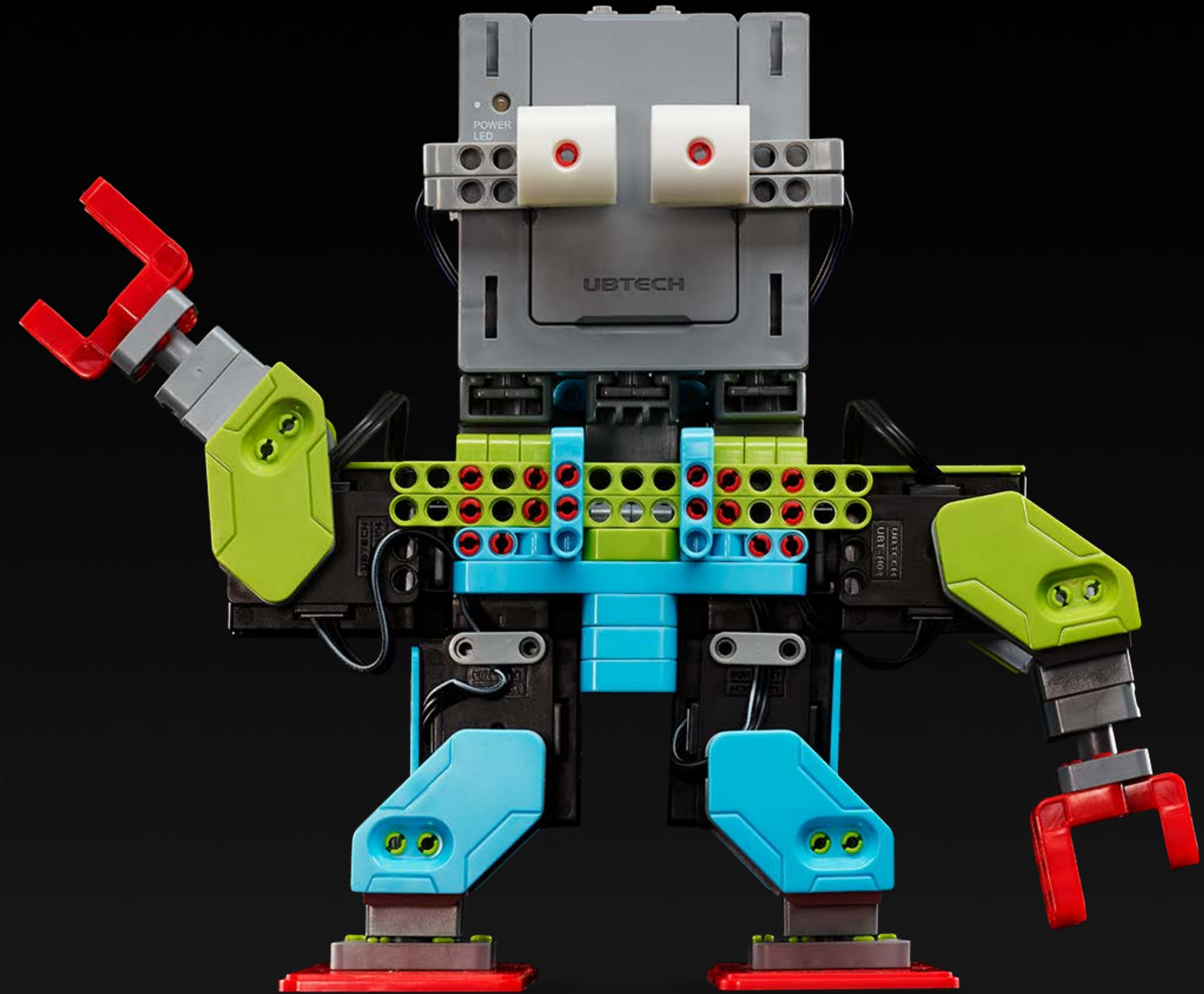
Parrot





UBTECH

Dream With Robots



 sphero

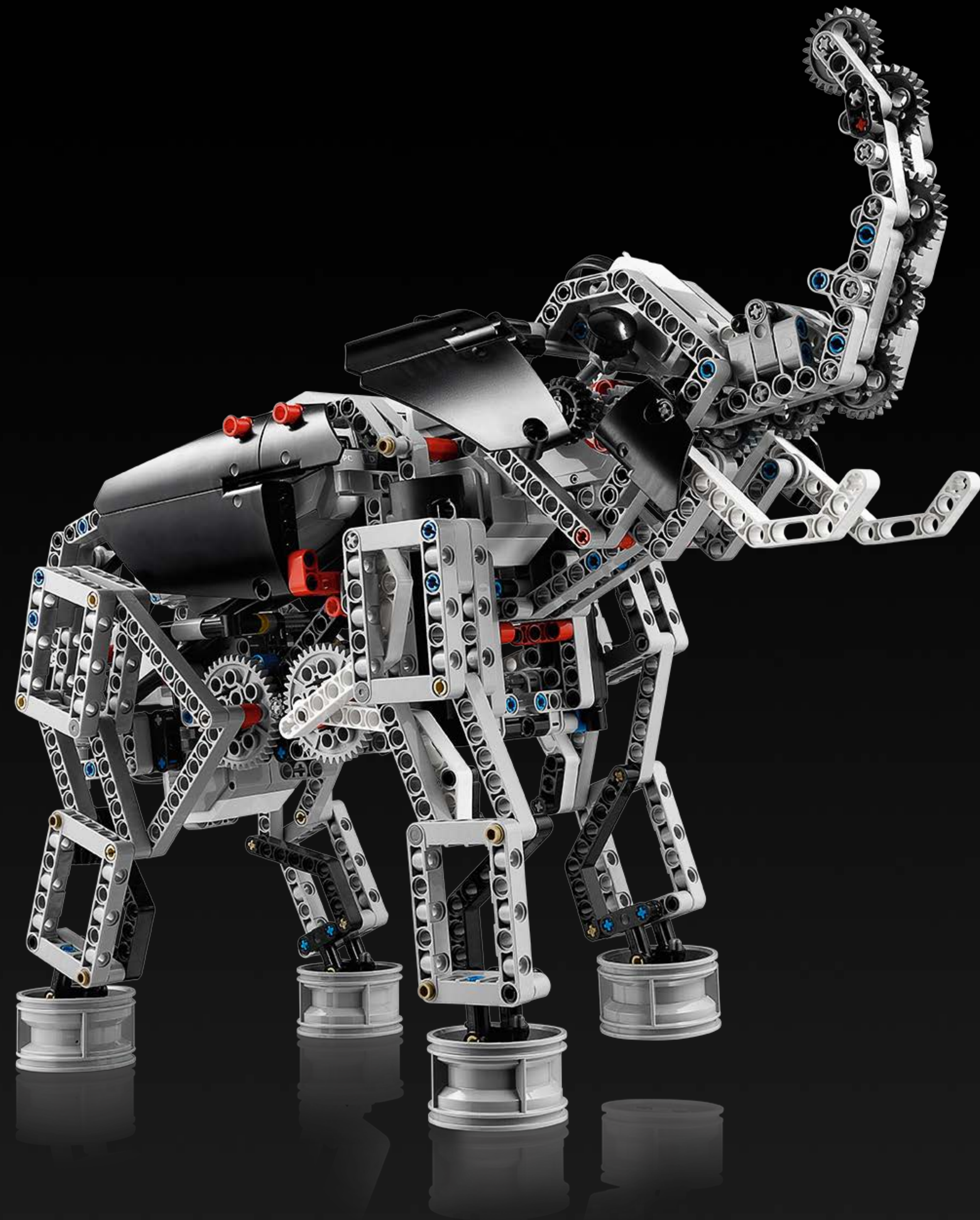


w.
wonder
workshop



Oskoog
music





Program

Using the turtle prototype you have just built, create a function named `moveTurtle` that:

- 1 Moves your turtle forward for a distance of 40 cm and stops at the "fruit" without using a sensor.

Example

To control motors, remember the `motorOn` function that was covered in the previous chapters. To control two motors simultaneously, you can explore the `move` function. For example:

```
move(forRotations: 1.0, leftPort: .b,
rightPort: .c, leftPower: 100.0,
rightPower: 100.0)
```

Analyze

Use the information found in Port View and Data View to determine the speed of your turtle.

```
func TurtleMove() {
  ev3.move(forSeconds: 23, leftPort: .b,
rightPort: .c, leftPower: 50,
rightPower: 50)
}
TurtleMove()
```

The screenshot shows the LEGO Mindstorms software interface with three data views stacked vertically. At the top, there are navigation icons and the text 'LEGO®'. The top view is red and labeled 'LARGE MOTOR' with a power level of 100.00 and a rotation count of 32.64. The middle view is blue and labeled 'ULTRASONIC' with a distance of 26.7 cm. The bottom view is yellow and labeled 'TOUCH' with a status of 1 (Touched). At the bottom of the interface, there are buttons for 'Run My Code' and 'Hint'.

Cancel

Accessories



EV3 Animal Res...
Swift 3.1 Edition

GET



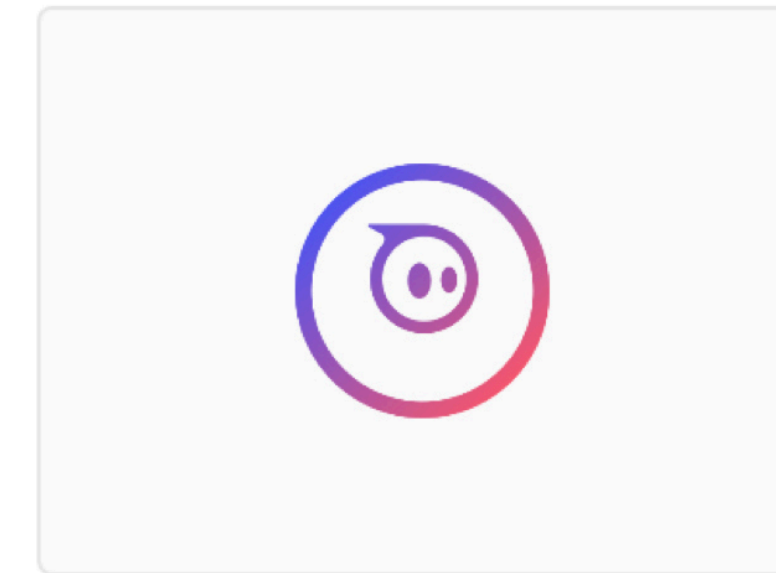
EV3 Template
Swift 3.1 Edition

GET



Sphero Arcade
Swift 3.1 Edition

GET



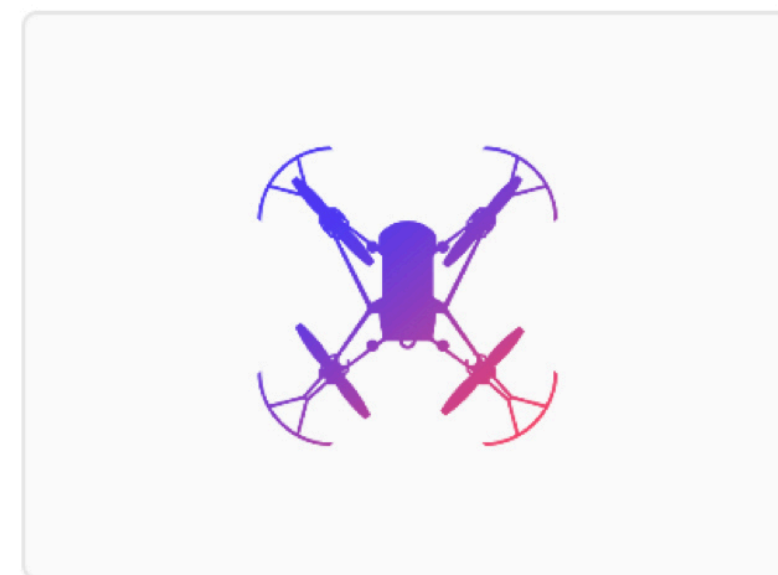
Sphero Template
Swift 3.1 Edition

GET



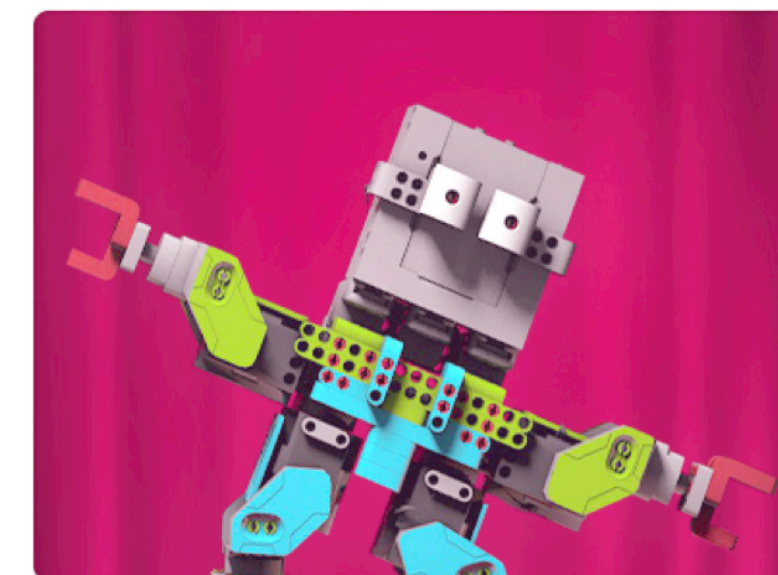
Parrot Education
Swift 3.1 Edition

GET



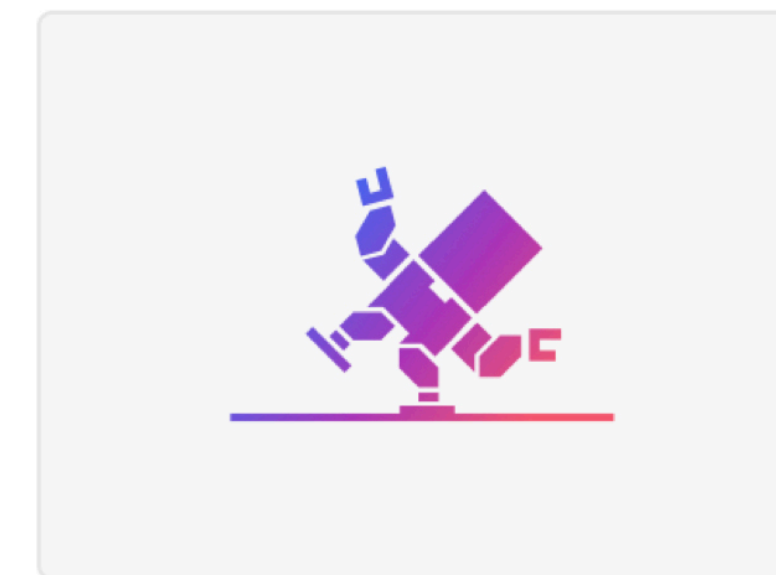
Parrot Template
Swift 3.1 Edition

GET



MeeBot Dances
Swift 3.1 Edition

GET



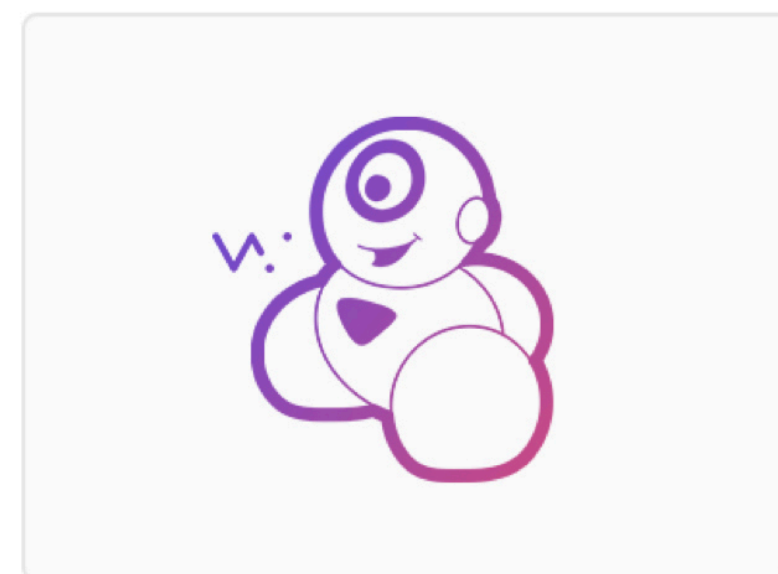
MeeBot Template
Swift 3.1 Edition

GET



Dash

GET



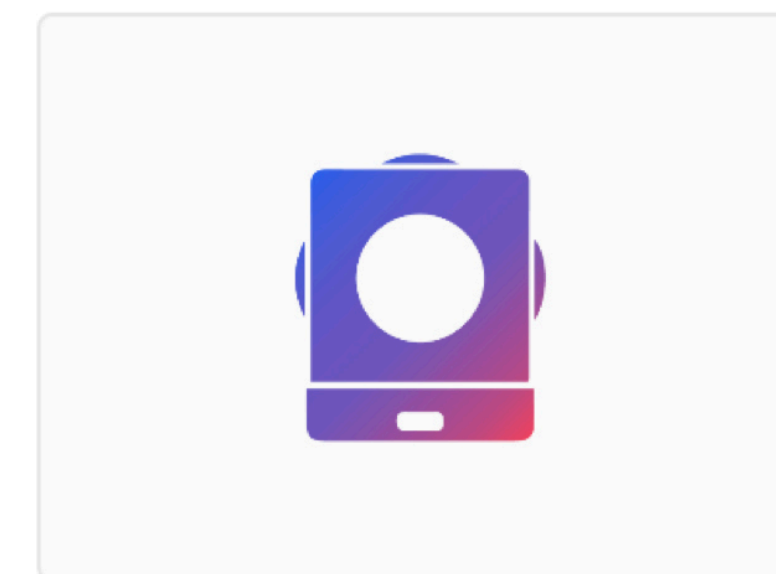
Dash Template

GET



Skoog

GET



Skoog Template

GET



Learn To Code



Challenges



Accessories



Starting Points

The best way to control robots and
drones with code you write yourself



Available

Today

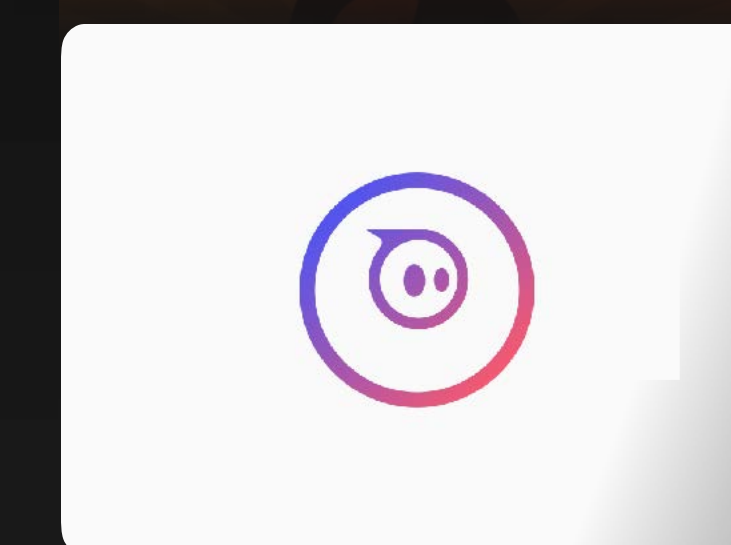
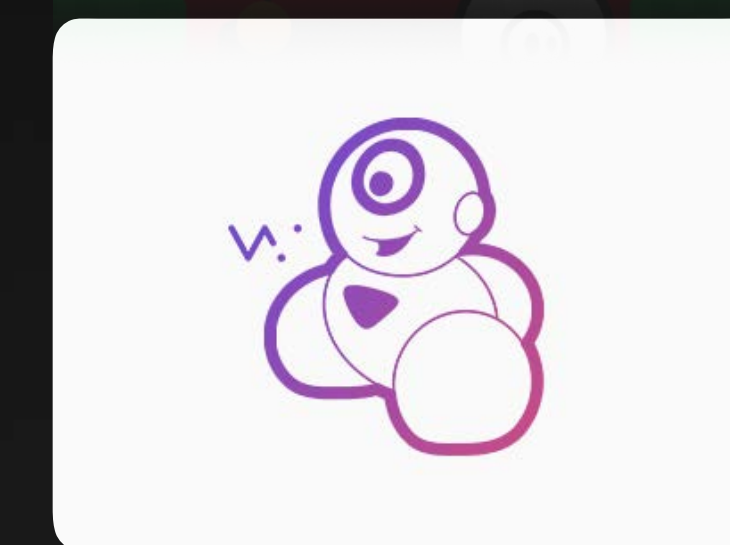
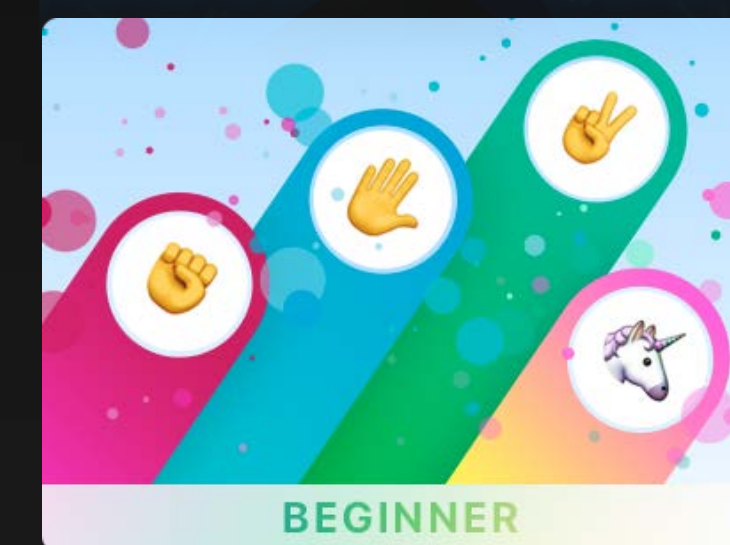
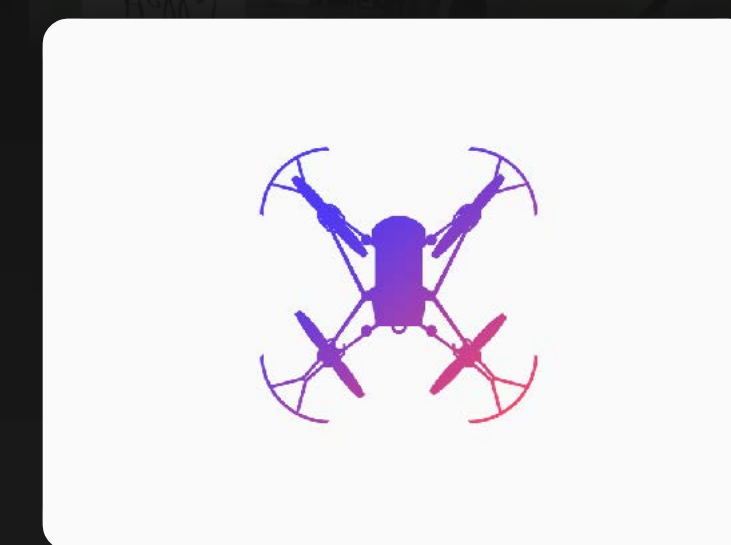
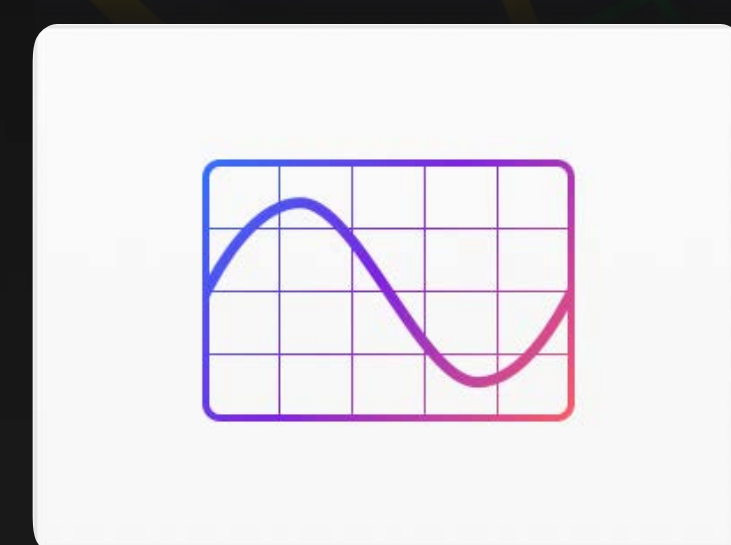
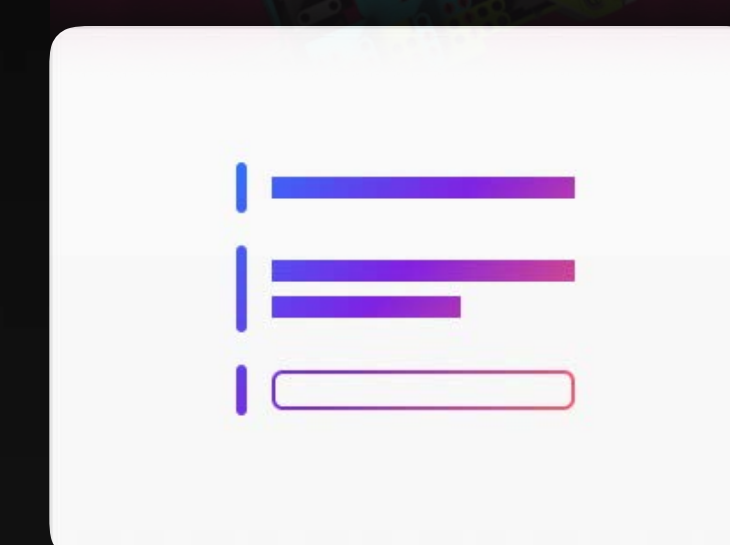
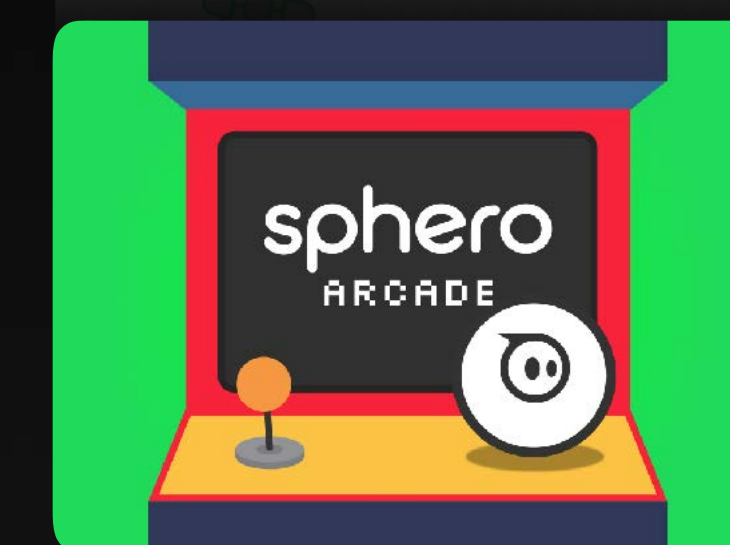
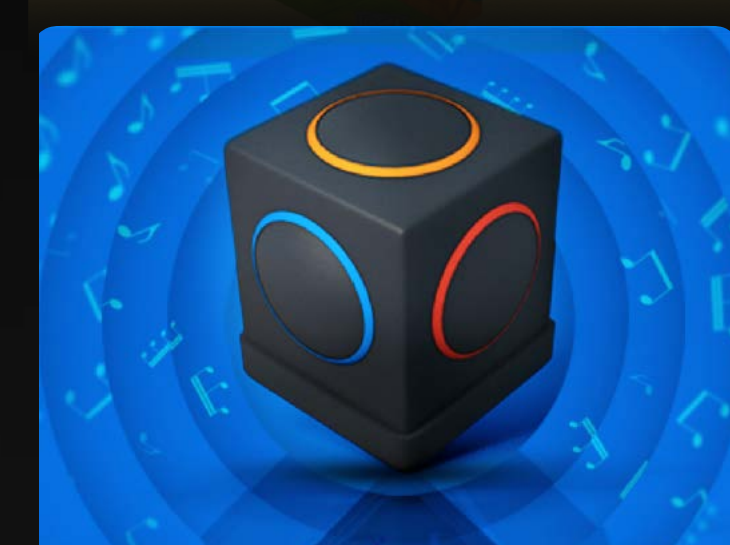
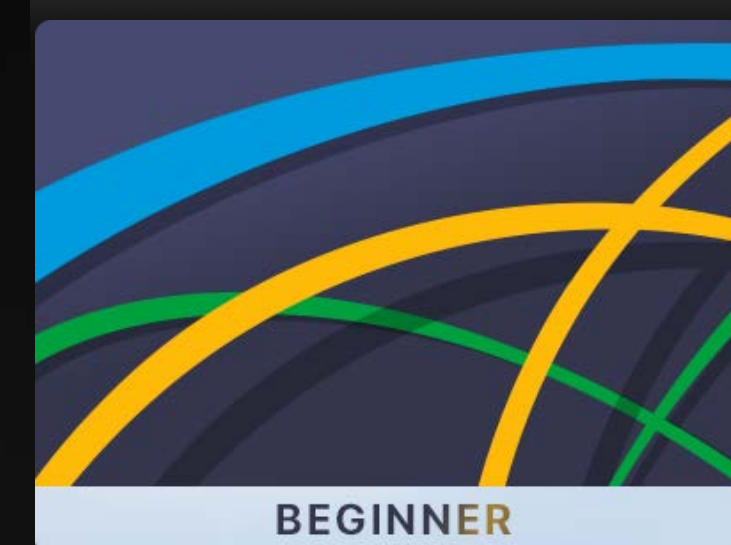
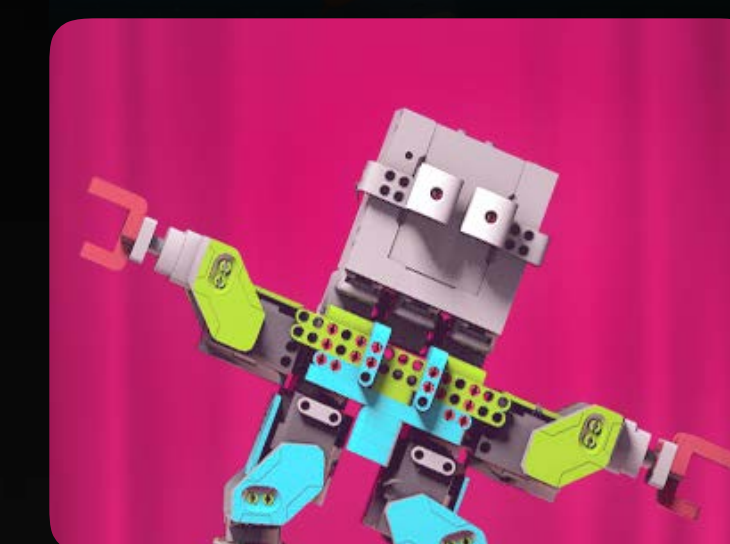
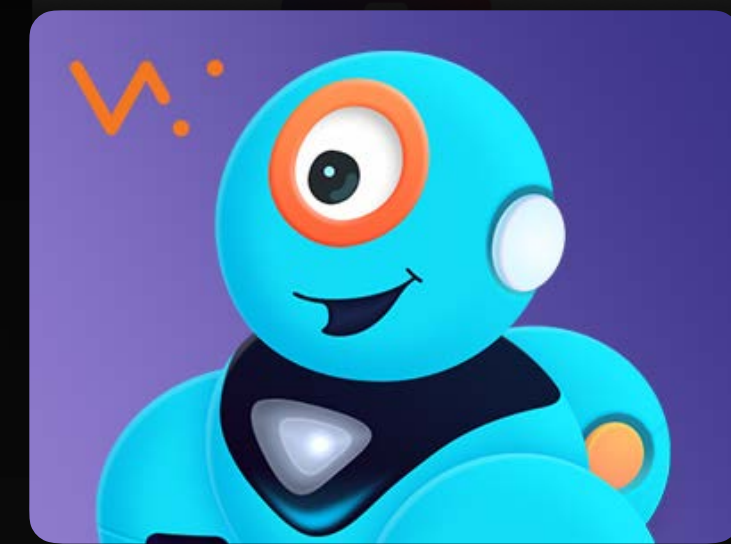
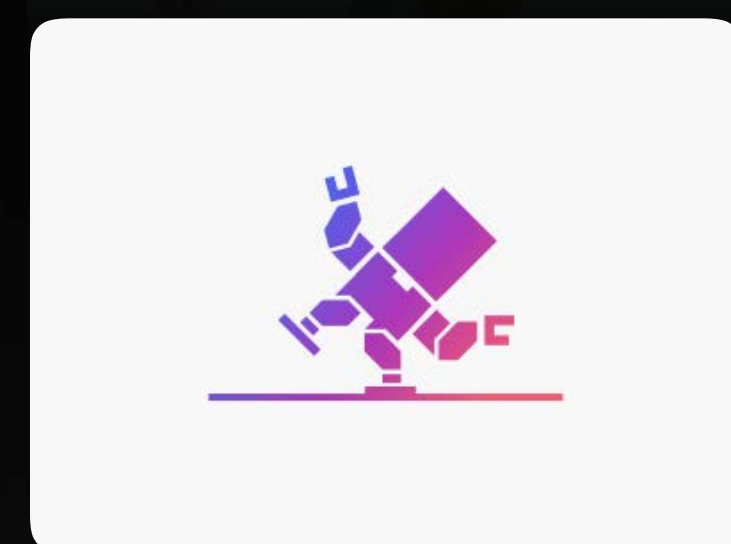
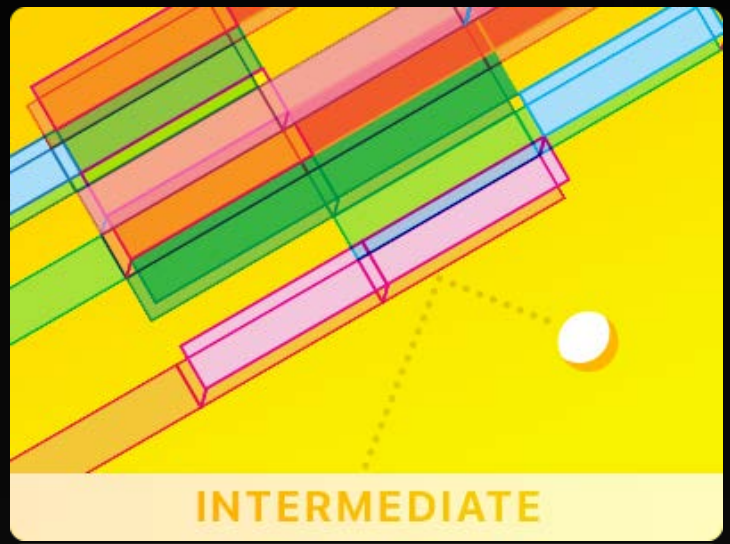
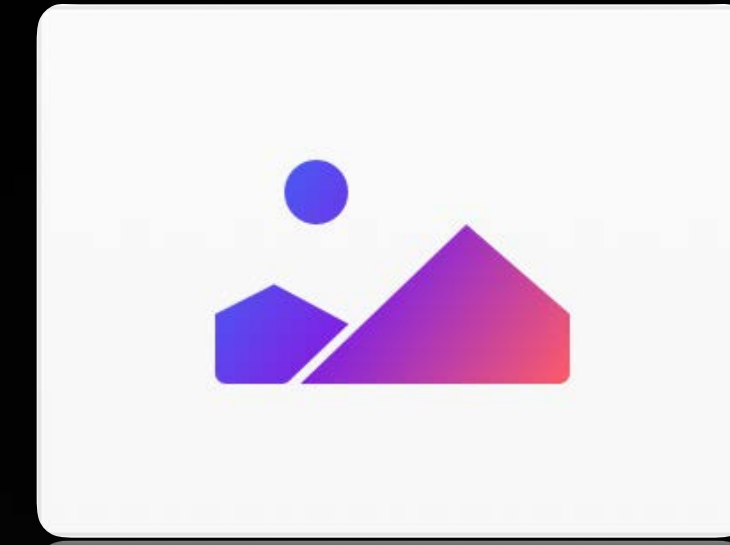
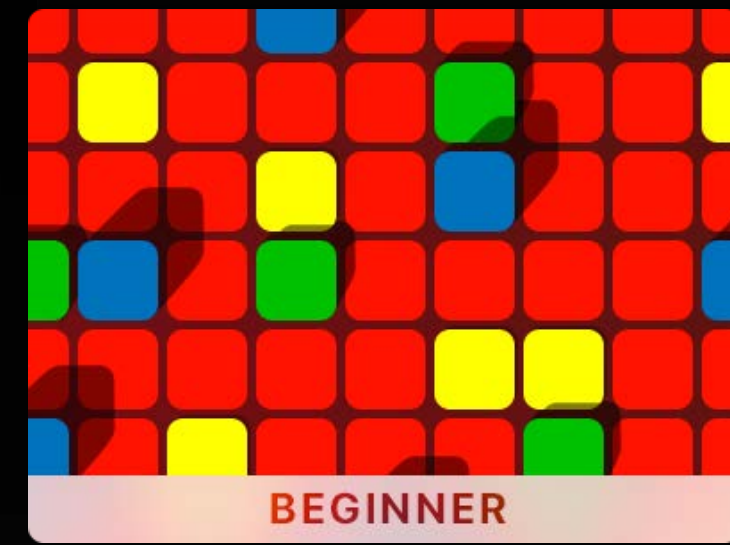
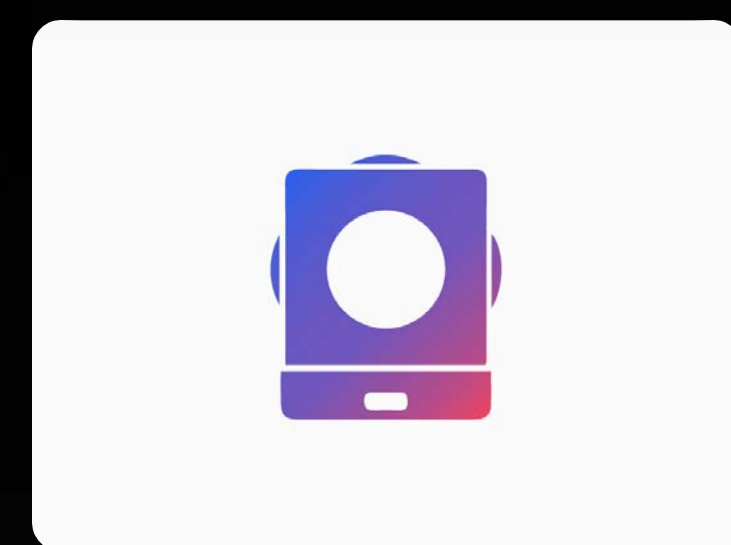
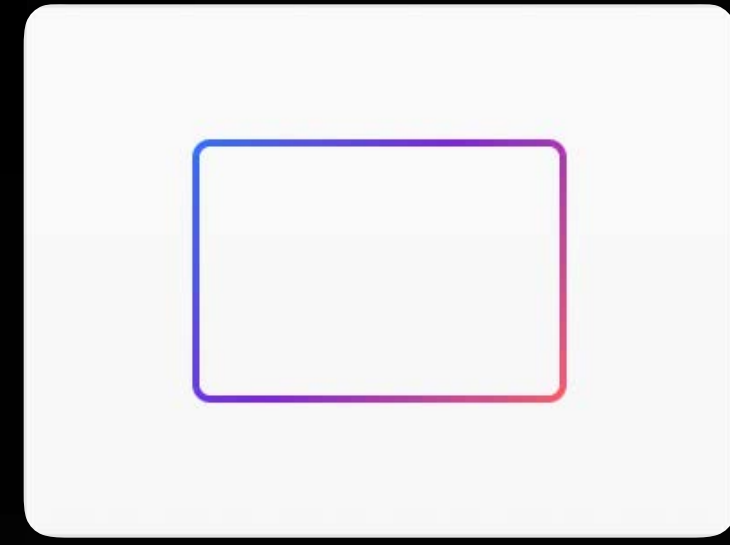
In the App Store

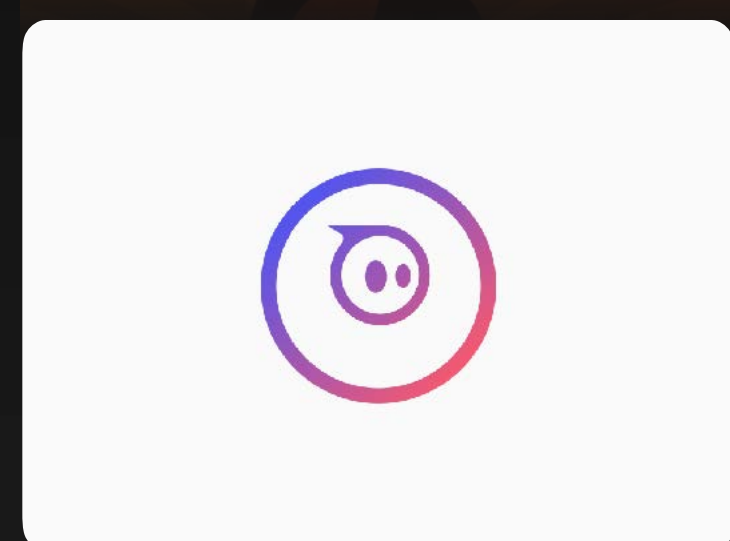
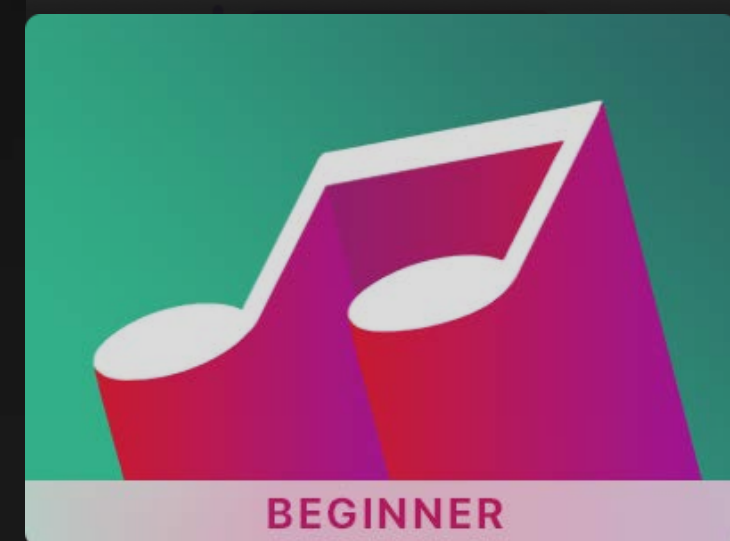
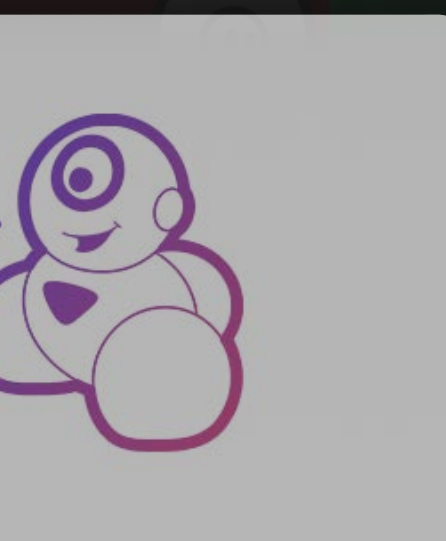
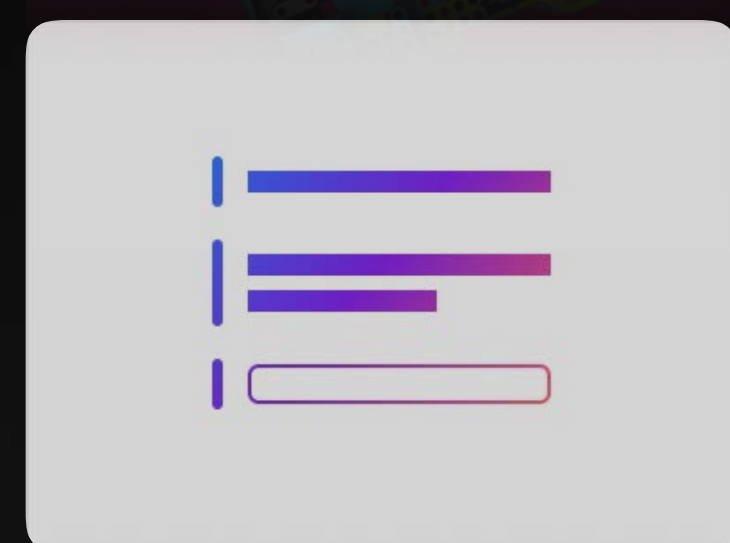
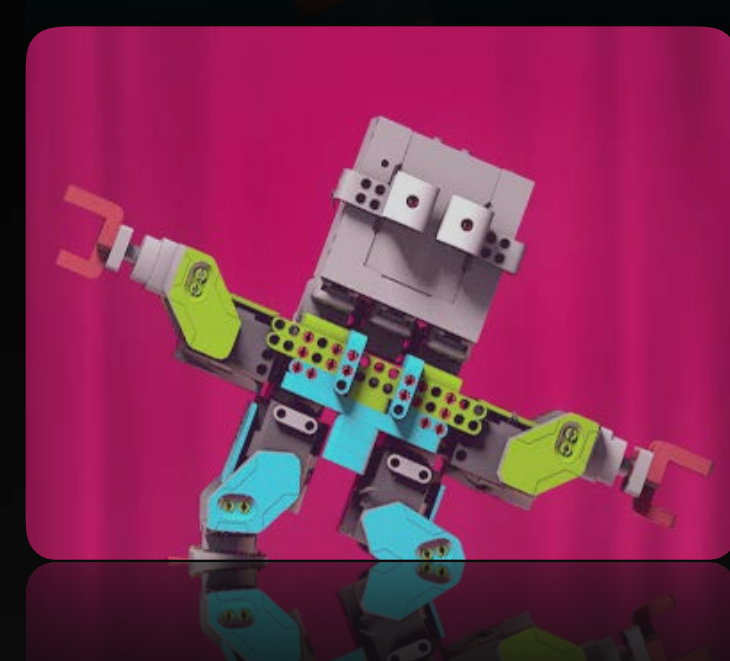
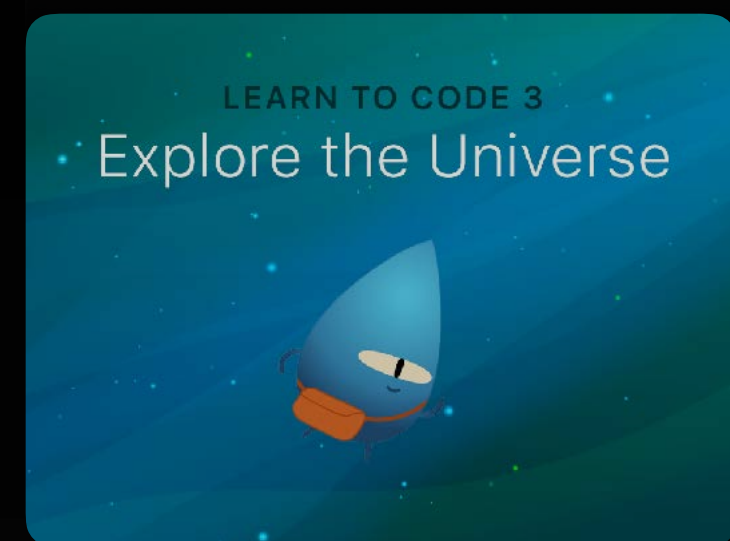


Available

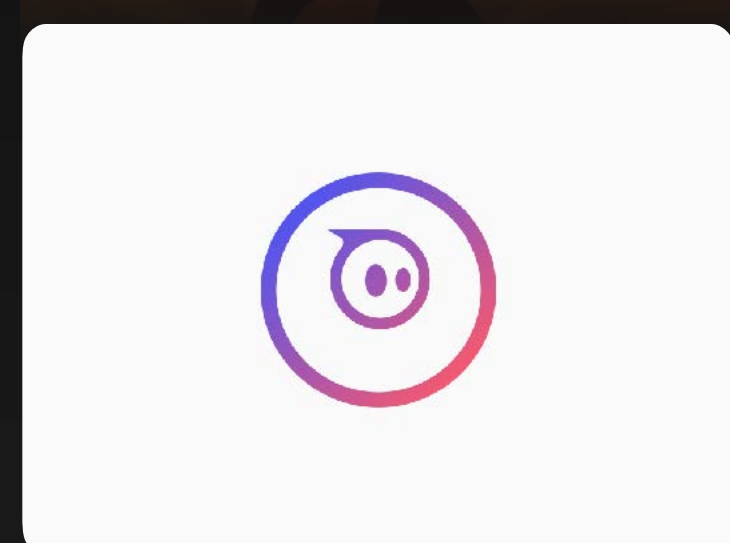
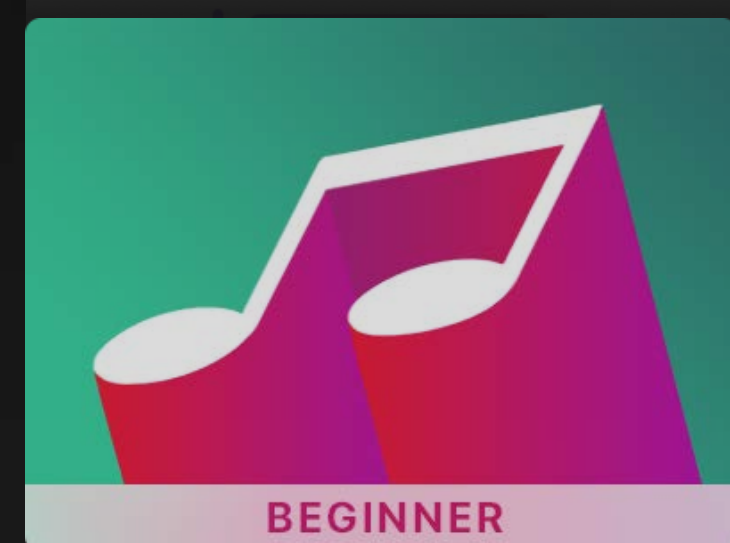
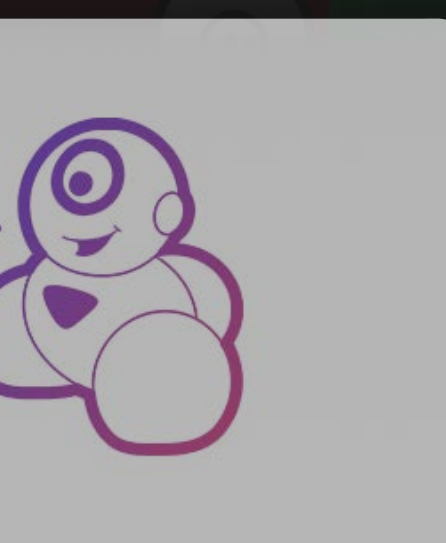
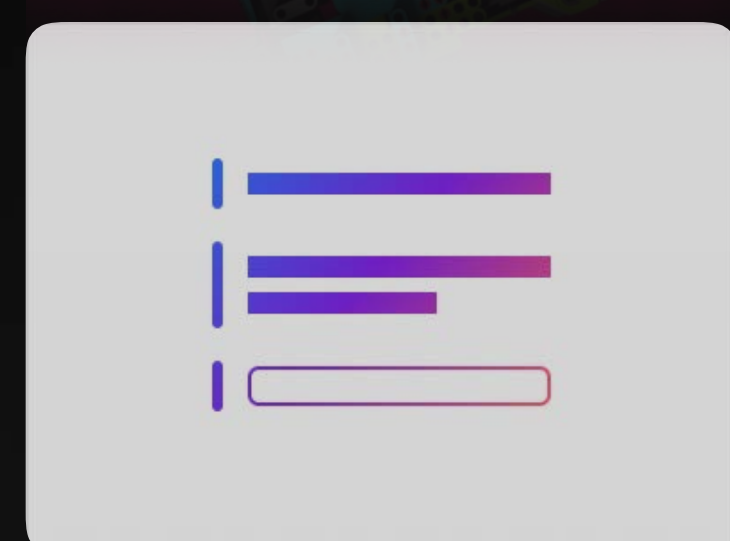
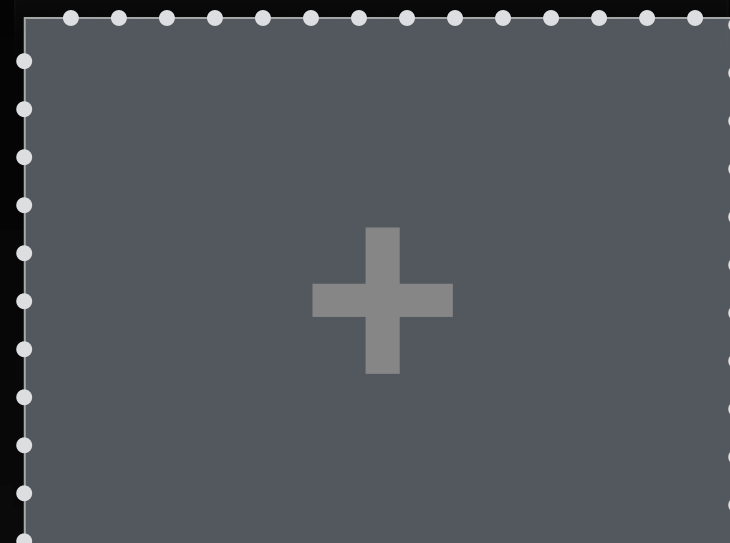
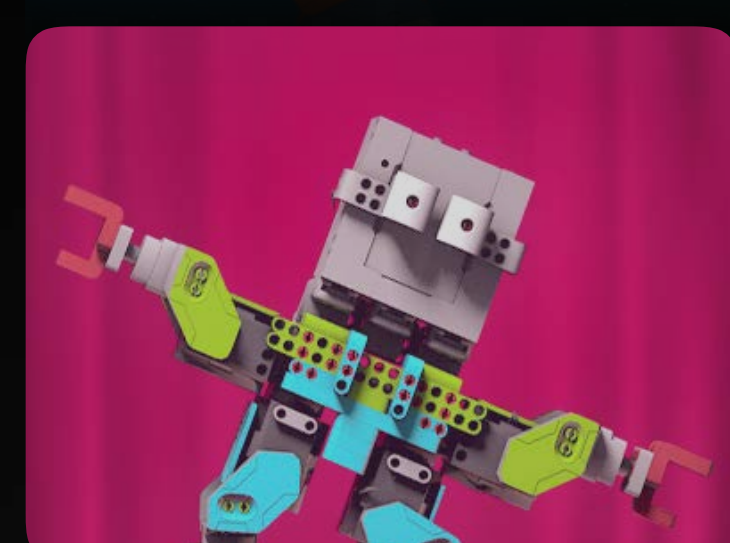
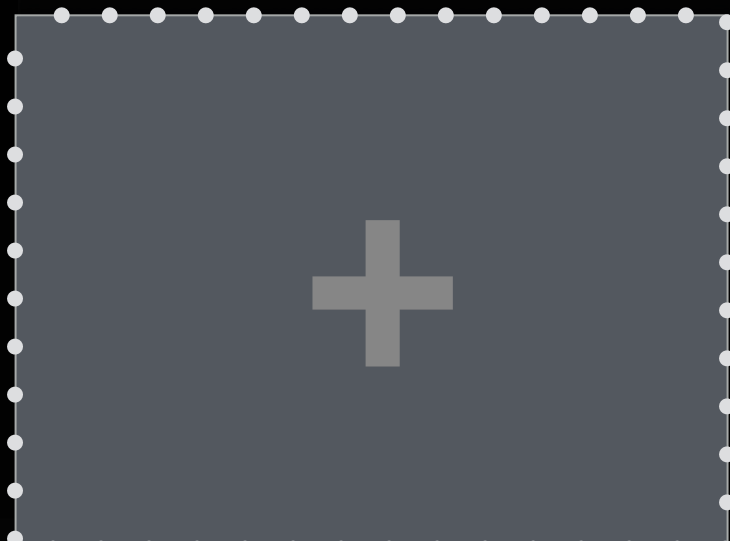
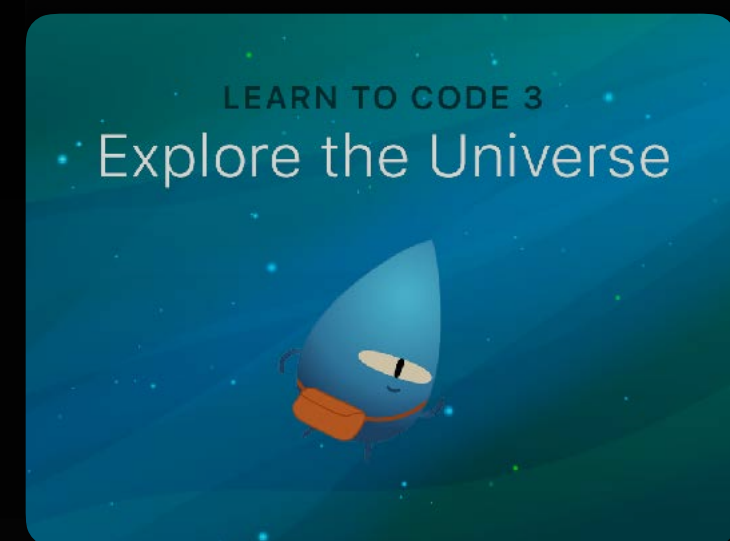
This Fall

Swift Playgrounds 2

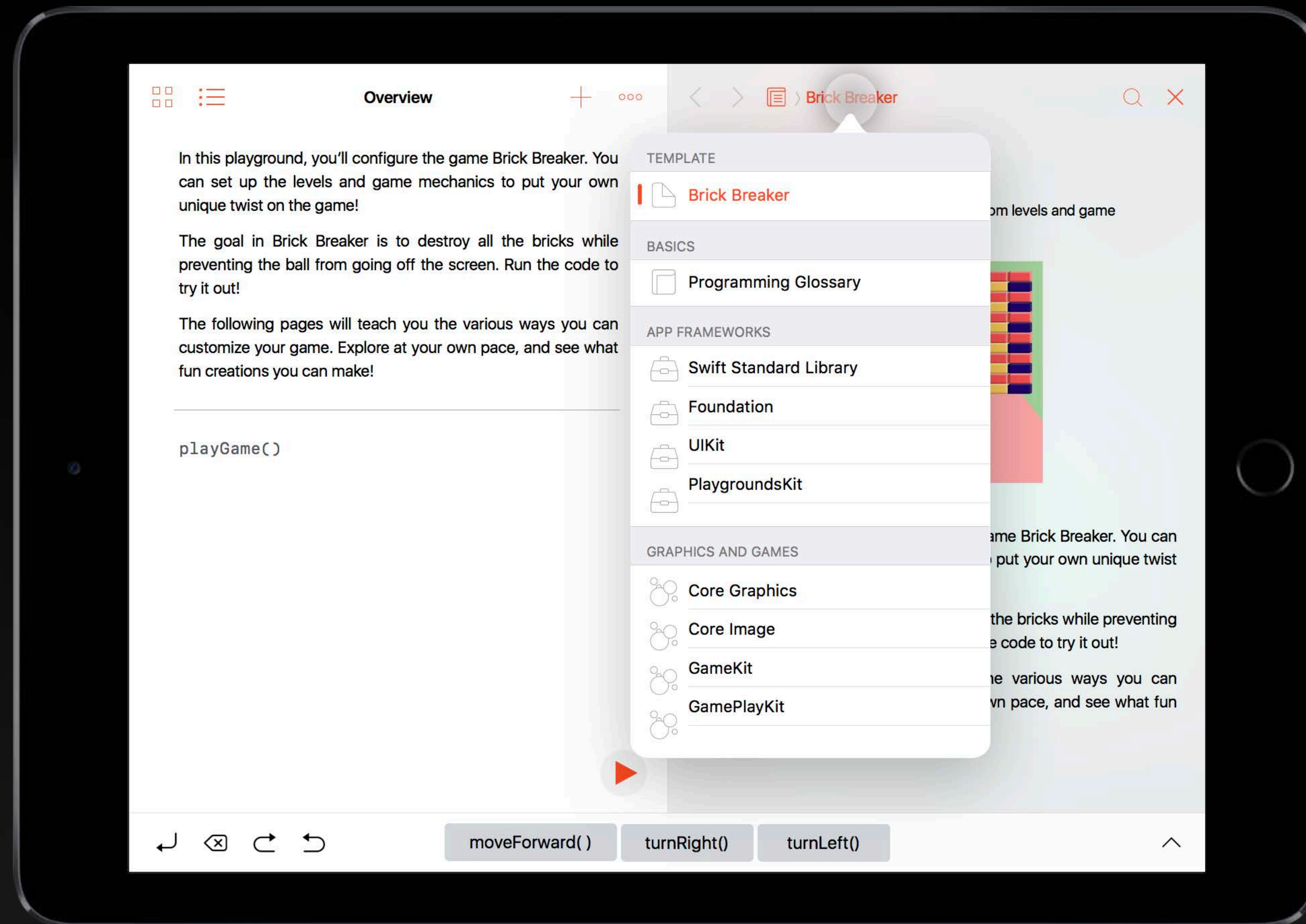




Playground Feeds
Author-hosted content
Subscription model
Update notifications
Localized content



Playground Feeds
Author-hosted content
Subscription model
Update notifications
Localized content



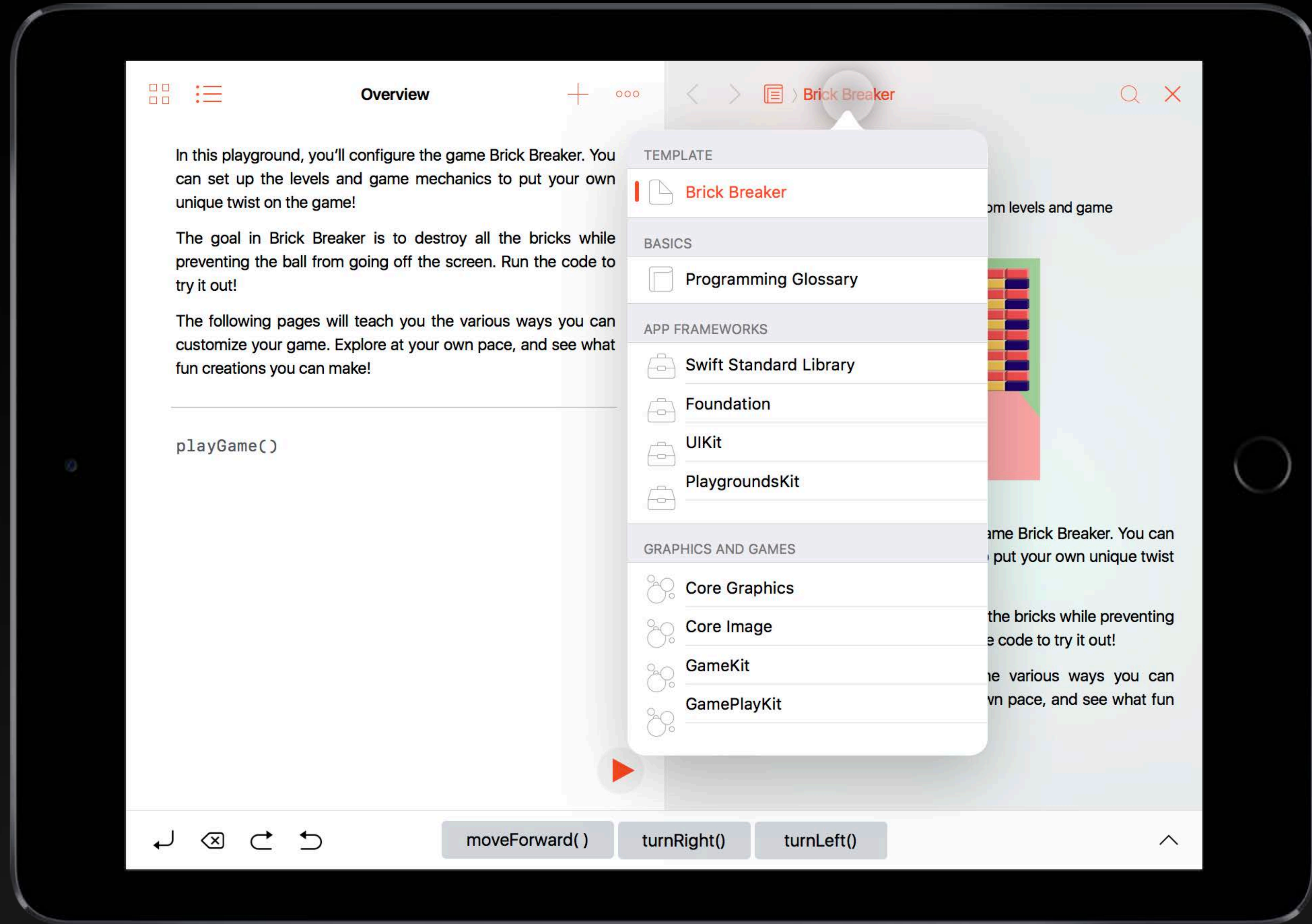
Available This Fall

Integrated API documentation

Swift 4 and Swift 3.2

iOS 11 SDK

Camera and Augmented Reality



Available This Fall

Eight additional localizations





Swift Playgrounds 2

Preview

Via TestFlight



Swift Playgrounds



Xcode 9




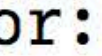










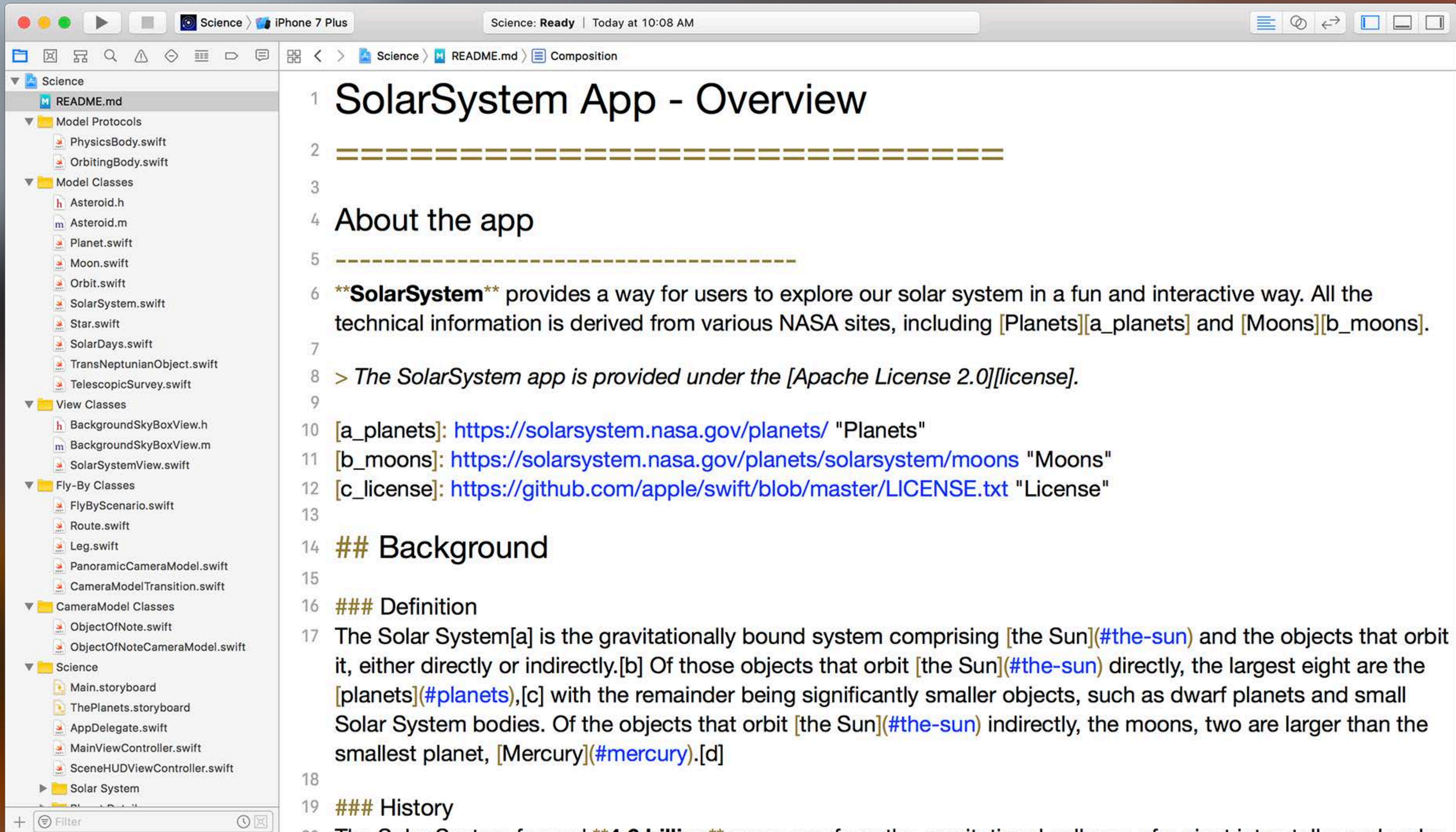
Source Editor

```
Science > iPhone 7 Plus | Science: Ready | Today at 9:28 AM
Science > Model Classes > SolarSystem.swift > SolarSystem

Science
├── README.md
├── Model Protocols
│   ├── PhysicsBody.swift
│   └── OrbitingBody.swift
├── Model Classes
│   ├── Asteroid.h
│   ├── Asteroid.m
│   ├── Planet.swift
│   ├── Moon.swift
│   ├── Orbit.swift
│   └── SolarSystem.swift
├── Star.swift
├── SolarDays.swift
├── TransNeptunianObject.swift
├── TelescopicSurvey.swift
├── View Classes
│   ├── BackgroundSkyBoxView.h
│   ├── BackgroundSkyBoxView.m
│   └── SolarSystemView.swift
├── Fly-By Classes
│   ├── FlyByScenario.swift
│   ├── Route.swift
│   ├── Leg.swift
│   ├── PanoramicCameraModel.swift
│   └── CameraModelTransition.swift
├── CameraModel Classes
│   ├── ObjectOfNote.swift
│   └── ObjectOfNoteCameraModel.swift
├── Science
│   ├── Main.storyboard
│   ├── ThePlanets.storyboard
│   ├── AppDelegate.swift
│   ├── MainViewController.swift
│   ├── SceneHUDViewController.swift
│   └── Solar System
└── ...

9
10 /// A model of a solar system, including a central #Star with orbiting #Planets.
11 public class SolarSystem {
12
13     // Create the sun.
14     let sun = Star(name: "Sun", color: )
15
16     // Create the planets.
17     let mercury = Planet(name: "Mercury", color: )
18     let venus = Planet(name: "Venus", color: )
19     let earth = Planet(name: "Earth", color: )
20     let mars = Planet(name: "Mars", color: )
21     let jupiter = Planet(name: "Jupiter", color: )
22     let saturn = Planet(name: "Saturn", color: )
23     let uranus = Planet(name: "Uranus", color: )
24     let neptune = Planet(name: "Neptune", color: )
25
26     // Create arrays for the objects in our SolarSystem.
27     let planets: [Planet]
28     let distantObjects: [TransNeptunianObject]
29
30     public init() {
31
32         // create the planets and distant objects.
33         planets = [mercury, venus, earth, mars, jupiter, saturn, uranus, venus]
34         distantObjects = SolarSystem.loadAndCreateDistantObjects(parentStar: sun)
35
36         // add Earth's moon.
37         earth.addMoon(Moon(name: "Moon", color: ))

```

```
29
30 public init() {
31
32     // create the planets and distant objects.
33     planets = [mercury, venus, earth, mars, jupiter, saturn, uranus, venus]
34     distantObjects = SolarSystem.loadAndCreateDistantObjects(parentStar: sunn)
35
36     // add Earth's moon.
37     earth.addMoon(Moon(name: "Moon", color: ))
38
39     // add Jupiter's 67 moons 😬!
40     if let path = Bundle.main.path(forResource: "MoonsOfJupiter", ofType: "txt") {
41         do {
42             // grab all the names of Jupiter's moons, which are separated by newlines.
43             let data = try String(contentsOfFile: path, encoding: .utf8)
44             let moonNames = data.components(separatedBy: .newlines)
45
46             // loop over all of Jupiter's moon names
```

🔴 Use of unresolved identifier 'sun' ✕
Did you mean 'sun'? Fix

300+
new diagnostics,
analyzers, and
fix-its

```

29
30     pub
31
32     // planets and distant objects.
33     [venus, earth, mars, jupiter, saturn, uranus, venus]
34     SolarSystem.loadAndCreateDistantObjects(parentStar: sunn)
35
36     // name: "Moon", color: ■)
37
38
39     // 57 moons 😬!
40     Bundle.main.path(forResource: "MoonsOfJupiter", ofType: "txt") {
41
42         // grab all the names of Jupiter's moons, which are separated by newlines.
43         let data = try String(contentsOfFile: path, encoding: .utf8)
44         let moonNames = data.components(separatedBy: .newlines)
45
46         // loop over all of Jupiter's moon names

```

Use of unresolved identifier 'sun' ✕
 Did you mean 'sun'? Fix



File opening



File opening



Scrolling



File opening



Scrolling



Jump to line

```
Science > iPhone 7 Plus | Science: Ready | Today at 11:00 AM
Science > Model Classes > SolarSystem.swift > SolarSystem

Science
├── README.md
├── Model Protocols
│   ├── PhysicsBody.swift
│   └── OrbitingBody.swift
├── Model Classes
│   ├── Asteroid.h
│   ├── Asteroid.m
│   ├── Planet.swift
│   ├── Moon.swift
│   ├── Orbit.swift
│   └── SolarSystem.swift
├── Star.swift
├── SolarDays.swift
├── TransNeptunianObject.swift
├── TelescopicSurvey.swift
├── View Classes
│   ├── BackgroundSkyBoxView.h
│   ├── BackgroundSkyBoxView.m
│   └── SolarSystemView.swift
├── Fly-By Classes
│   ├── FlyByScenario.swift
│   ├── Route.swift
│   ├── Leg.swift
│   ├── PanoramicCameraModel.swift
│   └── CameraModelTransition.swift
├── CameraModel Classes
│   ├── ObjectOfNote.swift
│   └── ObjectOfNoteCameraModel.swift
├── Science
│   ├── Main.storyboard
│   ├── ThePlanets.storyboard
│   ├── AppDelegate.swift
│   ├── MainViewController.swift
│   └── SceneHUDViewController.swift
└── Solar System

25
26 // Create arrays for the objects in our SolarSystem.
27 let planets: [Planet]
28 let distantObjects: [TransNeptunianObject]
29
30 public init() {
31
32     // create the planets and distant objects.
33     planets = [mercury, venus, earth, mars, jupiter, saturn, uranus, venus]
34     distantObjects = SolarSystem.loadAndCreateDistantObjects(parentStar: sun)
35
36     // add Earth's moon.
37     earth.addMoon(Moon(name: "Moon", color: ))
38
39     // add Jupiter's 67 moons 😬!
40     if let path = Bundle.main.path(forResource: "MoonsOfJupiter", ofType: "txt") {
41         do {
42             // grab all the names of Jupiter's moons, which are separated by newlines.
43             let data = try String(contentsOfFile: path, encoding: .utf8)
44             let moonNames = data.components(separatedBy: .newlines)
45
46             // loop over all of Jupiter's moon names.
47             for moonName in moonNames {
48                 jupiter.addMoon(Moon(name: moonName, color: ))
49             }
50         } catch { /* shouldn't end up here. */ }
51     }
52 }
53
54 /// Calculates the coordinate of a given #Planet, at a particular date (which includes time),
    relative to the Sun.
```

```
Science > iPhone 7 Plus | Science: Ready | Today at 11:00 AM
Science > Model Classes > SolarSystem.swift > SolarSystem

Science
├── README.md
├── Model Protocols
│   ├── PhysicsBody.swift
│   └── OrbitingBody.swift
├── Model Classes
│   ├── Asteroid.h
│   ├── Asteroid.m
│   ├── Planet.swift
│   ├── Moon.swift
│   ├── Orbit.swift
│   └── SolarSystem.swift
├── Star.swift
├── SolarDays.swift
├── TransNeptunianObject.swift
├── TelescopicSurvey.swift
├── View Classes
│   ├── BackgroundSkyBoxView.h
│   ├── BackgroundSkyBoxView.m
│   └── SolarSystemView.swift
├── Fly-By Classes
│   ├── FlyByScenario.swift
│   ├── Route.swift
│   ├── Leg.swift
│   ├── PanoramicCameraModel.swift
│   └── CameraModelTransition.swift
├── CameraModel Classes
│   ├── ObjectOfNote.swift
│   └── ObjectOfNoteCameraModel.swift
├── Science
│   ├── Main.storyboard
│   ├── ThePlanets.storyboard
│   ├── AppDelegate.swift
│   ├── MainViewController.swift
│   └── SceneHUDViewController.swift
└── Solar System

25
26 // Create arrays for the objects in our SolarSystem.
27 let planets: [Planet]
28 let distantObjects: [TransNeptunianObject]
29
30 public init() {
31
32     // create the planets and distant objects.
33     planets = [mercury, venus, earth, mars, jupiter, saturn, uranus, venus]
34     distantObjects = SolarSystem.loadAndCreateDistantObjects(parentStar: sun)
35
36     // add Earth's moon.
37     earth.addMoon(Moon(name: "Moon", color: ))
38
39     // add Jupiter's 67 moons 😬!
40     if let path = Bundle.main.path(forResource: "MoonsOfJupiter", ofType: "txt") {
41         do {
42             // grab all the names of Jupiter's moons, which are separated by newlines.
43             let data = try String(contentsOfFile: path, encoding: .utf8)
44             let moonNames = data.components(separatedBy: .newlines)
45
46             // loop over all of Jupiter's moon names.
47             for moonName in moonNames {
48                 jupiter.addMoon(Moon(name: moonName, color: ))
49             }
50         } catch { /* shouldn't end up here. */ }
51     }
52 }
53
54 /// Calculates the coordinate of a given #Planet, at a particular date (which includes time),
    relative to the Sun.
```


Science > iPhone 7 Plus | Science: Ready | Today at 11:00 AM

Science > Model Classes > SolarSystem.swift > SolarSystem

Science

- README.md
- Model Protocols
 - PhysicsBody.swift
 - OrbitingBody.swift
- Model Classes
 - Asteroid.h
 - Asteroid.m
 - Planet.swift
 - Moon.swift
 - Orbit.swift
 - SolarSystem.swift**
 - Star.swift
 - SolarDays.swift
 - TransNeptunianObject.swift
 - TelescopicSurvey.swift
- View Classes
 - BackgroundSkyBoxView.h
 - BackgroundSkyBoxView.m
 - SolarSystemView.swift
- Fly-By Classes
 - FlyByScenario.swift
 - Route.swift
 - Leg.swift
 - PanoramicCameraModel.swift
 - CameraModelTransition.swift
- CameraModel Classes
 - ObjectOfNote.swift
 - ObjectOfNoteCameraModel.swift
- Science
 - Main.storyboard
 - ThePlanets.storyboard
 - AppDelegate.swift
 - MainViewController.swift
 - SceneHUDViewController.swift
 - Solar System

```
25
26 // Create arrays for the objects in our SolarSystem.
27 let planets: [Planet]
28 let distantObjects: [TransNeptunianObject]
29
30 public init() {
31
32 // create the planets and distant objects.
33 planets = [mercury, venus, earth, mars, jupiter, saturn, uranus, venus]
34 distantObjects = [TransNeptunianObject(parentStar: sun)
35
36 // add earth
37
38 // add jupiter
39
40 if
41
42
43
44
45
46
47 for moonName in moonNames {
48     jupiter.addMoon(Moon(name: moonName, color: #f0e68c))
49 }
50 } catch { /* shouldn't end up here. */ }
51 }
52
53
54 /// Calculates the coordinate of a given #Planet, at a particular date (which includes time),
    relative to the Sun.
```

Actions

- +{ Add "else" Statement
- }|{ Add "else if" Statement
- ☰() Extract Method

Refactoring

All new engine and workflow for Swift, Objective-C, C, and C++

Q Actions

-]+{ Add "else" Statement
- }] { Add "else if" Statement**
- ⌘ Extract Method

Q Actions

- ⌘ Jump to Definition ^⌘
- ⌘ Show Quick Help ⌘
- ⌘ Edit All in Scope
- ⌘ Rename...**

Q Actions

- ⌘ Jump to Definition ^⌘
- ⌘ Show Quick Help ⌘
- ⌘ Edit All in Scope
- +() Add Method**
- ⌘ Add Property
- ⌘ Rename...

Q Actions

- ⌘ Add "case" Statement**
- ⌘ Add "default" Statement
- ⌘ Extract Method

Q Actions

- ⌘ Jump to Definition ^⌘
- ⌘ Show Quick Help ⌘
- (+) Add Parameter**

Refactoring

Demo

Science: Ready | Today at 11:37 AM

Science

- README.md
- Model Protocols
 - PhysicsBody.swift
 - OrbitingBody.swift
- Model Classes
 - Asteroid.h
 - Asteroid.m
 - Planet.swift
 - Moon.swift
 - Orbit.swift
 - SolarSystem.swift
 - Star.swift
 - SolarDays.swift
 - TransNeptunianObject.swift
 - TelescopicSurvey.swift
- View Classes
 - BackgroundSkyBoxView.h
 - BackgroundSkyBoxView.m
 - SolarSystemView.swift
- Fly-By Classes
 - FlyByScenario.swift
 - Route.swift
 - Leg.swift
 - PanoramicCameraModel.swift
 - CameraModelTransition.swift
- CameraModel Classes
 - ObjectOfNote.swift
 - ObjectOfNoteCameraModel.swift
- Science
 - Main.storyboard
 - ThePlanets.storyboard
 - AppDelegate.swift
 - MainViewController.swift
 - SceneHUDViewController.swift
 - Solar System

Rename

All (41) Code (36) File Names (1) Comments (2) Other (2)

Cancel Rename

SolarSystem.swift

```
27 let planets: [Planet]
28 let distantObjects: [MinorMoon]
29
91 /// - Returns: an array of distant objects.
92 private static func loadAndCreateDistantObjects(parentStar: Star) -> [MinorMoon] {
93     var distantObjects: [MinorMoon] = []
94     if let path = Bundle.main.path(forResource: "DistantObjects", ofType: "txt") {
98         let distantObjectNames = data.components(separatedBy: .newlines)
99         let defaultShape = MinorMoon.Shape.spherical
100
101         // loop over all of distant object names.
102         for distantObjectName in distantObjectNames {
103             distantObjects.append(MinorMoon(name: distantObjectName, shape: defaultShape,
104                                     parentStar: parentStar))
104         }

```

ThePlanets.storyboard

Trans Neptunian Object

Class Name: MinorMoon

Class Name: MinorMoon

Asteroid.m

```
12
13 - (void)addDistantObject:(MinorMoon *)object
14 {
15     NSMutableArray<MinorMoon *> *mutableObjects = self.distantObjects.mutableCopy;
16     [mutableObjects addObject:object];
17     NSArray<MinorMoon *> *newDistantObjects = [NSArray arrayWithArray:mutableObjects];

```



Transformations

Add missing protocol requirements

Generate missing implementation stubs

Add missing overrides for abstract methods

Extract to local variable

Extract method / expression

Expand 'default' in switch statements

Convert if / else to / from switch statement

Wrap string in NSLocalizedString macro



Open Source Transformation Engine



Source Editor



Source Editor with Refactoring



Swift

250,000

Swift apps on the App Store

IBM

SAP



Source compatibility

Dictionary enhancements

Smaller binaries

Easier to use String

Improved Package Manager

Class-constrained protocols

Memory exclusivity

Encoding and decoding

Smart keypaths

Numeric protocols

Improved generics



String

String

Easier to use

More Unicode correct

Fast

Swift 3

```
var s = "Hello!"  
for c in s.characters { print(c) }
```

```
s.characters.last == "!"  
s.characters.index(of: "!")
```

```
String(s.characters.dropLast(1)) + " world!"
```

Swift 3

```
var s = "Hello!"  
for c in s.characters { print(c) }
```

```
s.characters.last == "!"  
s.characters.index(of: "!")
```

```
String(s.characters.dropLast(1)) + " world!"
```

Swift 3

```
var s = "Hello!"  
for c in s.characters { print(c) }  
  
s.characters.last == "!"  
s.characters.index(of: "!")  
  
String(s.characters.dropLast(1)) + " world!"
```

```
var s = "Hello!"  
for c in s { print(c) }
```

```
s.last == "!"  
s.index(of: "!")
```

```
s.dropLast(1) + " world!"
```

String **simplified**

Strings are now range-replaceable,
bidirectional Collections

```
var s = "Hello!"  
for c in s { print(c) }
```

```
s.last == "!"  
s.index(of: "!")
```

```
s.dropLast(1) + " world!"
```

```
let poem = """
'Twas brillig, and the slithy toves
    Did gyre and gimble in the wabe:
All mimsy were the borogoves,
    And the mome raths outgrabe.
"""
```

Multi-Line String Literals

```
let index = poem.index(of:",")!  
let substring = poem[..<poem.index] // 'Twas brillig
```

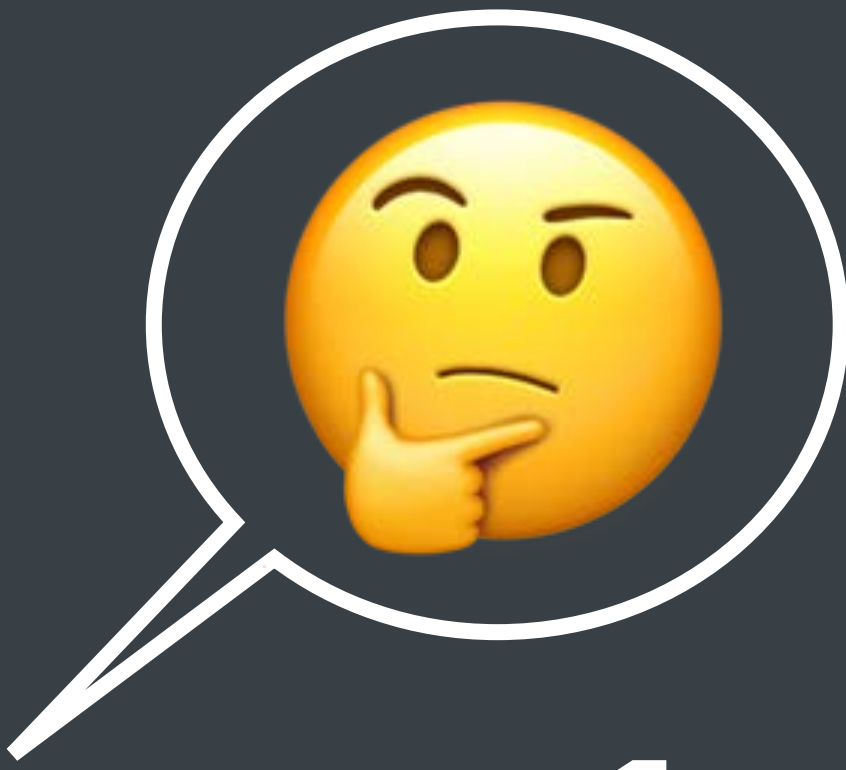
Simple and Expressive Slicing

Swift 3

```
"🇸🇴🇷🇺🇰🇪🇰🇲".count == 1
```


Swift 3

"🇸🇴🇷🇺🇰🇲🇳" .count == 1







1F1F8

1F1F8



1F1EC

1F1F1



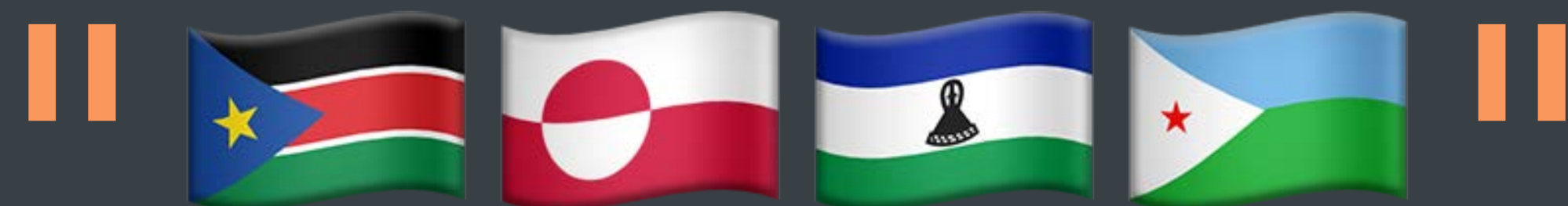
1F1F1

1F1F8



1F1E9

1F1EF



1F1F8

1F1F8

1F1EC

1F1F1

1F1F1

1F1F8

1F1E9

1F1EF

Unicode 9 Grapheme Breaking



1F1F8

1F1F8

1F1EC

1F1F1

1F1F1

1F1F8

1F1E9

1F1EF

Unicode 9 Grapheme Breaking

"🇸🇴🇷🇺🇮🇰🇪" .count == 4

Unicode 9 Grapheme Breaking



English

French

German

Spanish

String processing



Simplified Chinese
Japanese

String processing

Codable

```
public struct Farm {  
    public let name: String  
    public let location: Location  
    public let animals: [Animal]  
}
```

```
public struct Farm      {
    public let name: String
    public let location: Location
    public let animals: [Animal]
}
```

```
public struct Farm: Codable {  
    public let name: String  
    public let location: Location  
    public let animals: [Animal]  
}
```

```
public struct Farm: Codable {  
    public let name: String  
    public let location: Location  
    public let animals: [Animal]  
}
```

```
let farm = Farm(name: "Old MacDonald's Farm",  
                location: Location(latitude: 51.6216,  
                                   longitude: 0.2692),  
                animals: [.cow, .dog, .chicken])
```

```
let farm = Farm(name: "Old MacDonald's Farm",  
                location: Location(latitude: 51.6216,  
                                   longitude: 0.2692),  
                animals: [.cow, .dog, .chicken])
```

```
let farm = Farm(name: "Old MacDonald's Farm",  
                location: Location(latitude: 51.6216,  
                                   longitude: 0.2692),  
                animals: [.cow, .dog, .chicken])
```

```
let payload: Data = try JSONEncoder().encode(farm)
```

```
let payload: Data = try JSONEncoder().encode(farm)
```



```
let payload: Data = try JSONEncoder().encode(farm)
```

```
{  
  "name": "Old MacDonald's Farm",  
  "location": {"longitude": 0.2692, "latitude": 51.6216},  
  "animals": [0, 1, 2]  
}
```

JSON

```
let payload: Data = try JSONEncoder().encode(farm)
```

```
{  
  "name": "Old MacDonald's Farm",  
  "location": {"longitude": 0.2692, "latitude": 51.6216},  
  "animals": [0, 1, 2]  
}
```

JSON

```
let farm = try JSONDecoder().decode(Farm.self, from: payload)  
let coordinates = farm.location
```

Easy to Adopt

▶ Swift Language Version

▶ Swift Language Version

Swift 3.2

✓ Swift 4.0

Swift 3.2

✓ Swift 4.0

One compiler

Two Swift language modes

Mix-and-match 3.2 and 4.0 targets



Swift 3.2

Build projects with no modifications

Use almost all new Swift features

Use new APIs from SDK

No Swift-improvements to existing APIs



Swift 4.0

Use all the new language features

Improved performance

SDK improvements for Swift

Some migration required

Building Large Projects

40% faster

Building large mix-and-match Swift/Objective-C projects

2x faster

Building projects using multiple WMO targets



Swift 4



Core Technologies



Indexer



**Open Quickly
first result**



**Open Quickly
first result**



**Large project
search**

 Building 'SolarSystem' (and indexing) ...

Indexing While Building



Build System



New Build System



Built in Swift

Layers atop `11build`

Modern architecture



Process separation

Unified dependency graph

Improved configuration analysis

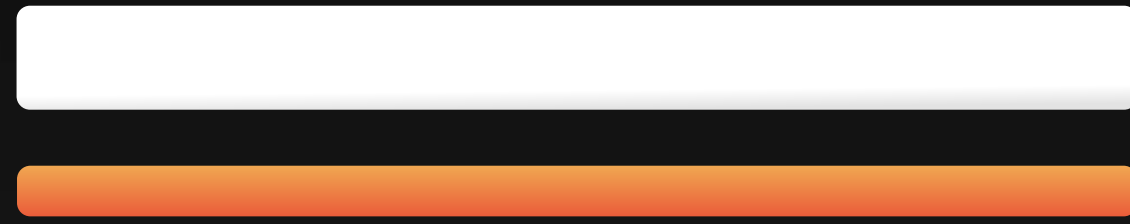
Increased parallelism and caching



Build System



Build Tools



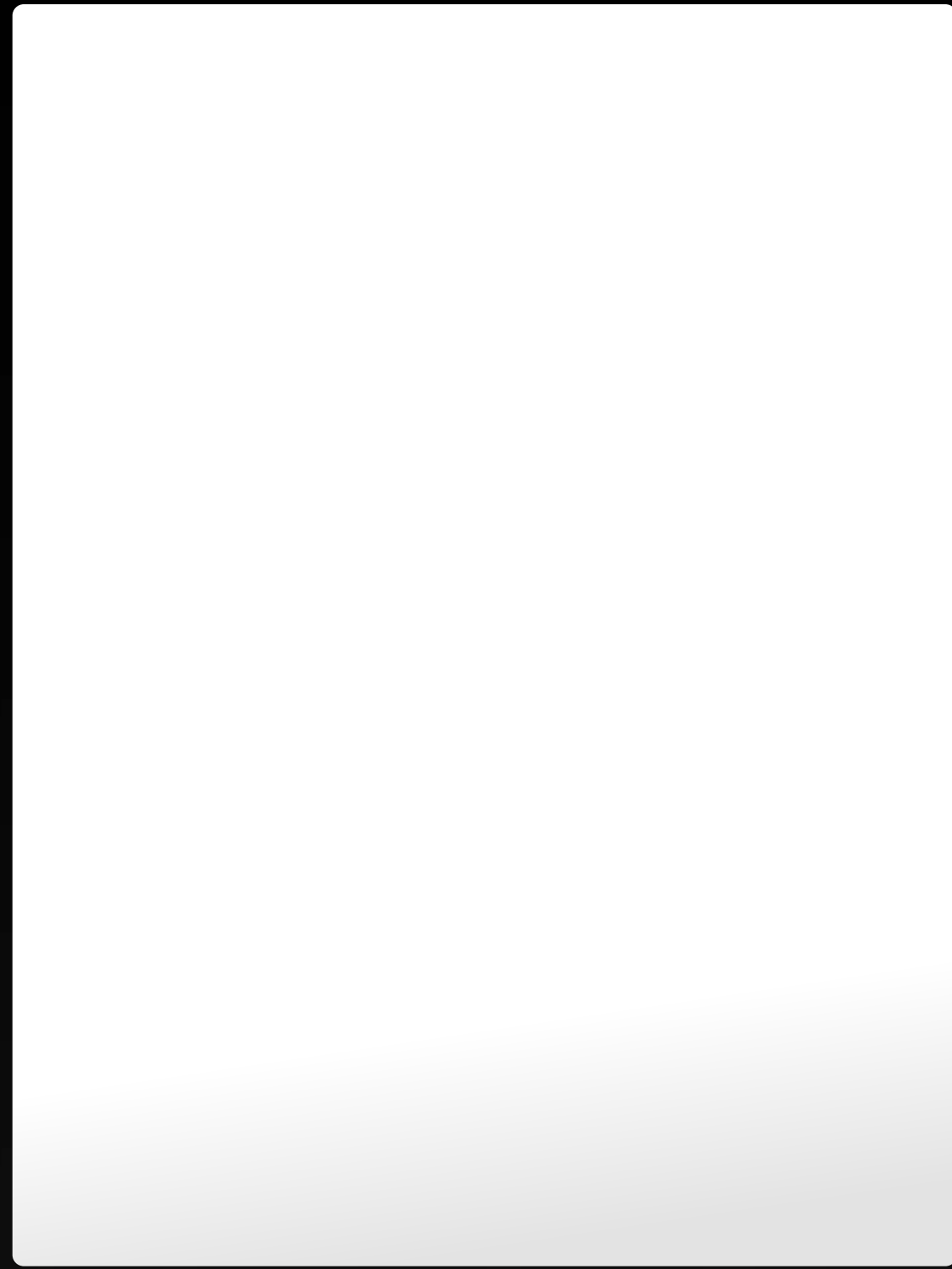
Small project



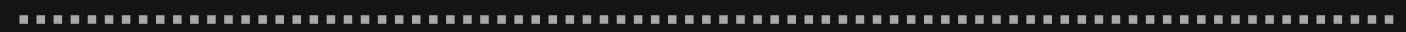
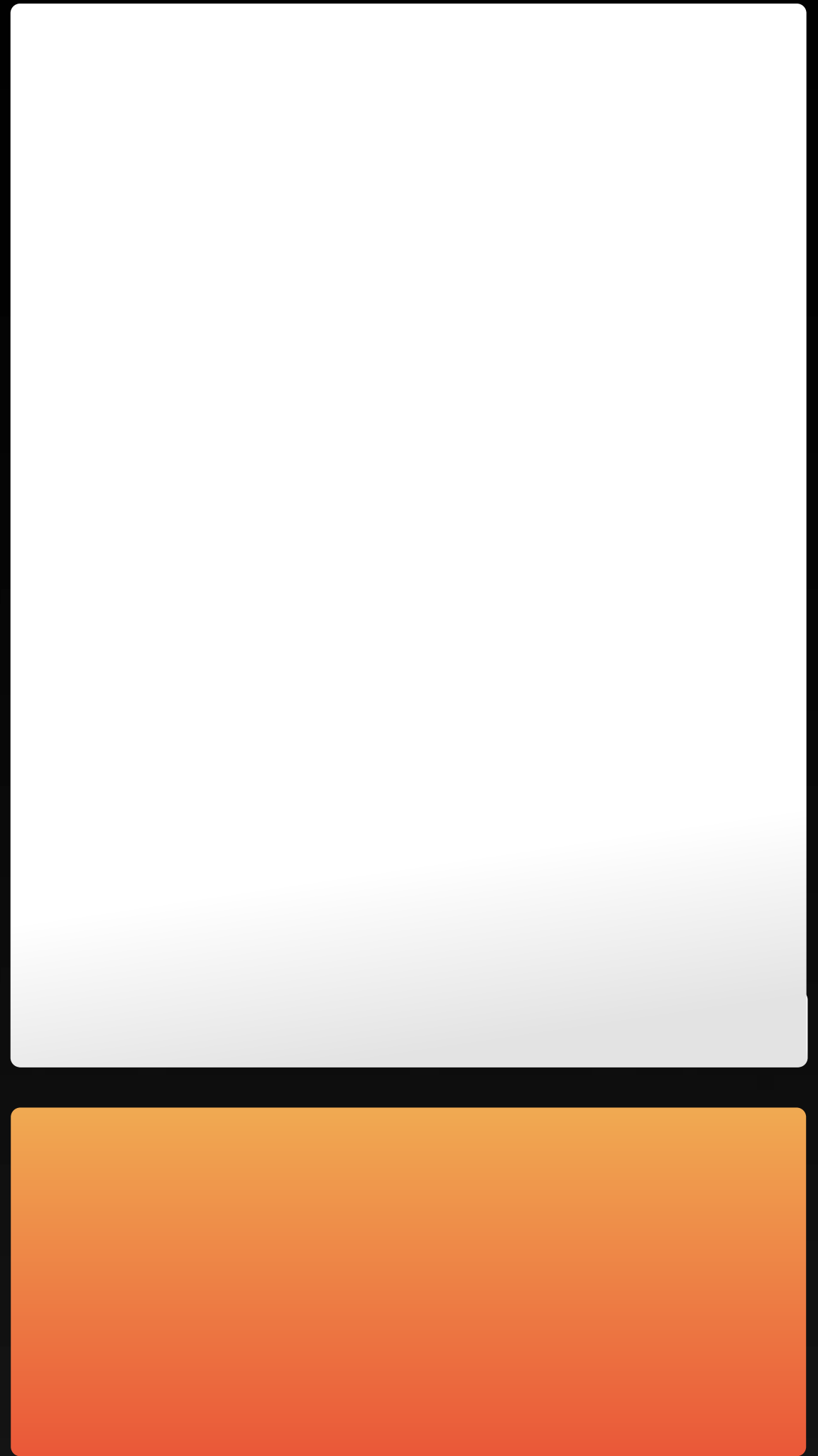
Medium project



Large project



Full build



Full build



Incremental build



Incremental build

Up to
2.5x

Faster build operations

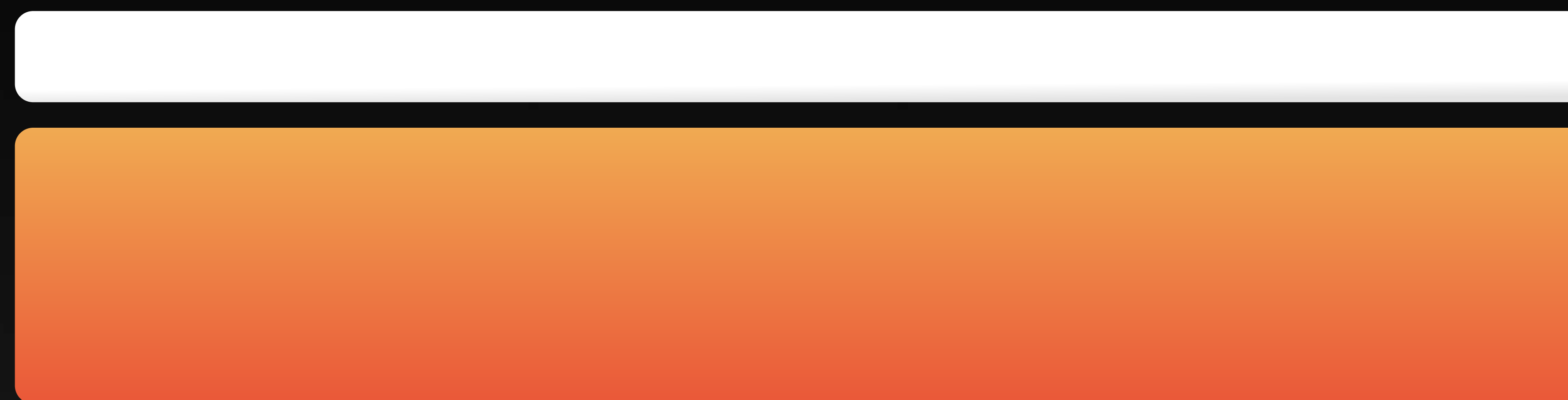


Incremental build

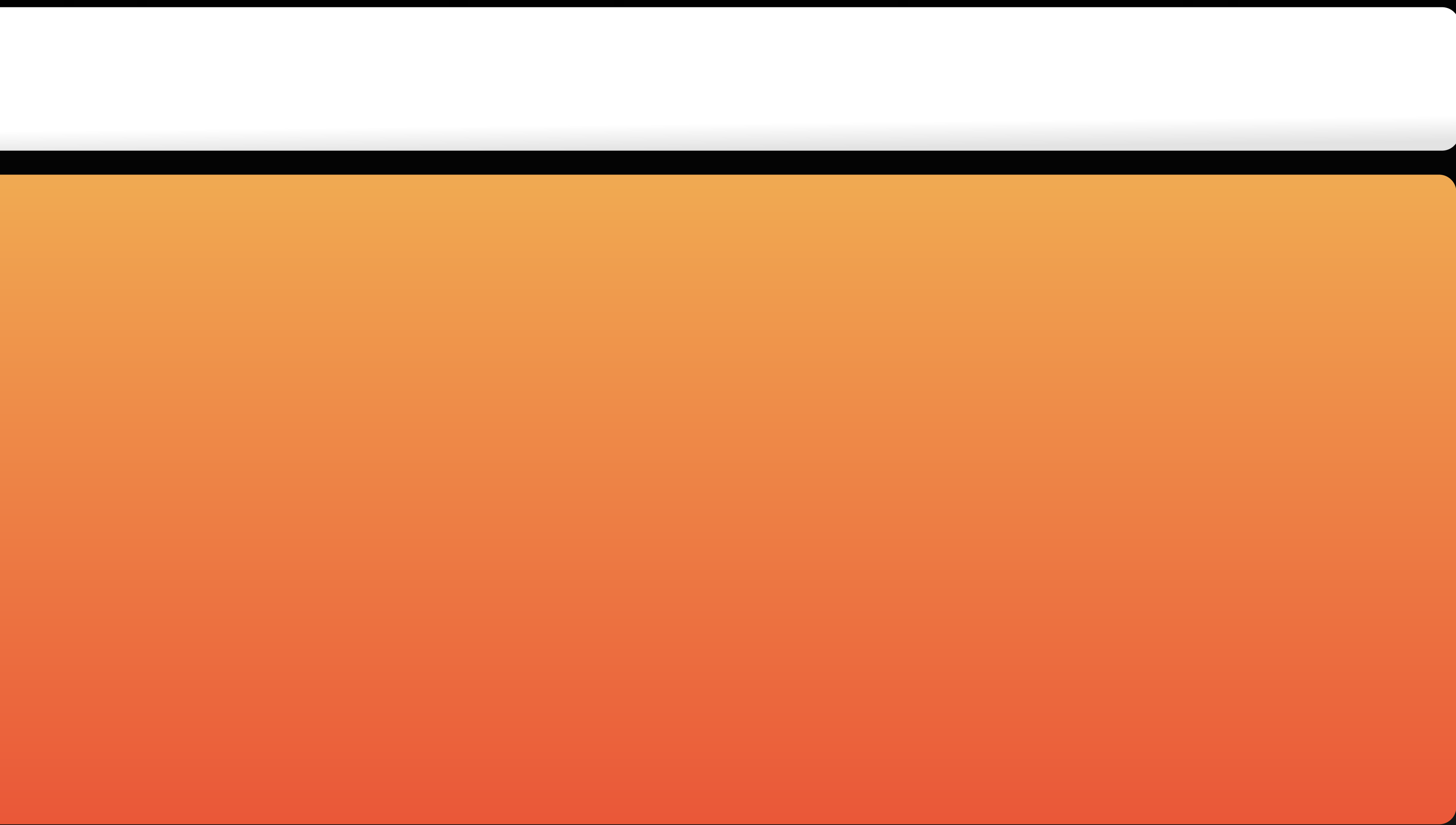
Up to

40%

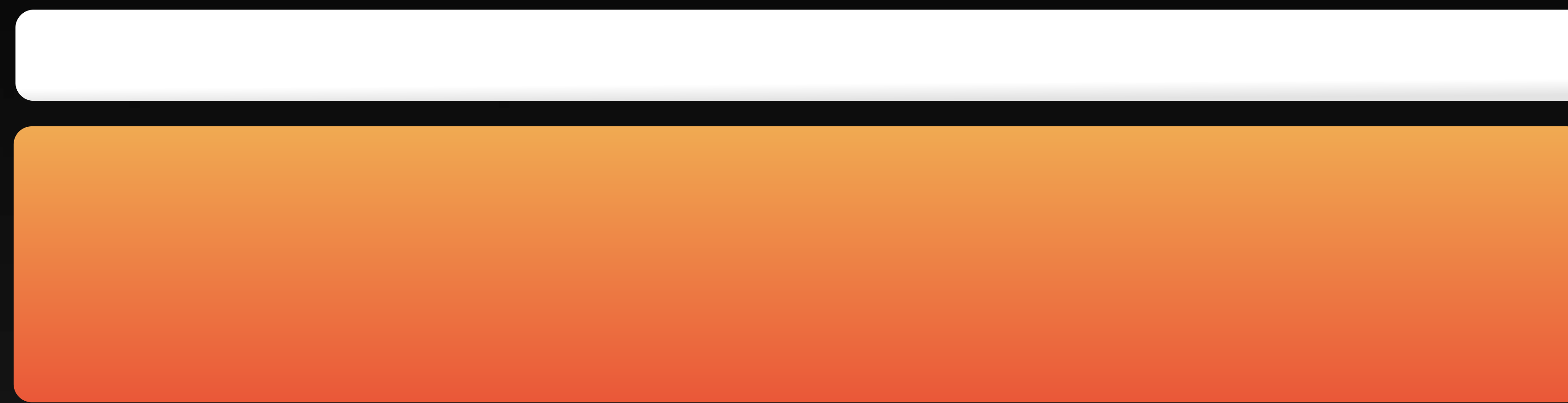
Faster file compilation



Incremental build



Xcode 8.3



Xcode 9



Preview in Xcode 9

Opt-in via Workspace Settings

Default build system soon



Source Control

2/3











of all pull request related activity on
GitHub happens from a Mac






GitHub

Integrated into Xcode

networking

GitHub	Last Updated	Owner
 Networking	Apr 4, 2017, 12:33 PM	3lvis
 DotNetty	May 17, 2017, 7:29 PM	Azure
 Networking	Nov 6, 2015, 5:23 AM	YouXianMing
 VolleyPlus	Mar 4, 2017, 10:58 PM	DWorkS
 libnetwork	Yesterday, 3:16 PM	docker
 networking	Mar 1, 2013, 10:38 AM	CoderDojoGitHub
 Awesome-Networking	Feb 21, 2017, 8:31 PM	clowwindy
 Alamofire	Yesterday, 11:35 PM	Alamofire
 handy	Apr 26, 2017, 9:50 AM	yedf
 gns3-gui	Yesterday, 6:52 AM	GNS3

3lvis/Networking
Easy HTTP Networking in Swift a NSURLSession wrapper with image caching support

Swift  76  1007  [README](#)

Done Clone

networking

GitHub

- Net
- Dot
- Net
- Voll
- libn
- netv
- Awe
- Alar
- han
- gns

3lvis/
Easy HT
Swift

NETWORKING

pod v3.0.3 Carthage compatible compatible swift 3.0
platform ios | osx | watchos | tvos license MIT chat on gitter

Networking was born out of the necessity of having a simple networking library that doesn't have crazy programming abstractions or uses the latest reactive programming techniques, but just a plain, simple and convenient wrapper around `NSURLSession` that supports common needs such as faking requests and caching images out of the box. A library that is small enough to read in one go but useful enough to include in any project. That's how **Networking** came to life, a fully tested library for iOS, tvOS, watchOS and OS X that will always be there for you.

- Super friendly API

Close



▼ Science

▼ Branches

WorkInProgress (current)

master ↓9

▼ Tags

Beta-1

Beta-2

WWDC

▼ Remotes

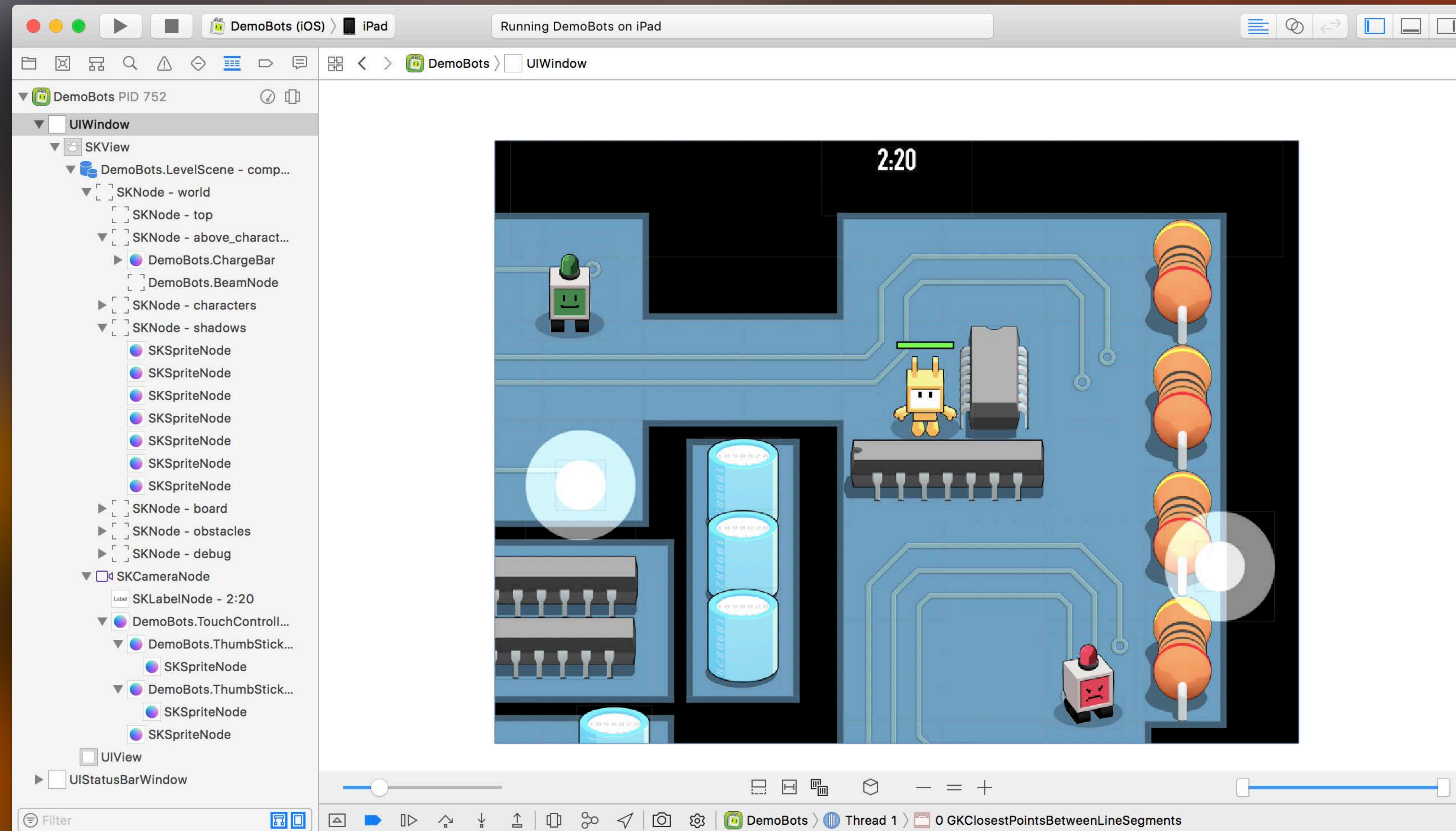
▶ origin

Demo

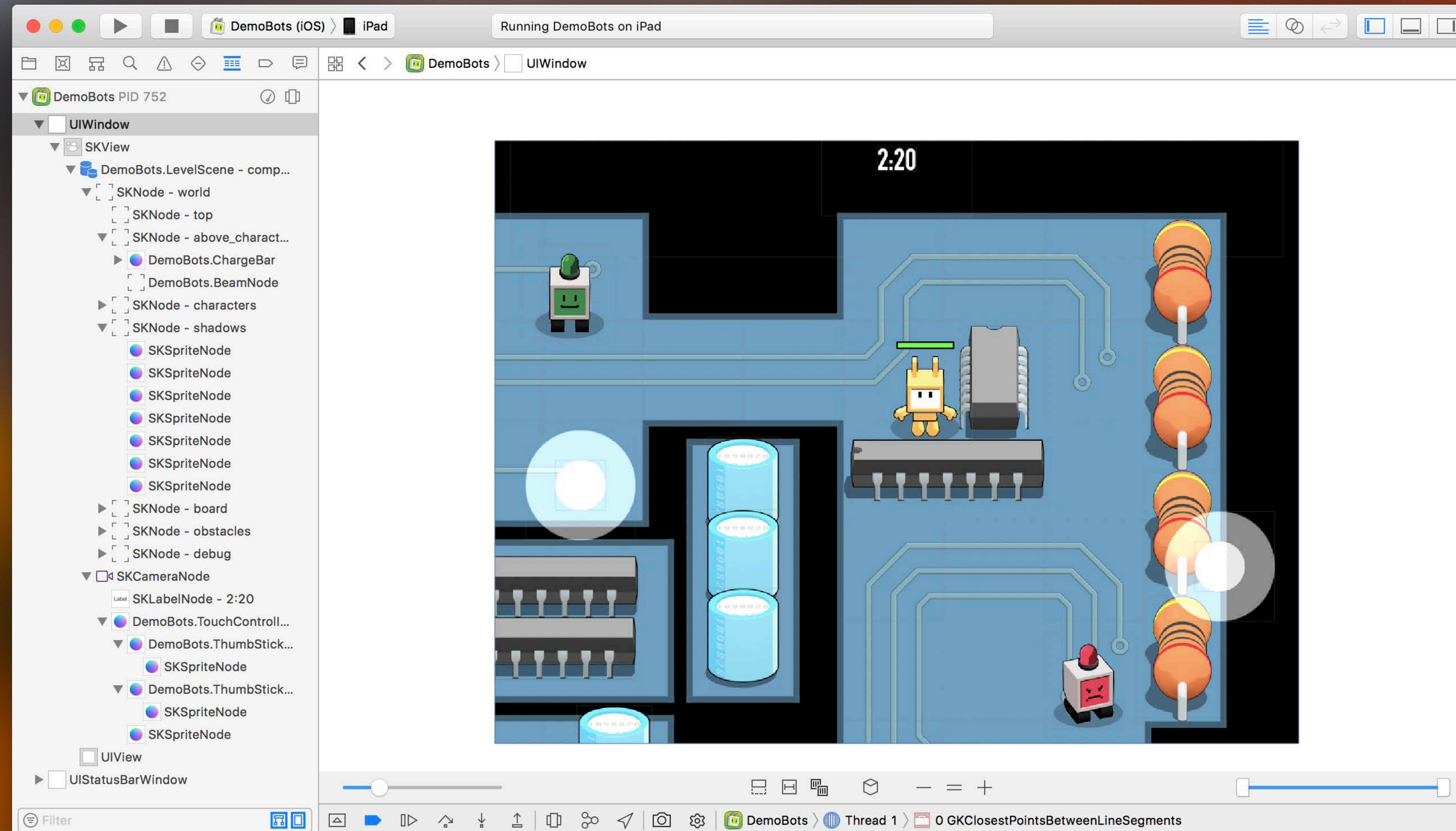


Debugging and Analysis

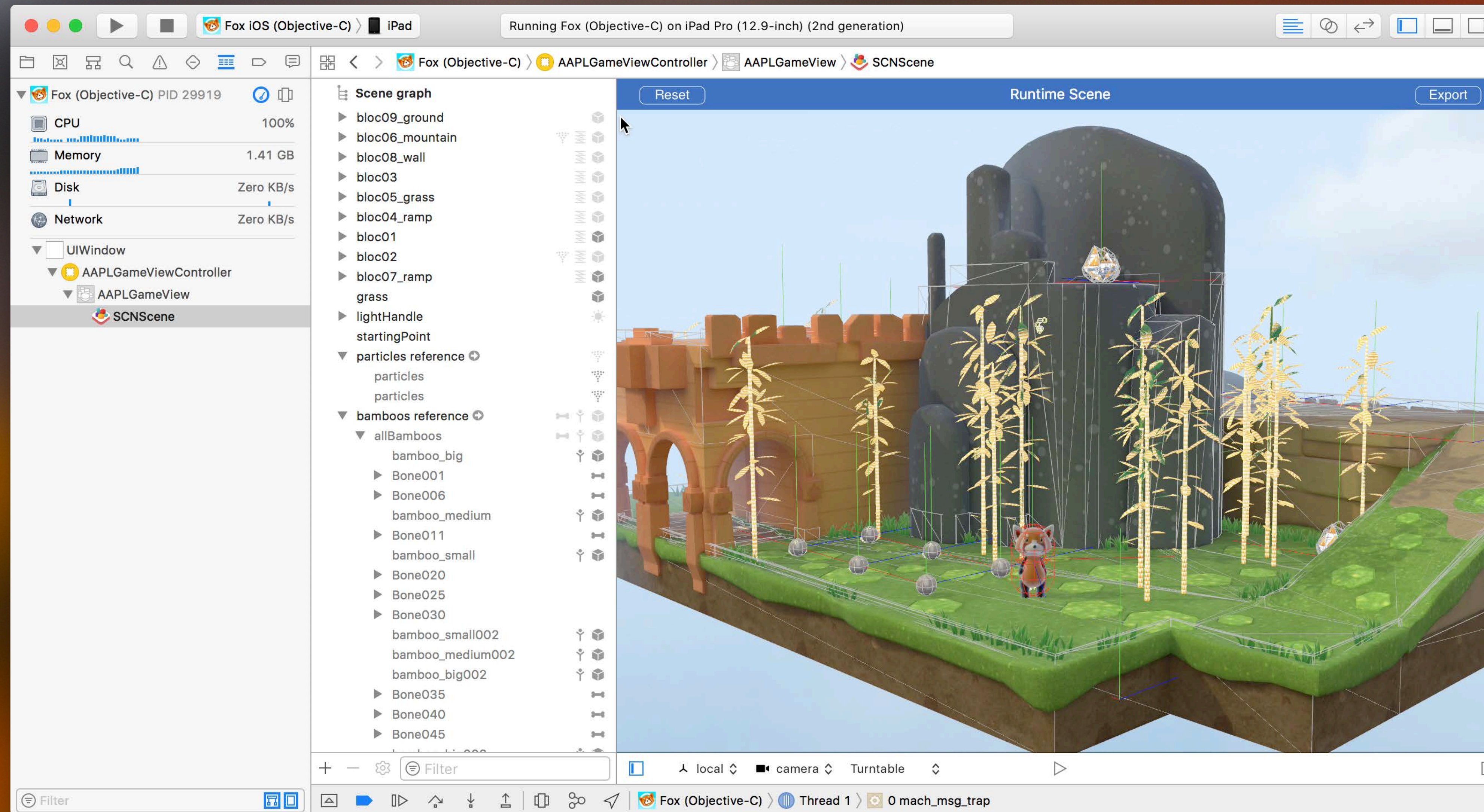
SpriteKit Scenes in View Debugger



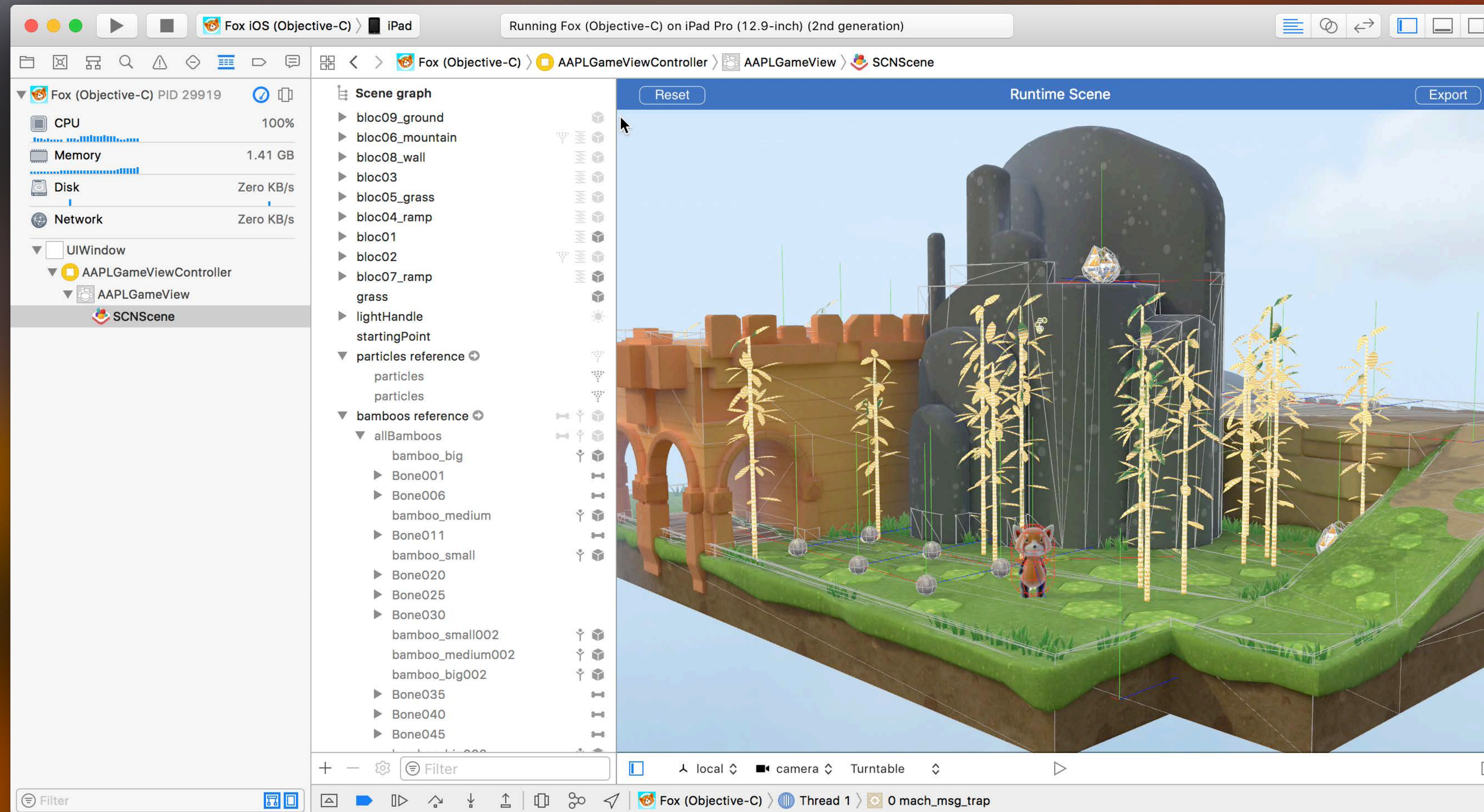
SpriteKit Scenes in View Debugger



SceneKit Scenes in View Debugger



SceneKit Scenes in View Debugger





Runtime Sanitizers



Thread Sanitizer

Address Sanitizer



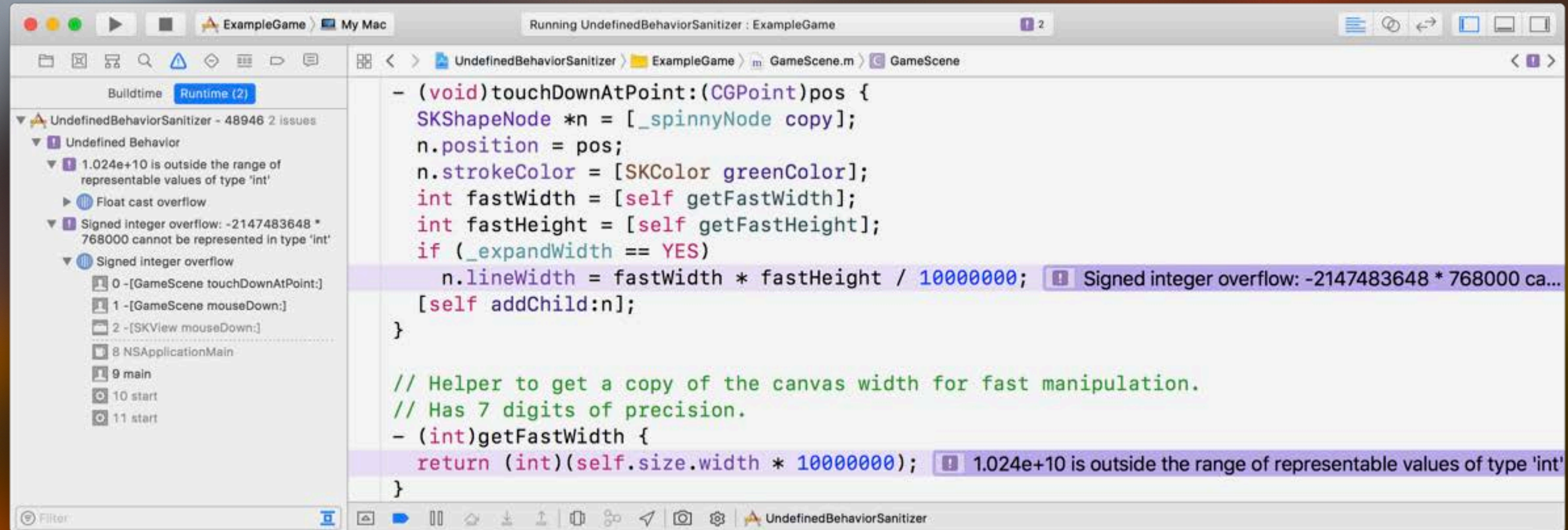
Thread Sanitizer

Address Sanitizer

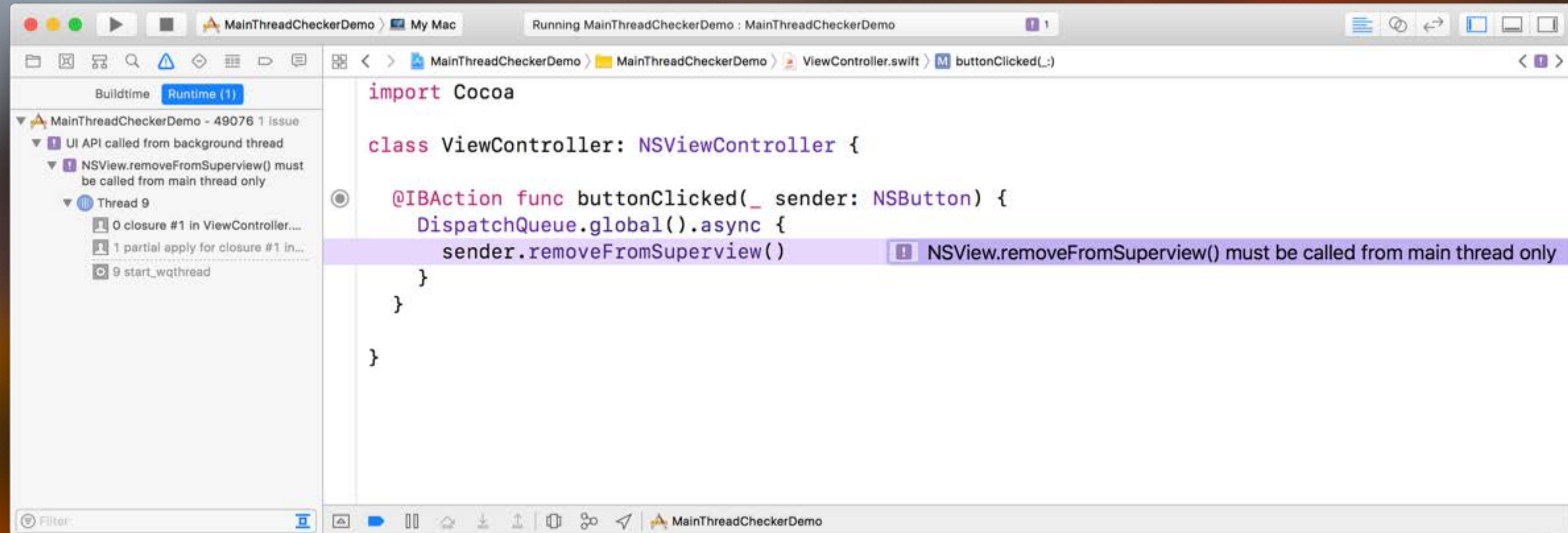
Undefined Behavior Sanitizer

Main Thread API Checker

Undefined Behavior Sanitizer



Main Thread API Checker



Build 3 targets

Run Debug

Test Debug

Profile Release

Analyze Debug

Archive Release

Info Arguments Options **Diagnostics**

Runtime Sanitization Requires recompilation

- Address Sanitizer
 - Detect use of stack after return
- Thread Sanitizer
 - Pause on issues
- Undefined Behavior Sanitizer
 - Pause on issues

Runtime API Checking

- Main Thread Checker
 - Pause on issues

Memory Management

- Malloc Scribble
- Malloc Guard Edges
- Guard Malloc
- Zombie Objects

Logging

- Malloc Stack
 - All Allocation and Free History ▾
- Dynamic Linker API Usage
- Dynamic Library Loads

Duplicate Scheme

Manage Schemes...

Shared

Close

Build 3 targets | Info | Arguments | Options | Diagnostics

Run Debug | Runtime Sanitization | Address Sanitizer

Runtime Sanitization Requires recompilation

- Address Sanitizer
 - Detect use of stack after return
- Thread Sanitizer
 - Pause on issues
- Undefined Behavior Sanitizer
 - Pause on issues

Runtime API Checking

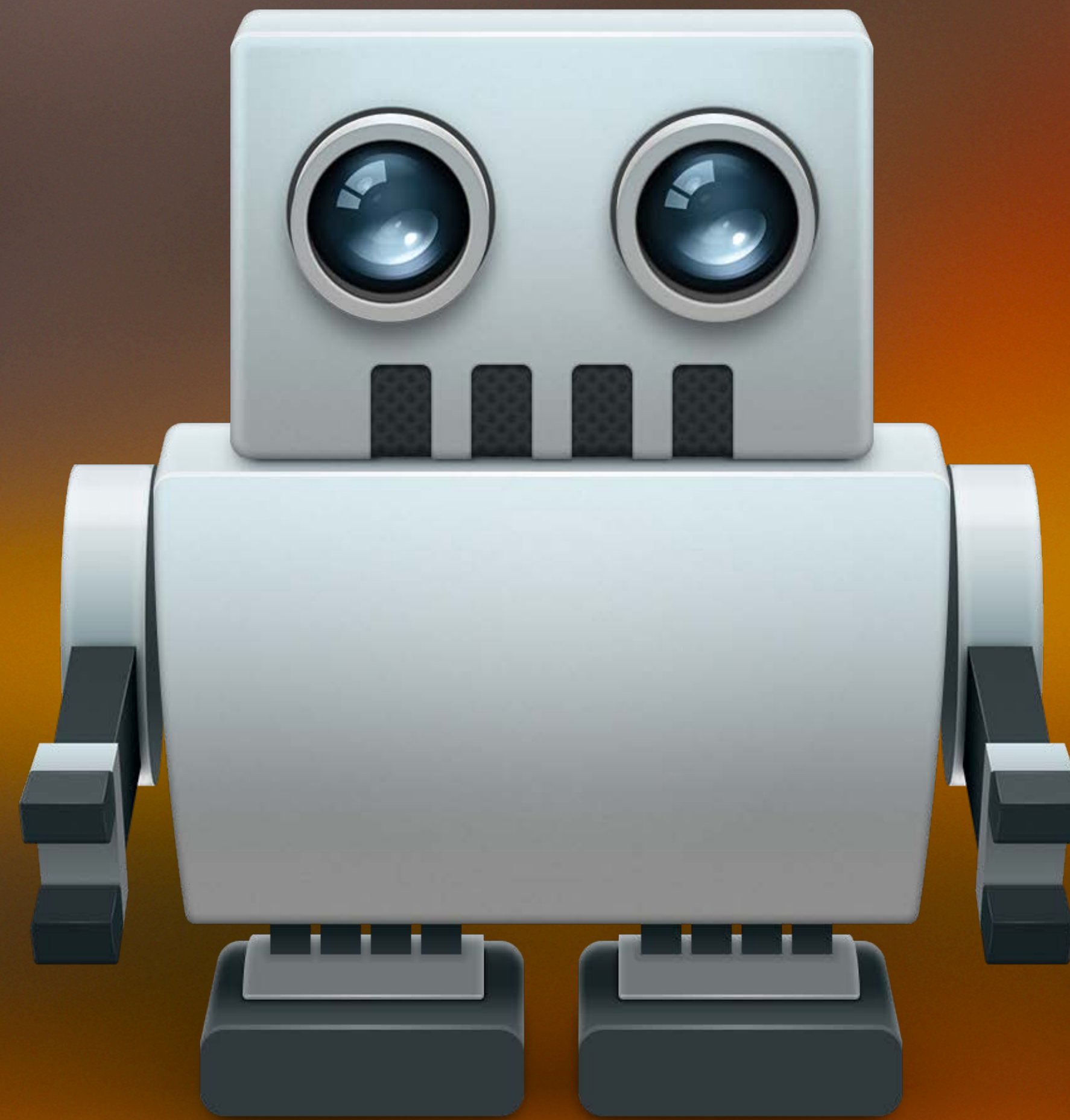
- Main Thread Checker
 - Pause on issues

Duplicate Scheme | Manage Schemes... | Shared | Close

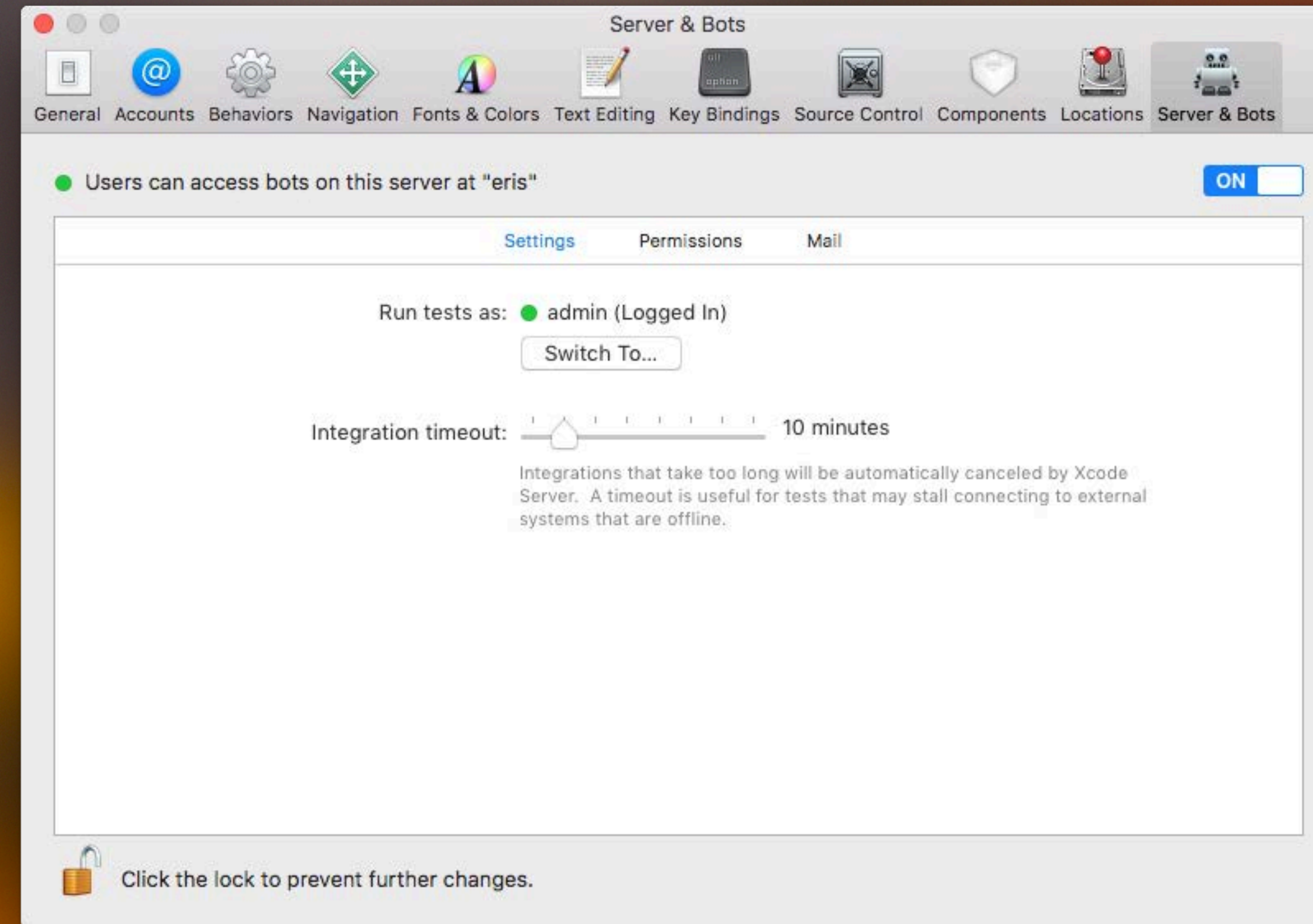


Testing and Continuous Integration

Xcode Server Built In



Xcode Server Built In



UI Testing Multiple Applications



UI Testing Multiple Applications





Query performance



Query performance



Using first match API

Parallel Device and Simulator Testing

```
> xcodebuild test  
  -destination "name=iPad Pro"  
  -destination "name=iPhone7"■
```

Parallel Device and Simulator Testing

```
> xcodebuild test  
  -destination "name=iPad Pro"  
  -destination "name=iPhone7"■
```

Multiple Booted Simulator Devices



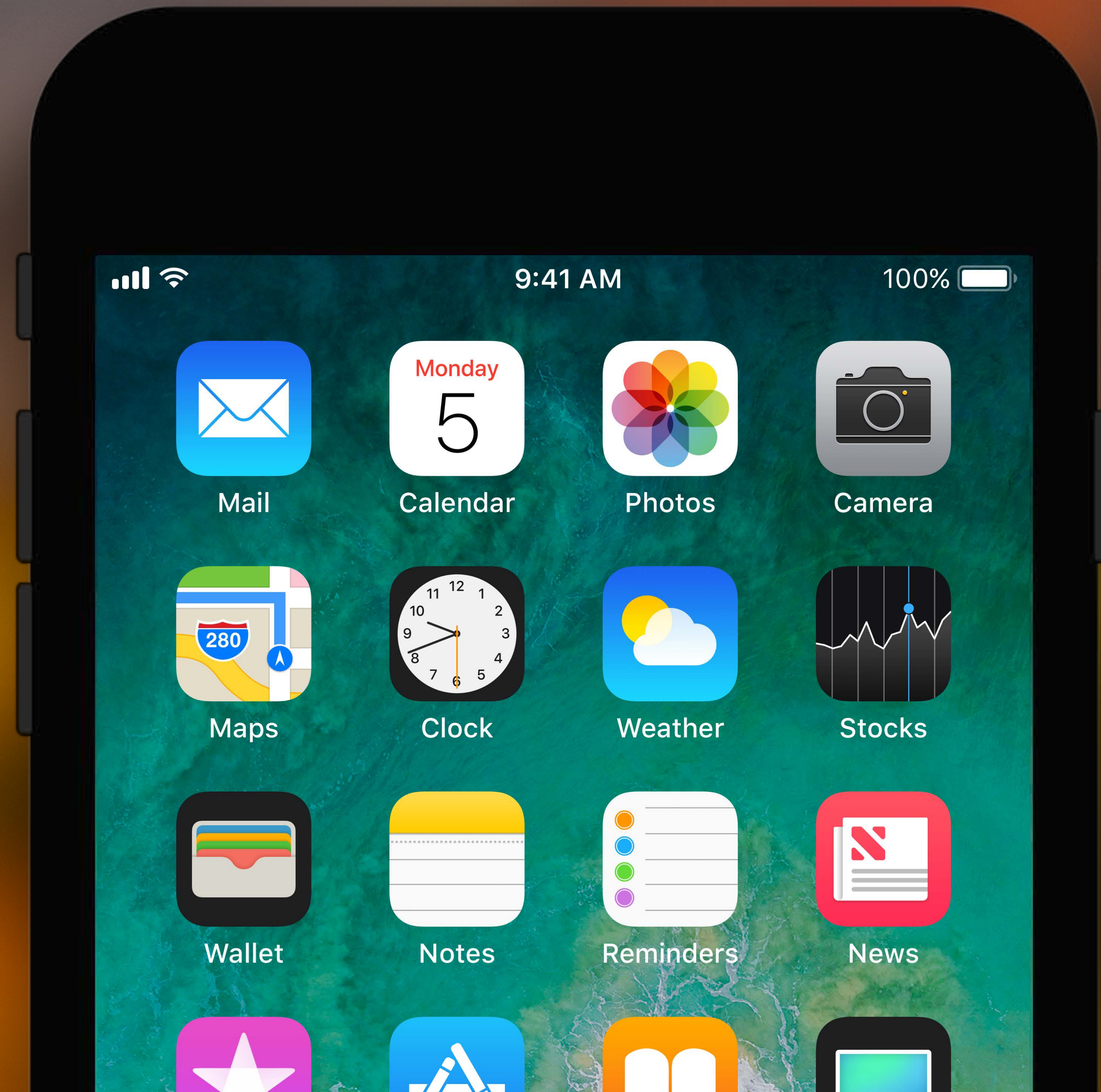
Simulator bezel

Hardware controls

Support for edge swipes

Fully resizable

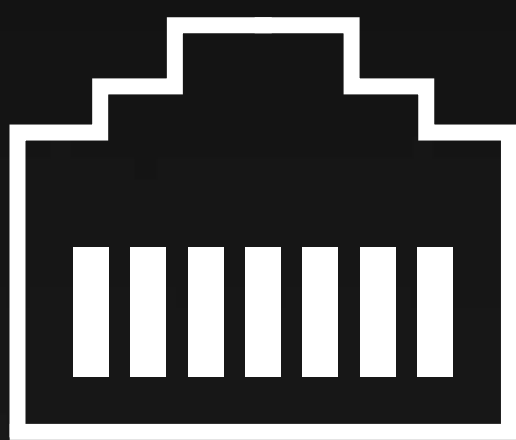
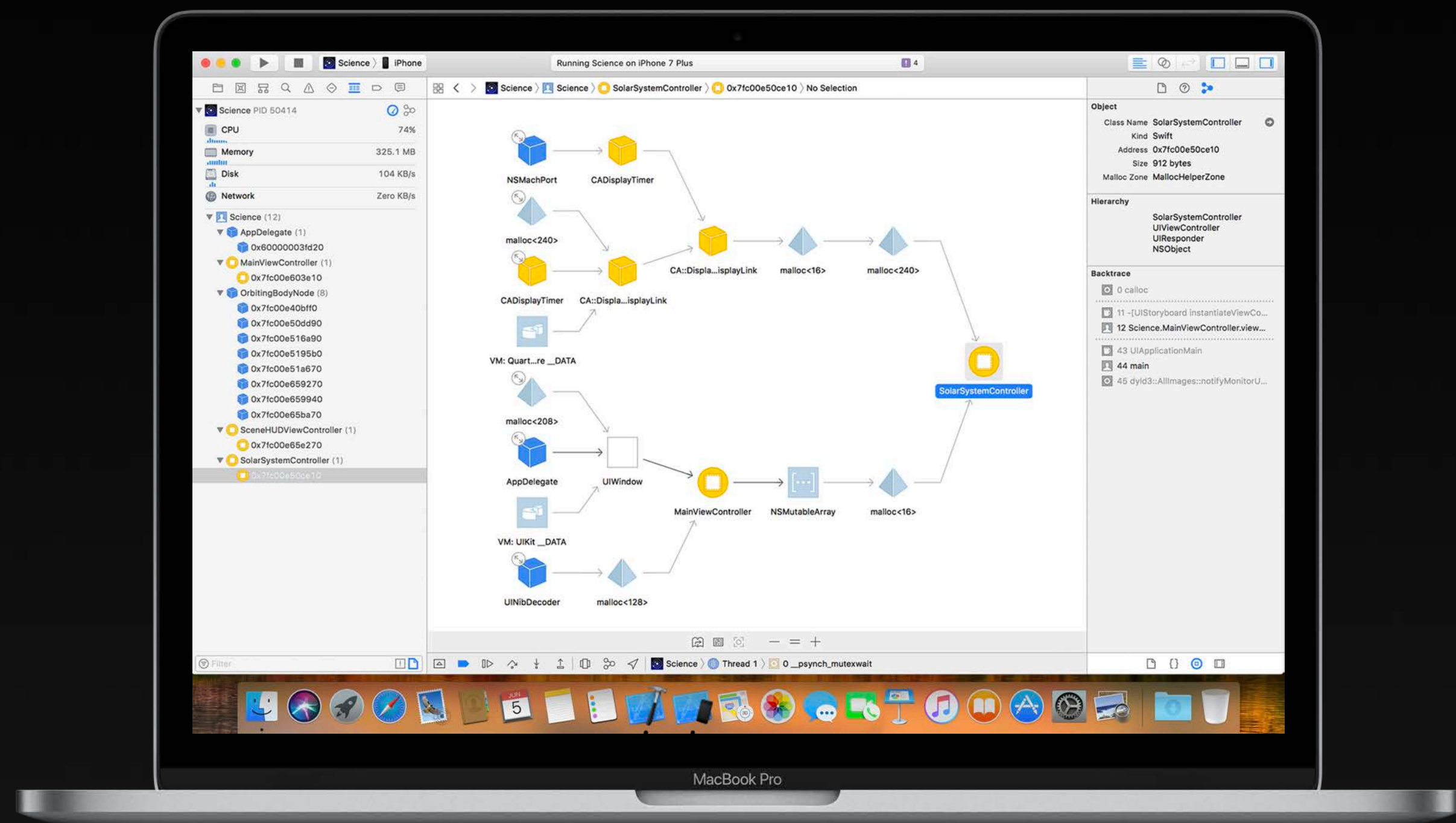
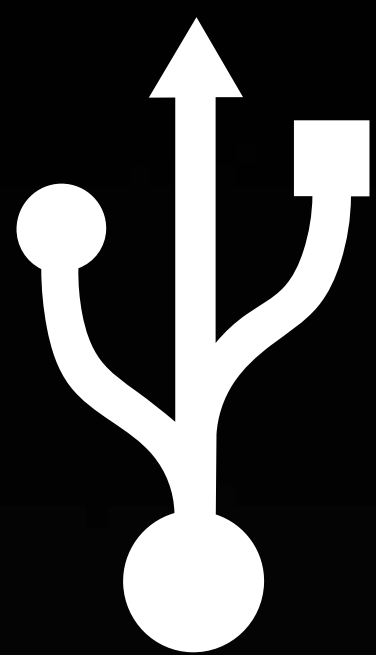
Touch Bar controls





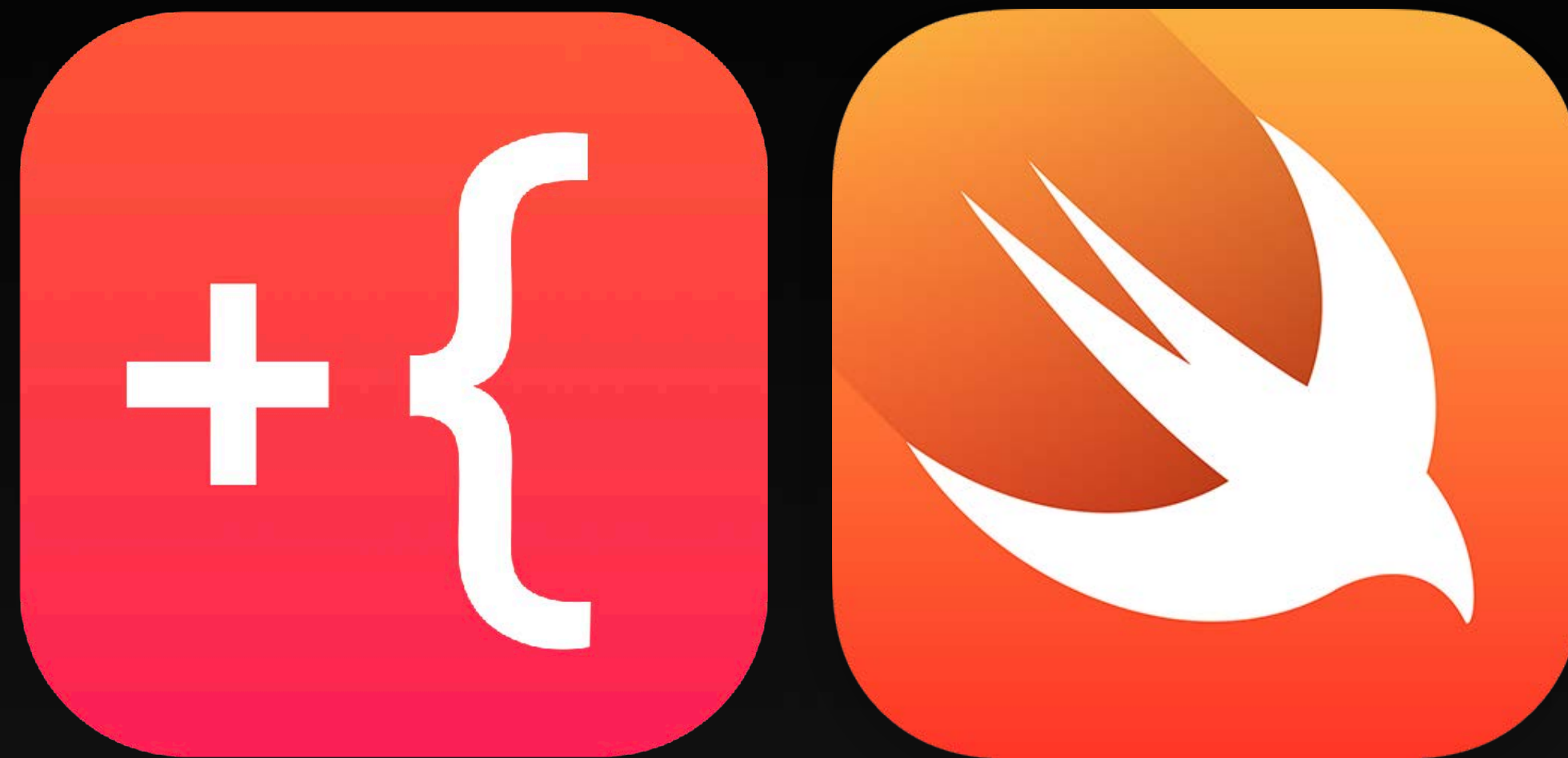


Wireless Development





WARNING
MAY 7:36

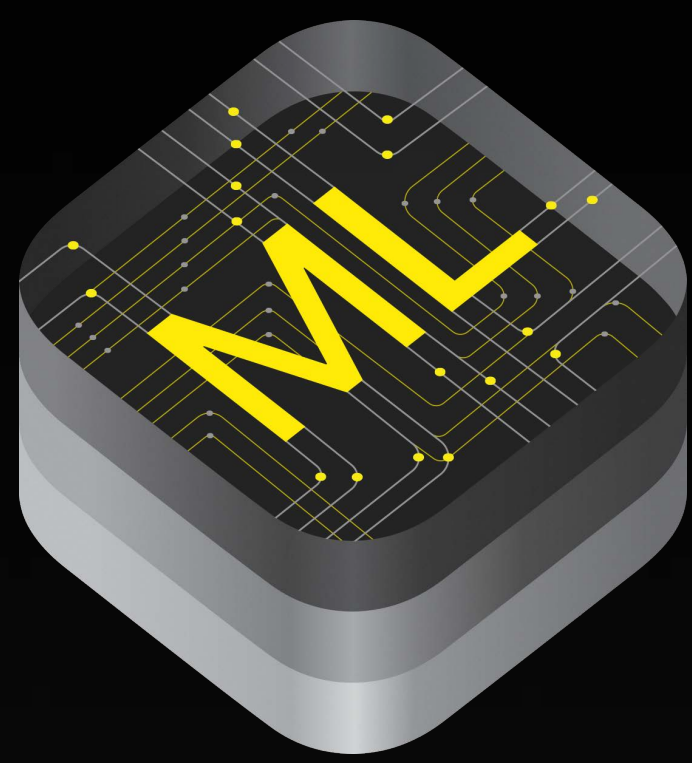














Drag and Drop



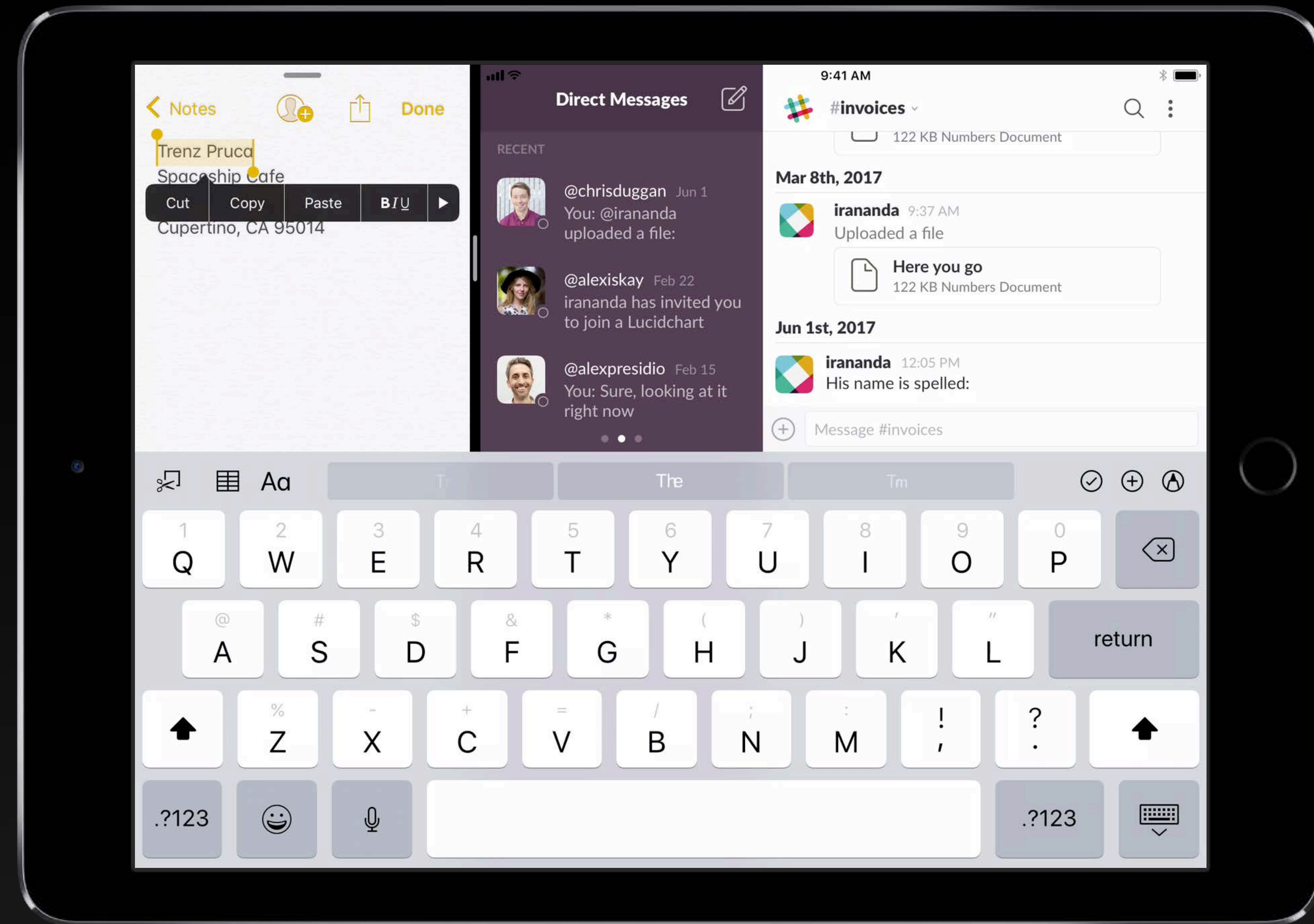
Easy to adopt

Flexible and customizable

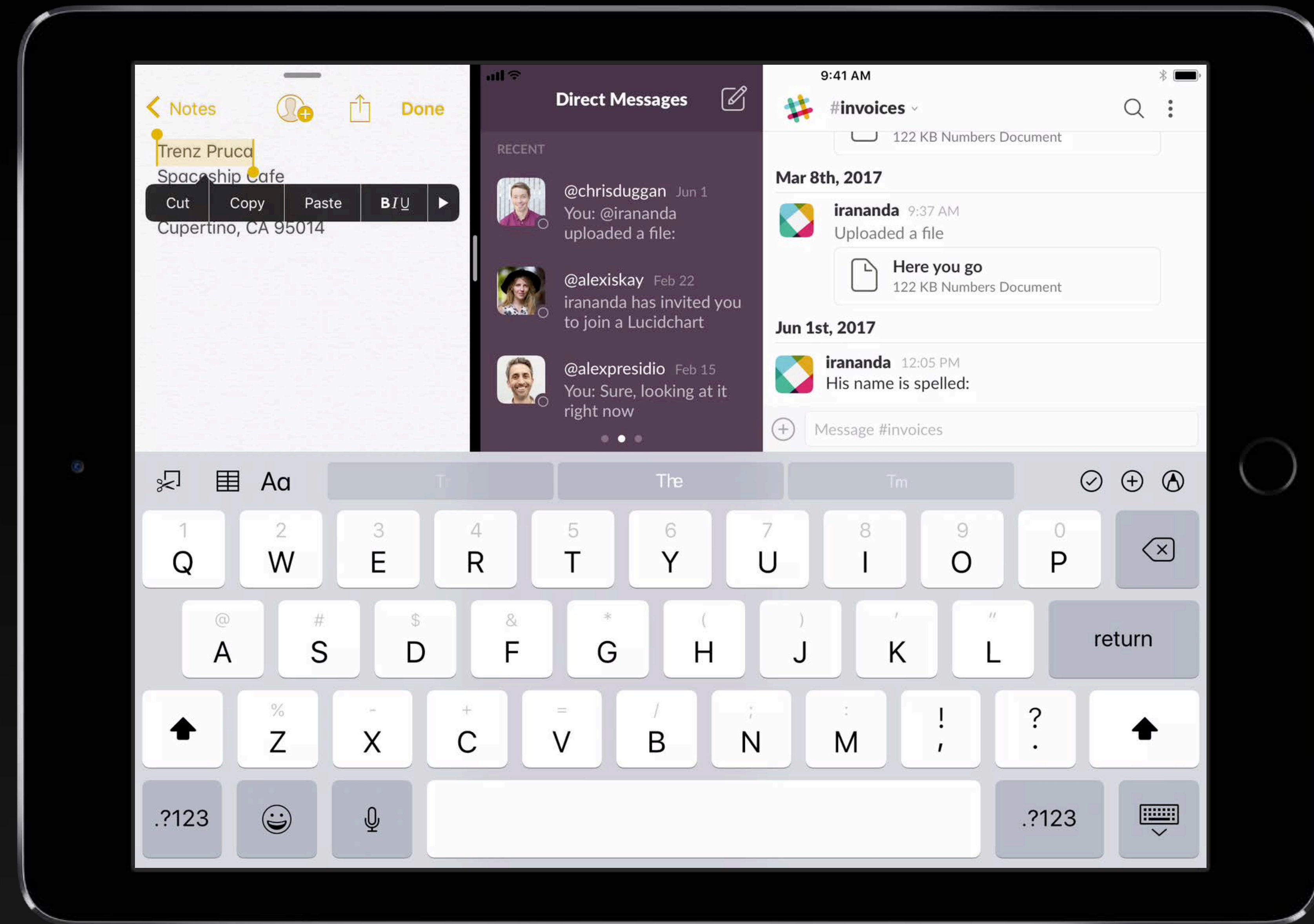
Fully multi-touch enabled

Secure by design

Automatic for Text and Web



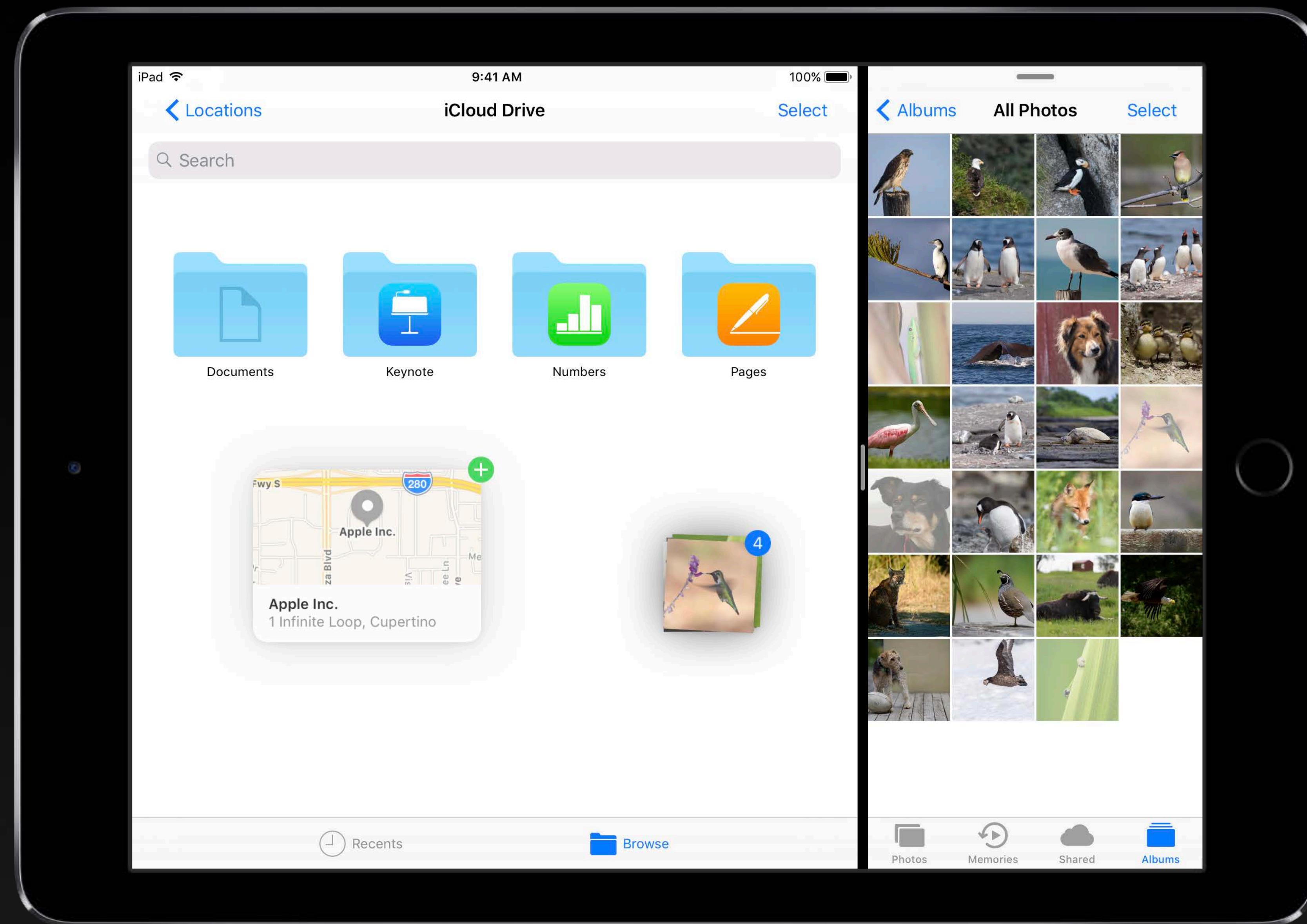
Automatic for Text and Web



```
// Begin Drag
let dragData = self.data(at: sourceIndexPath)
let itemProvider = NSItemProvider(object: dragData)
return [UIDragItem(itemProvider: itemProvider)]
```

```
// Begin Drag
let dragData = self.data(at: sourceIndexPath)
let itemProvider = NSItemProvider(object: dragData)
return [UIDragItem(itemProvider: itemProvider)]

// Perform Drop
coordinator.session.loadObjects(ofClass: MyDataType.self)
{ (data) in
    self.insertData(data, at: destinationIndexPath)
    collectionView.reloadSections(IndexSet(integer: 0))
}
```



Gestures

Lift animations

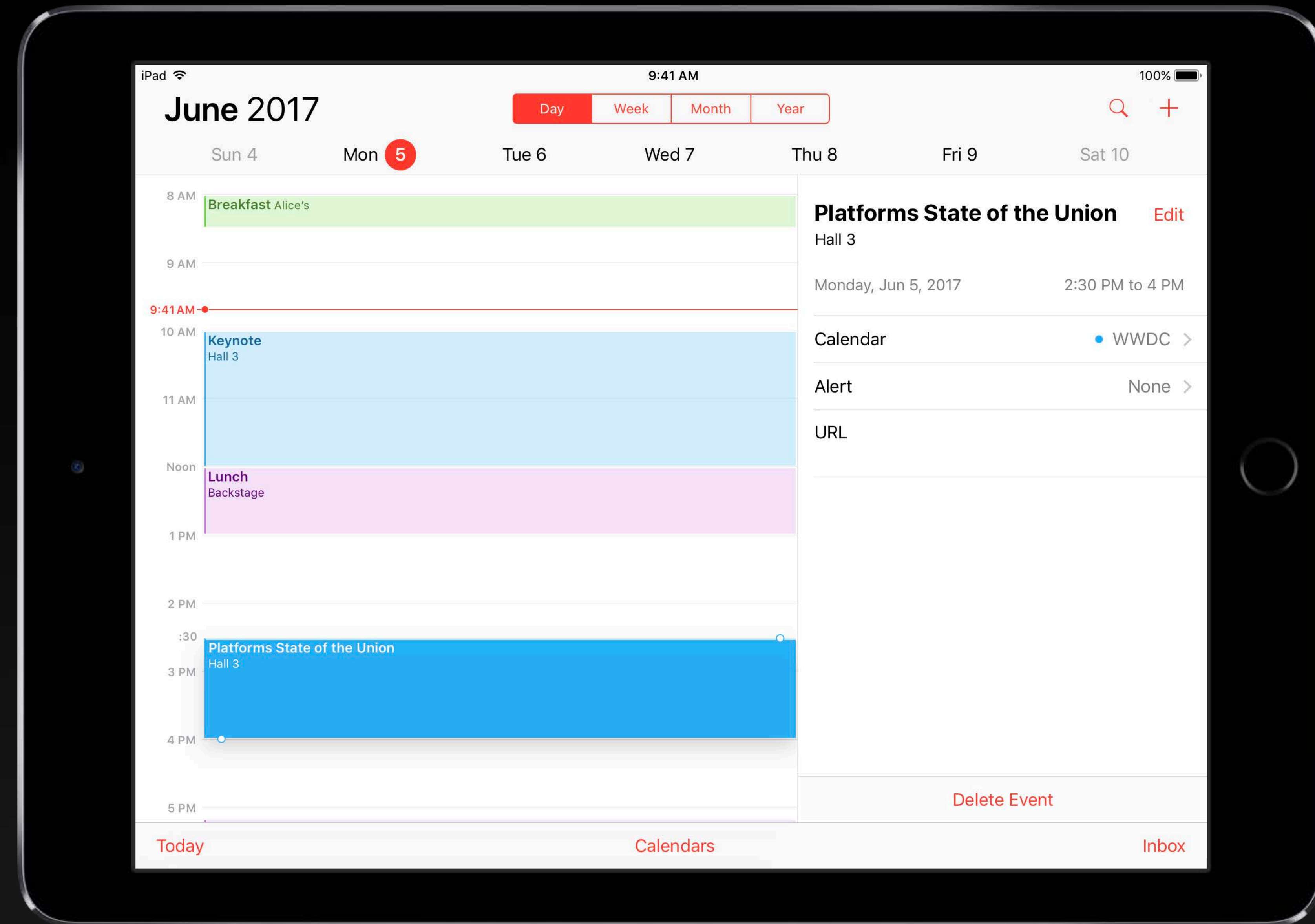
Multi-item drags

Previews

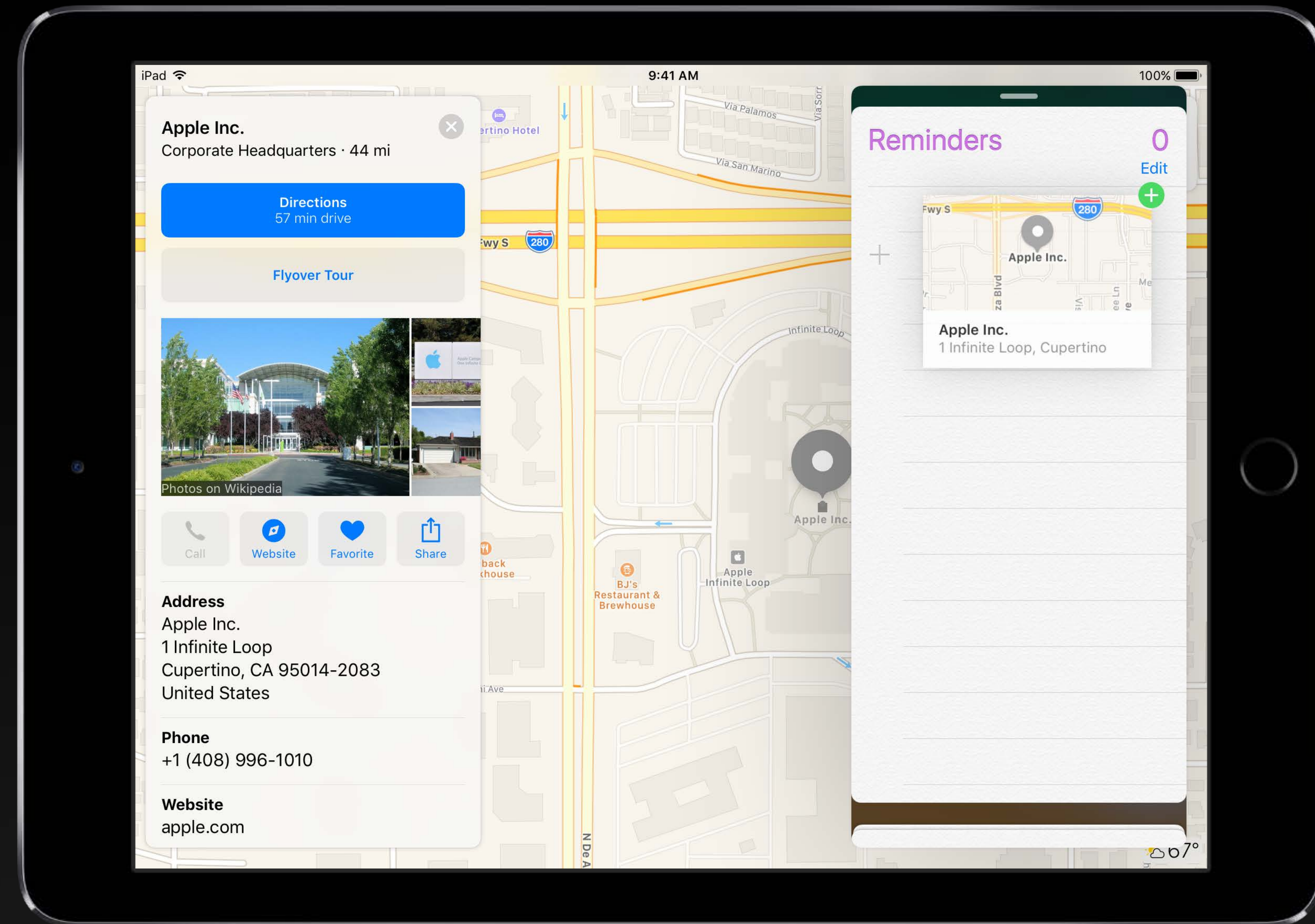
Badges

Set-down animations

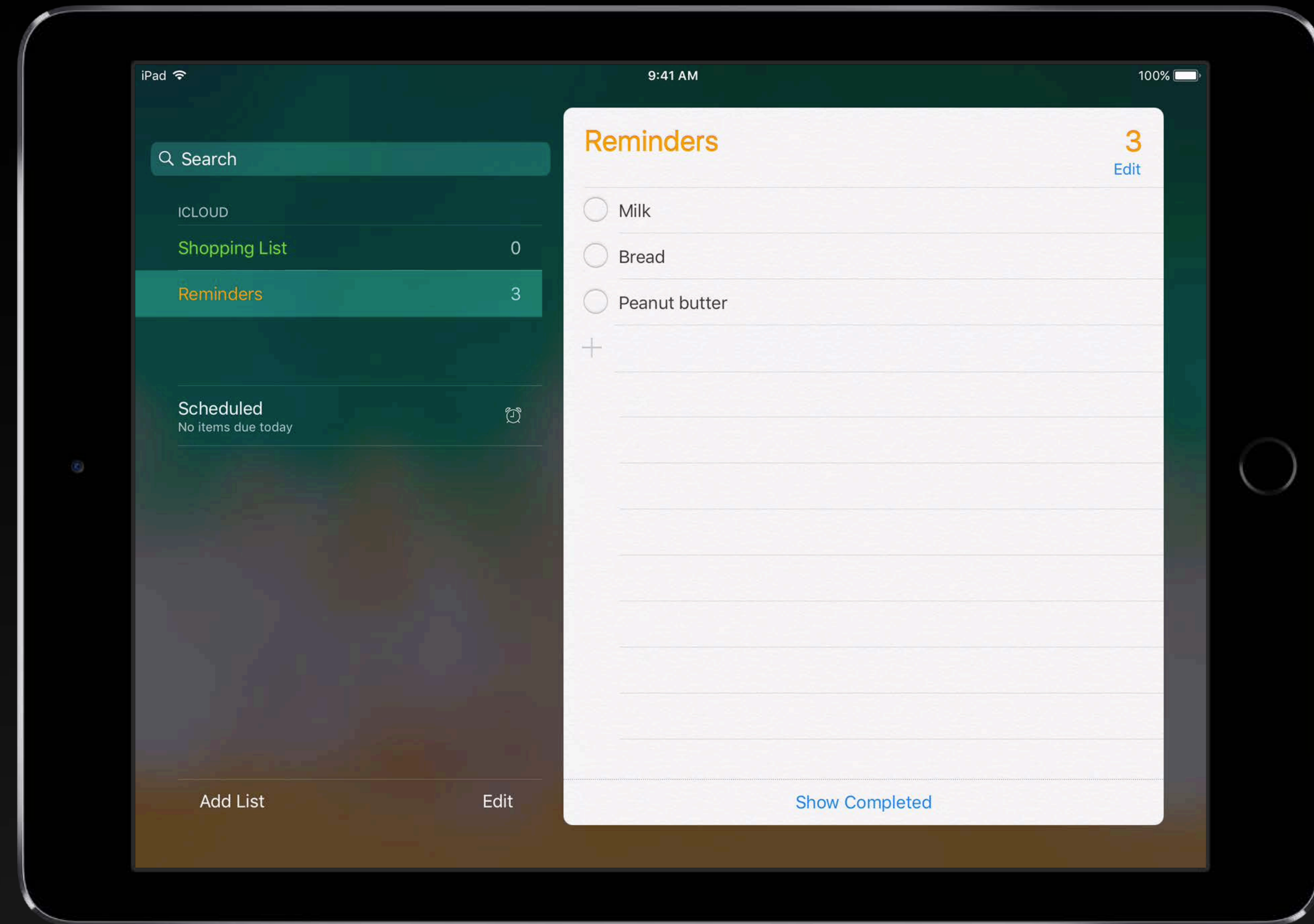
Customizable Appearance



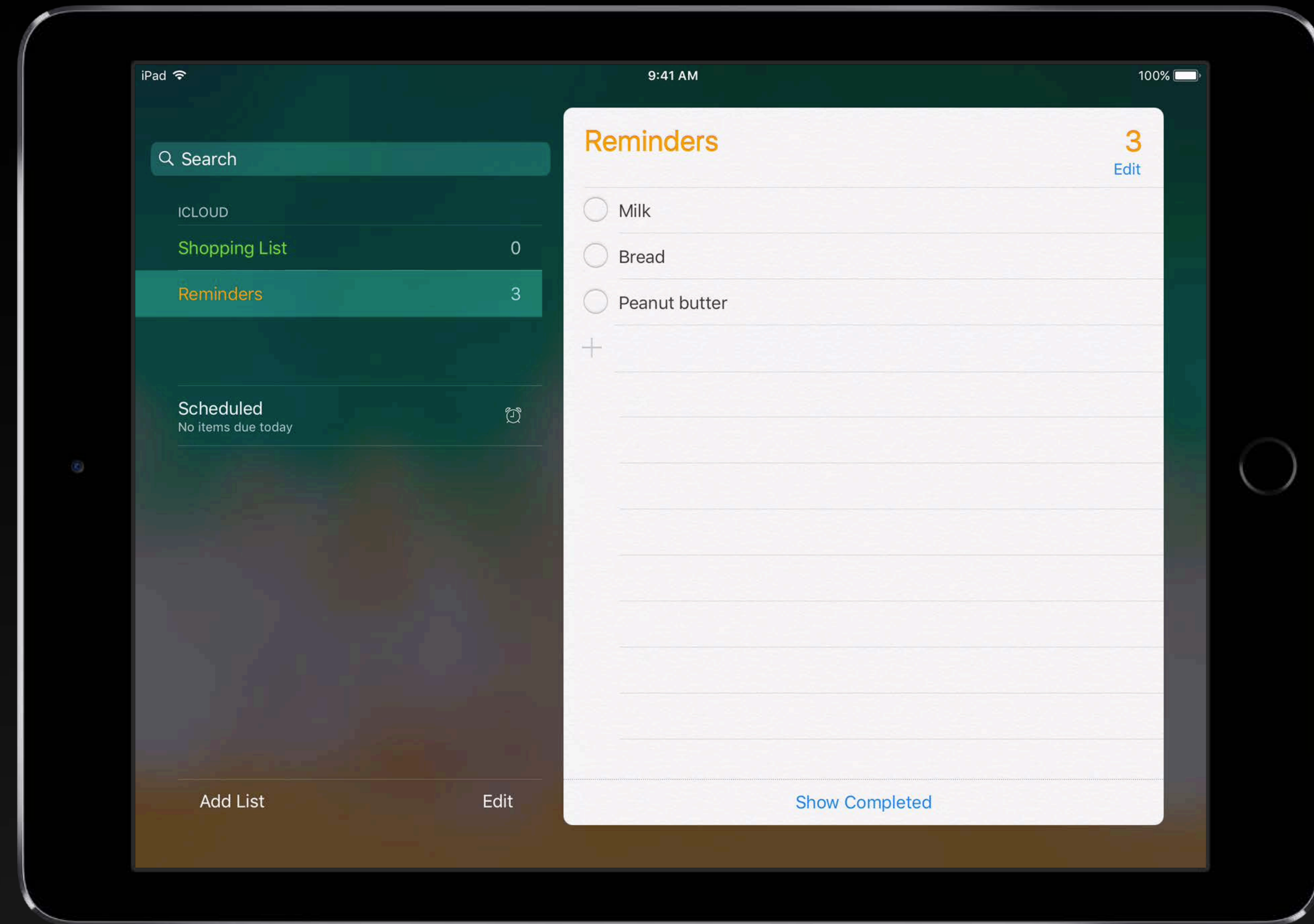
Custom Data Types



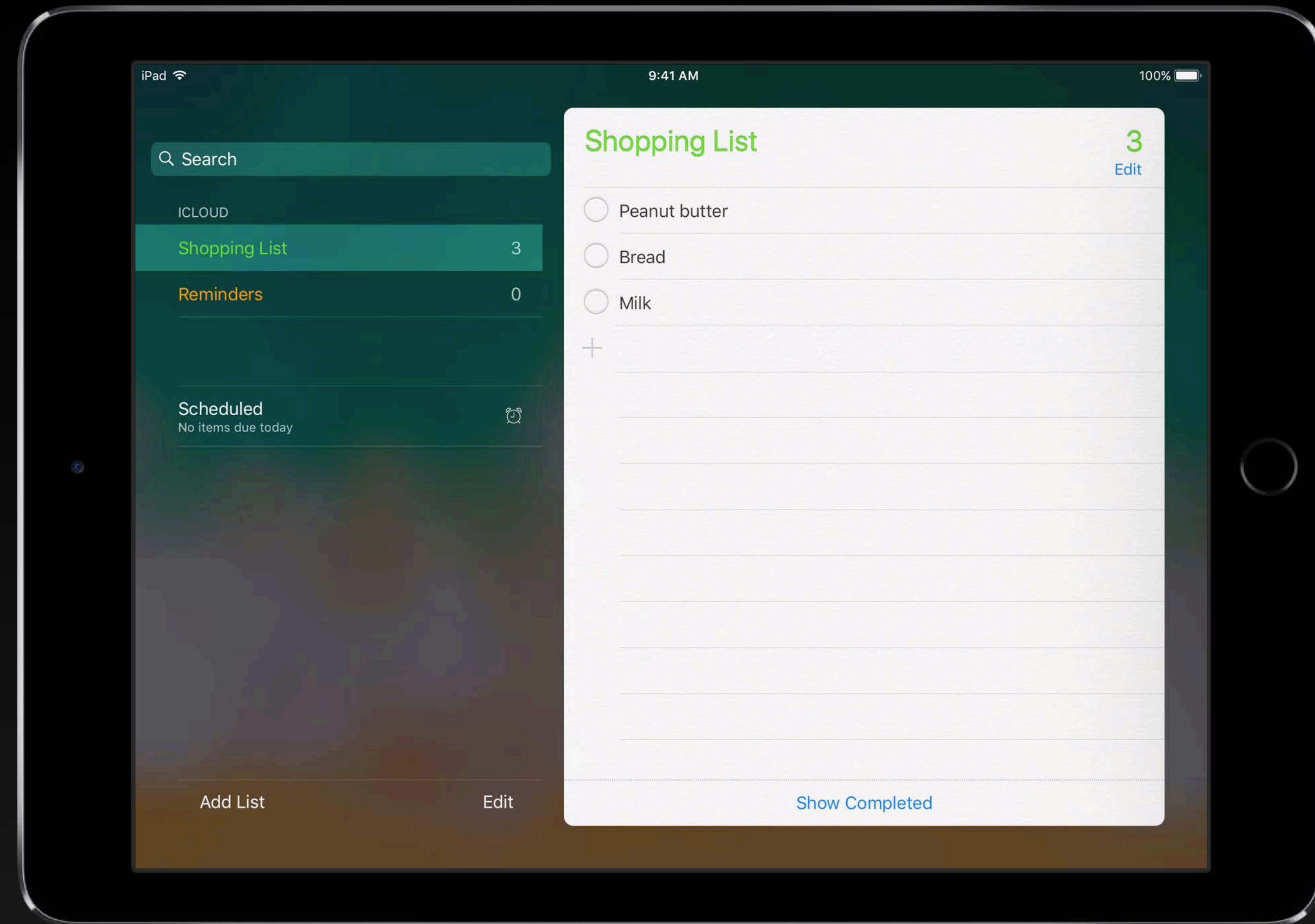
Reordering Support



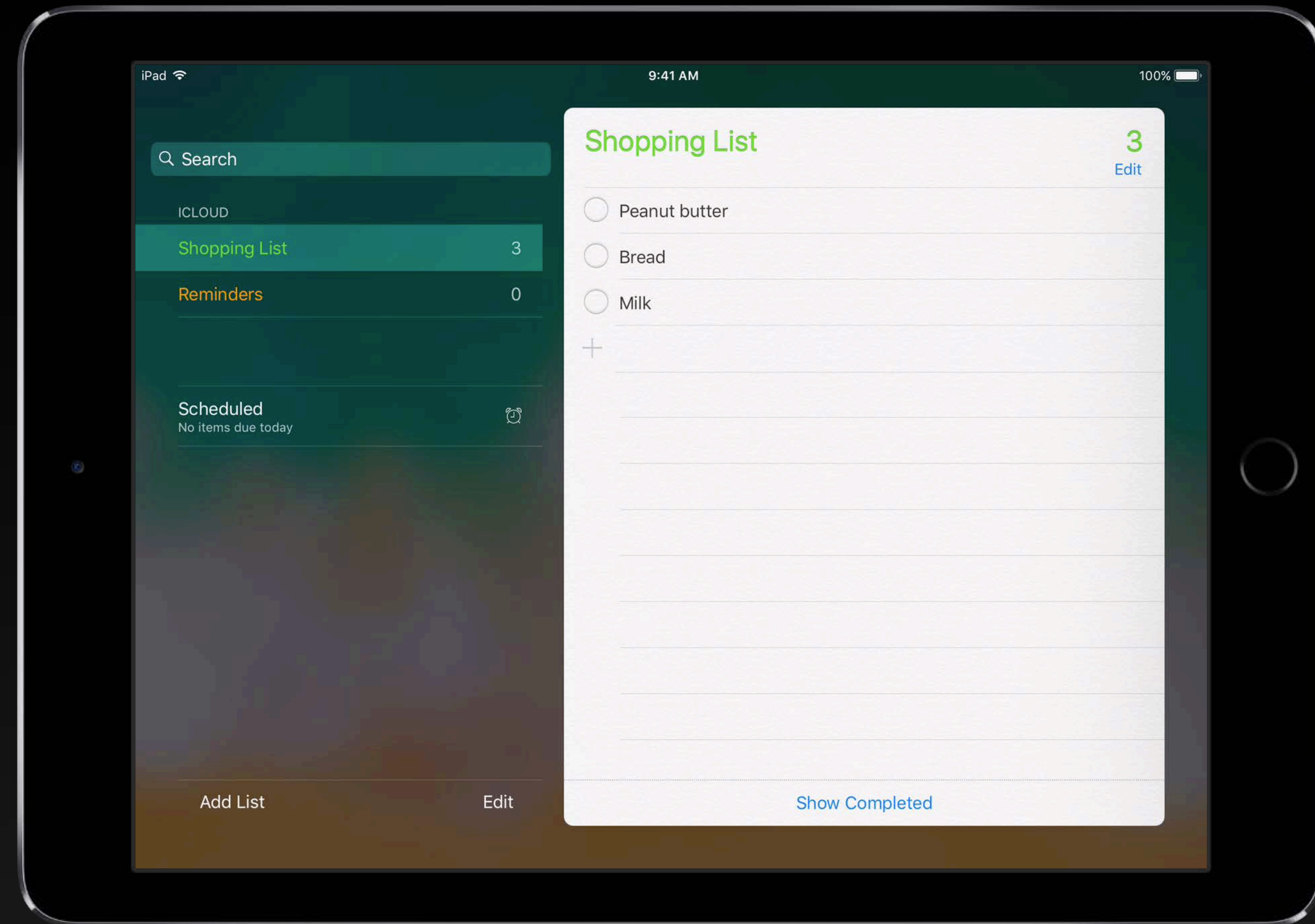
Reordering Support



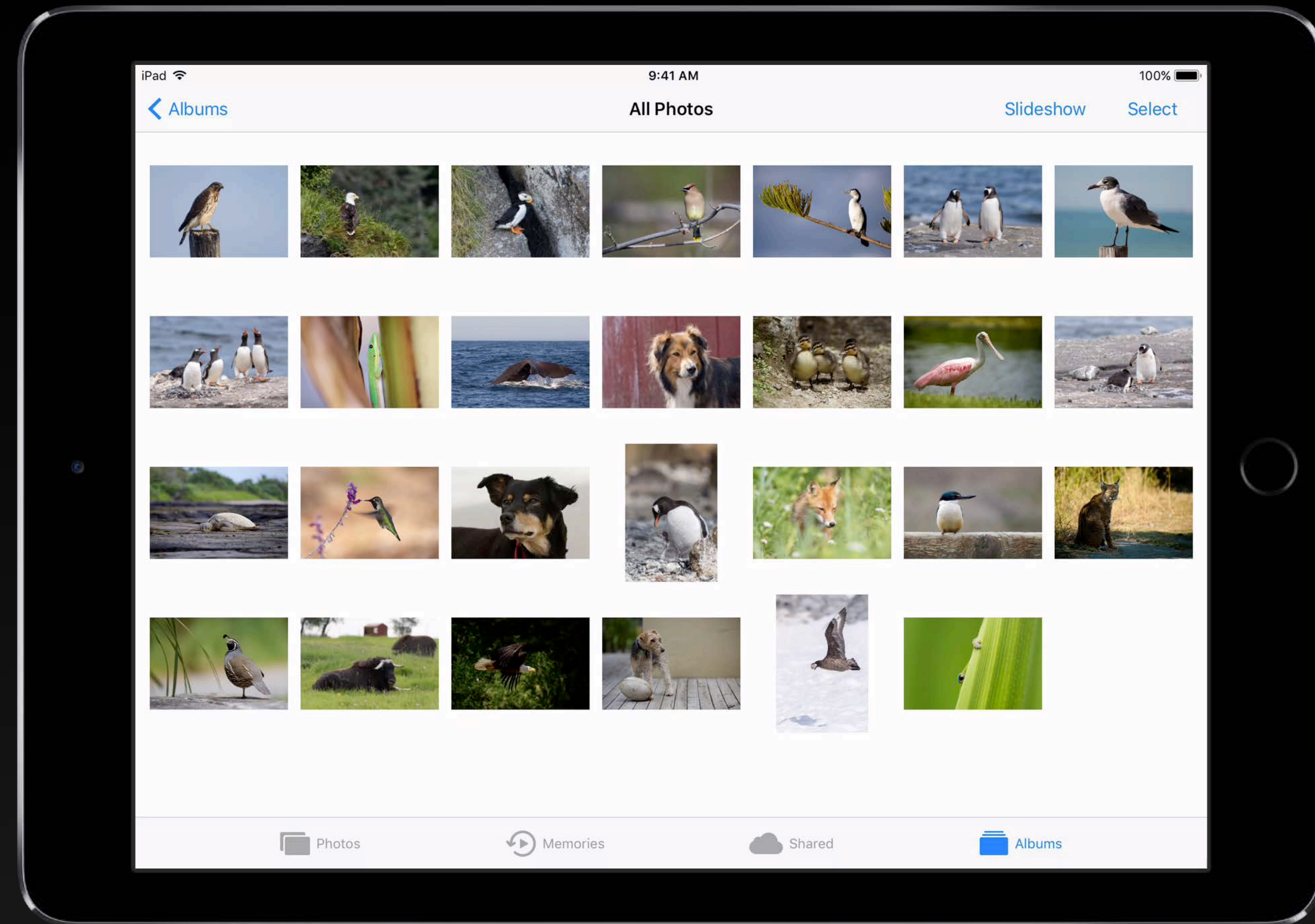
Cross-Process



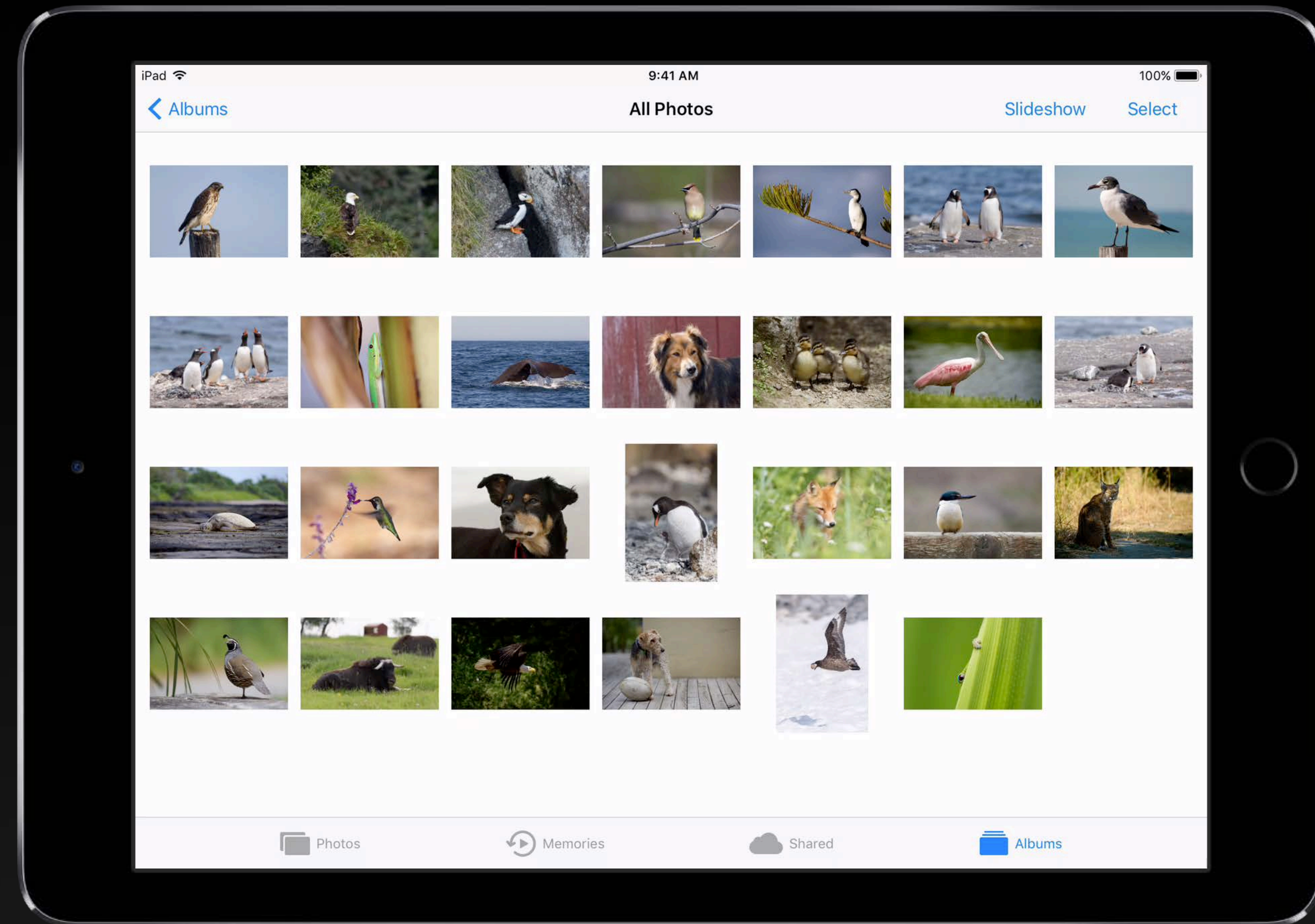
Cross-Process



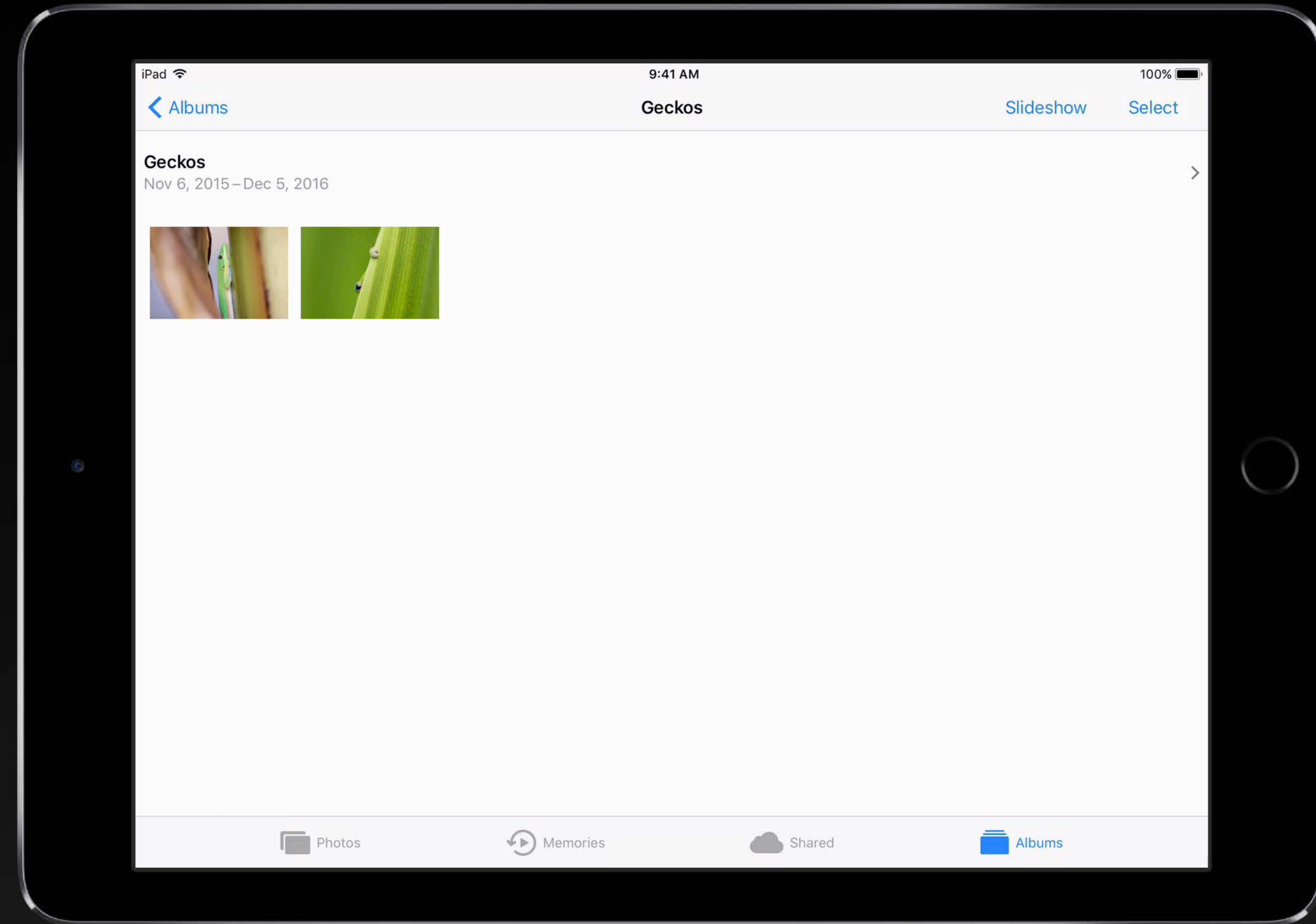
In-App Multi-Touch



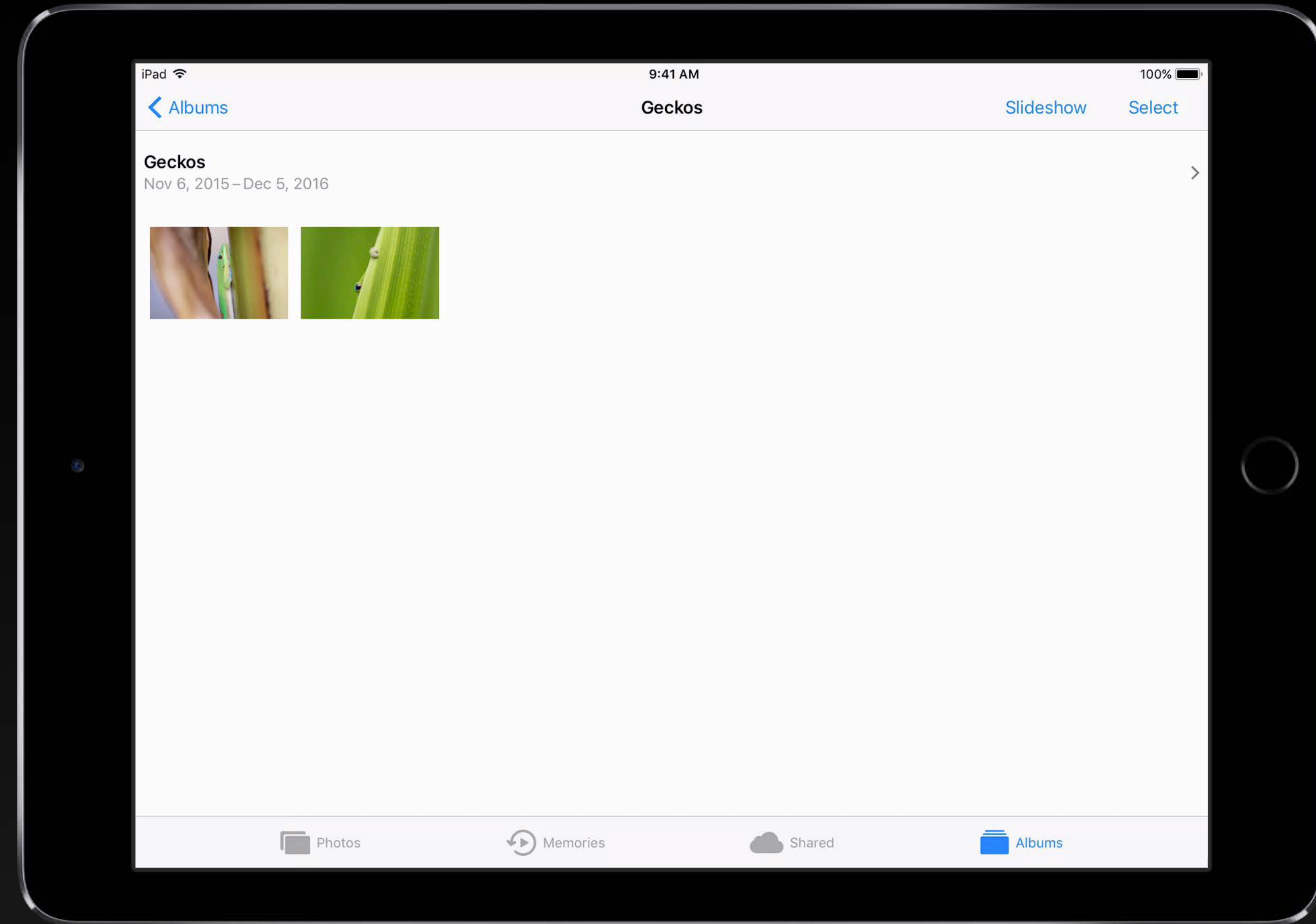
In-App Multi-Touch



System-Wide Multi-Touch



System-Wide Multi-Touch





Secure

Demo



Adobe

Brooke Francesi



9:41 AM

100%

Edit



Messages

Search

Megan 8:30 AM >
Hope you have a great flight! See you tonight!

Gram Yesterday >
Thinking about you a lot this week


Samantha Saturday >
How's the new place in San Francisco?

Mom Friday >
Looking forward to seeing you soon!

9:41 AM

Settings

Settings

 **Josh Shaffer**
Apple ID, iCloud, iTunes

- Airplane Mode
- Wi-Fi
- Bluetooth
- Cellular
- VPN
- Carrier
- Notifications

9:41 AM

Mailboxes

- Inbox
- VIP
- Unread
- Attachments
- Today
- Trash

Updated Just

9:41 AM

Folders

iCLOUD

- All iCloud
- Notes
- Drawings
- Home
- Presents
- Work


9:41 AM

Games



NEW GAME

Injustice 2

When iconic superh



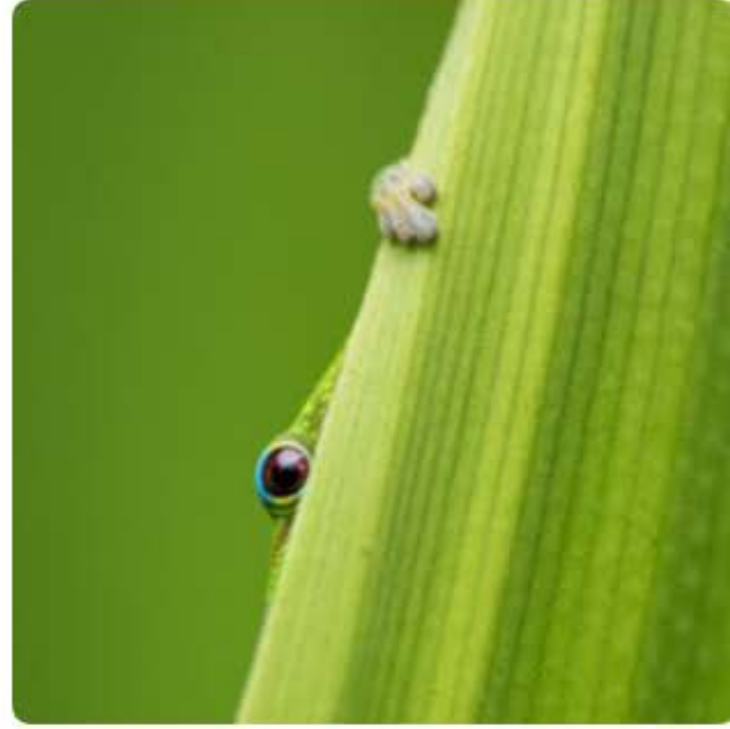
New Games We Lo

-  **Zombie Gunshi Survival**
Tour the apocalyps
-  **Vignettes**


Today Games Apps

9:41 AM

Albums



All Photos
33



Places
1

My Albums

Photos Memories

9:41 AM 100%

Messages

Search

- Megan** 8:30 AM >
Hope you have a great flight! See you tonight!
- Gram** Yesterday >
Thinking about you a lot this week
- Samantha** Saturday >
How's the new place in San Francisco?
- Mom** Friday >
Looking forward to seeing you soon!

```
// Adopt Large Titles
```

```
navigationBar.prefersLargeTitles = true
```

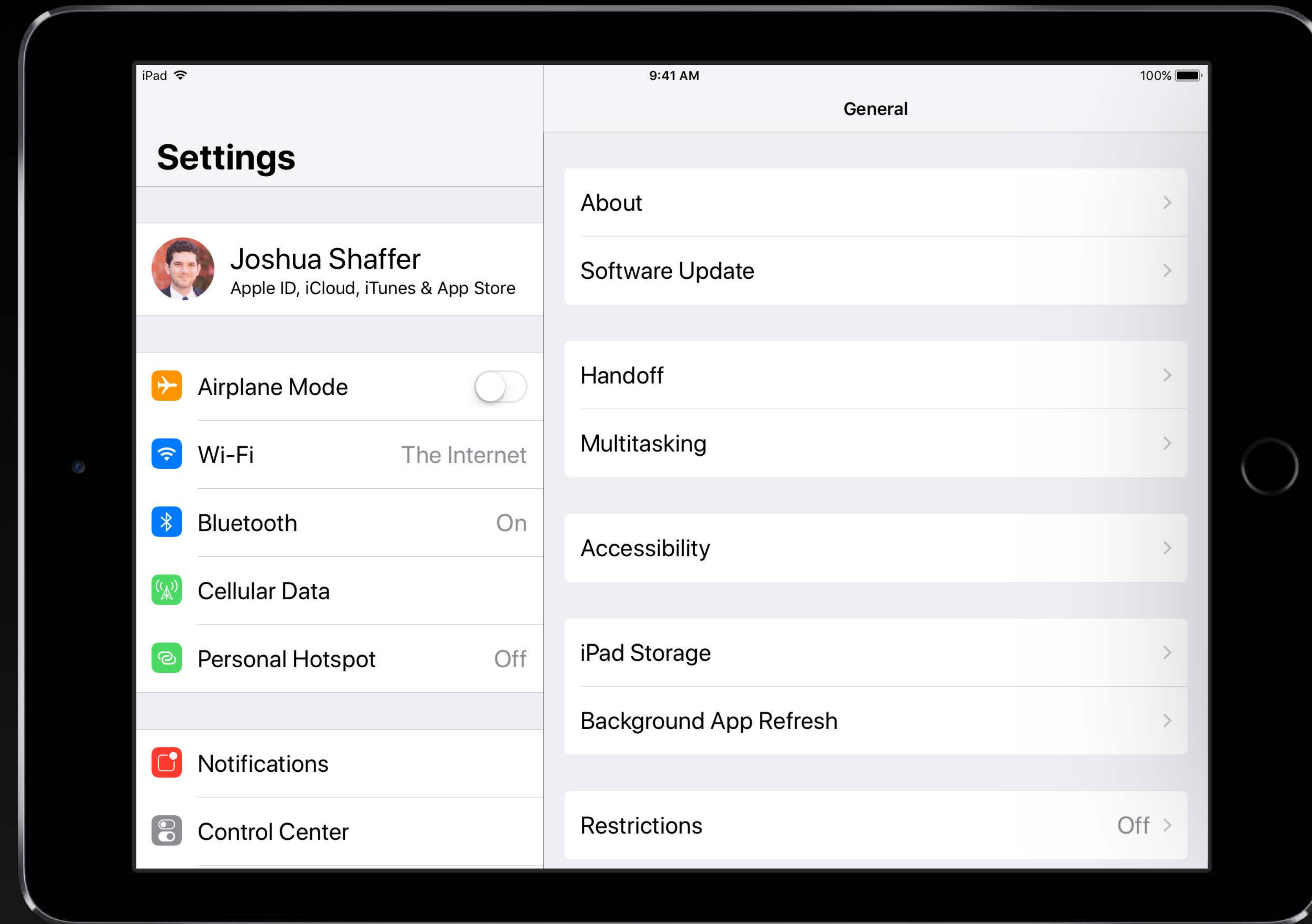
```
// Automatically Choose Large Title
```

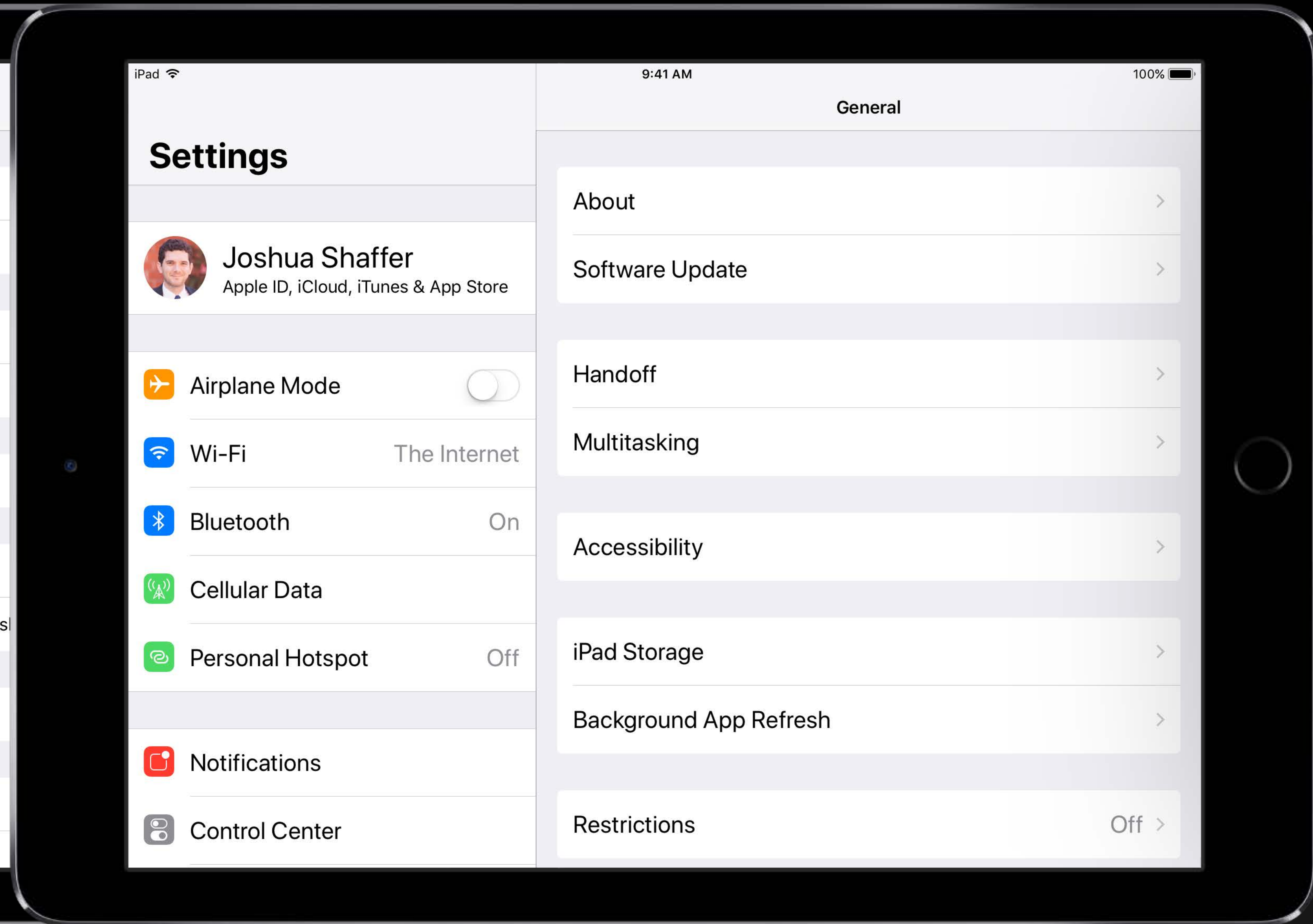
```
navigationItem.largeTitleDisplayMode = .automatic
```

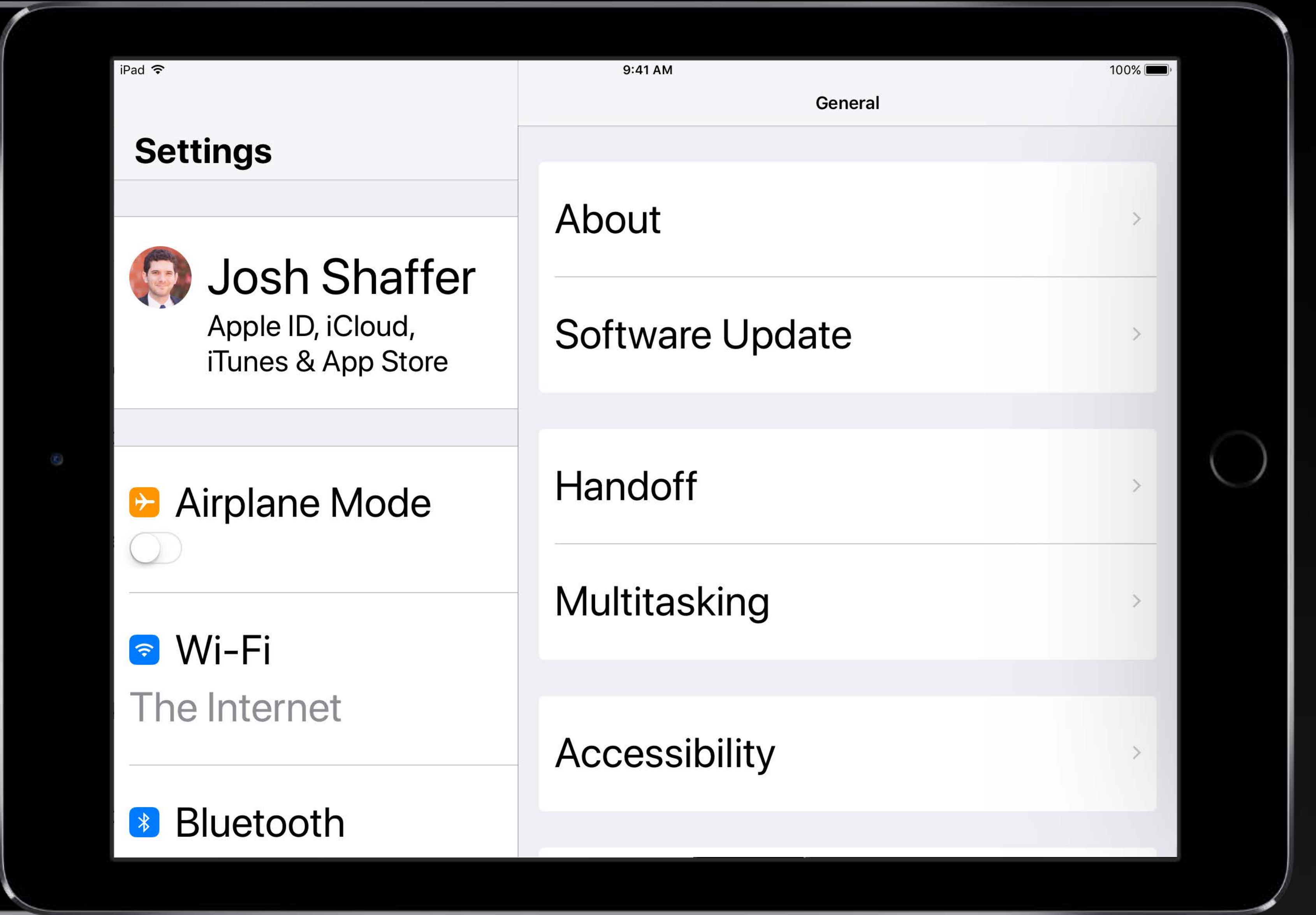
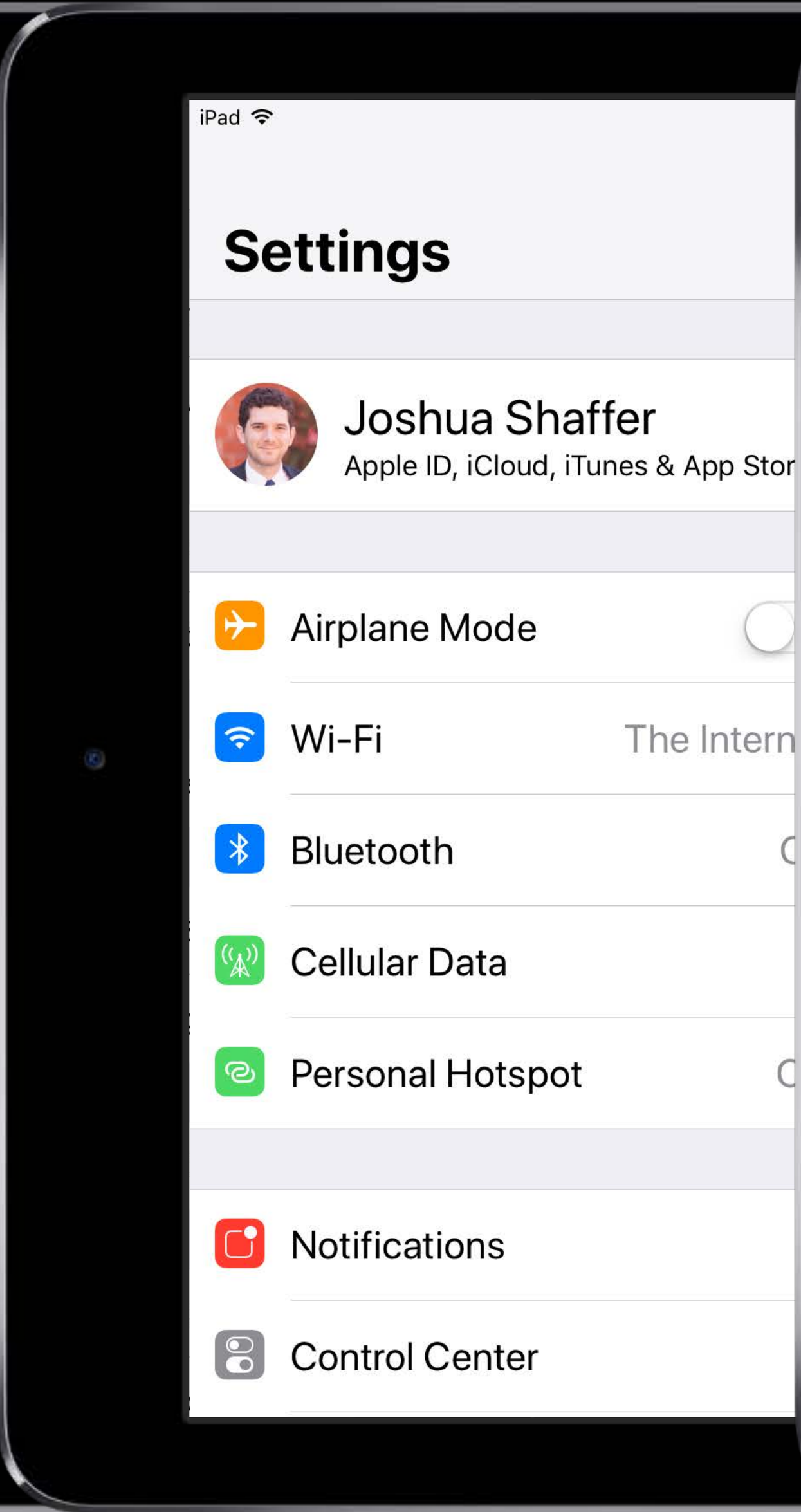
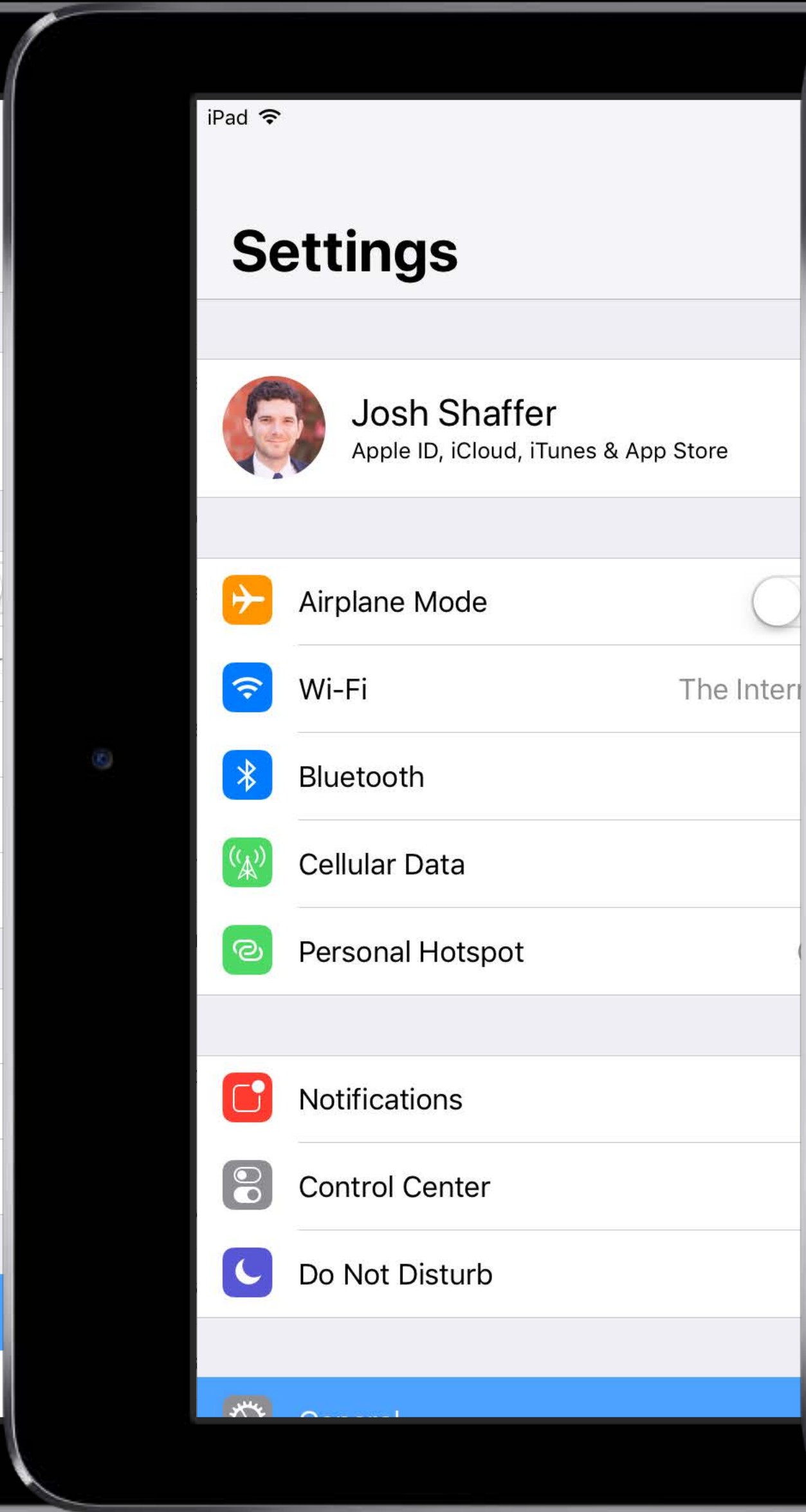
```
// Adopt Unified Search Bar
```

```
navigationItem.searchController = searchController
```


Dynamic Type

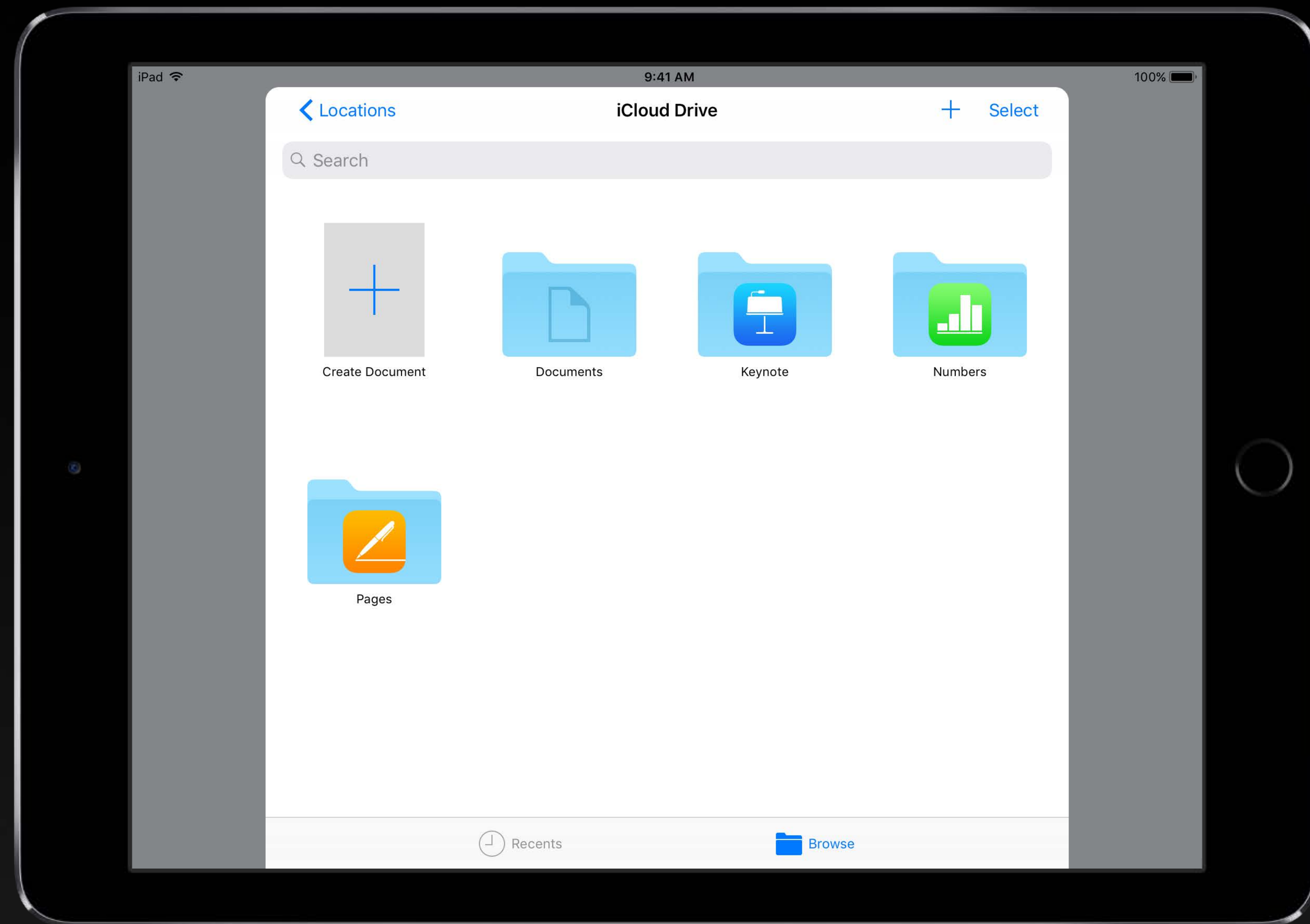






Files



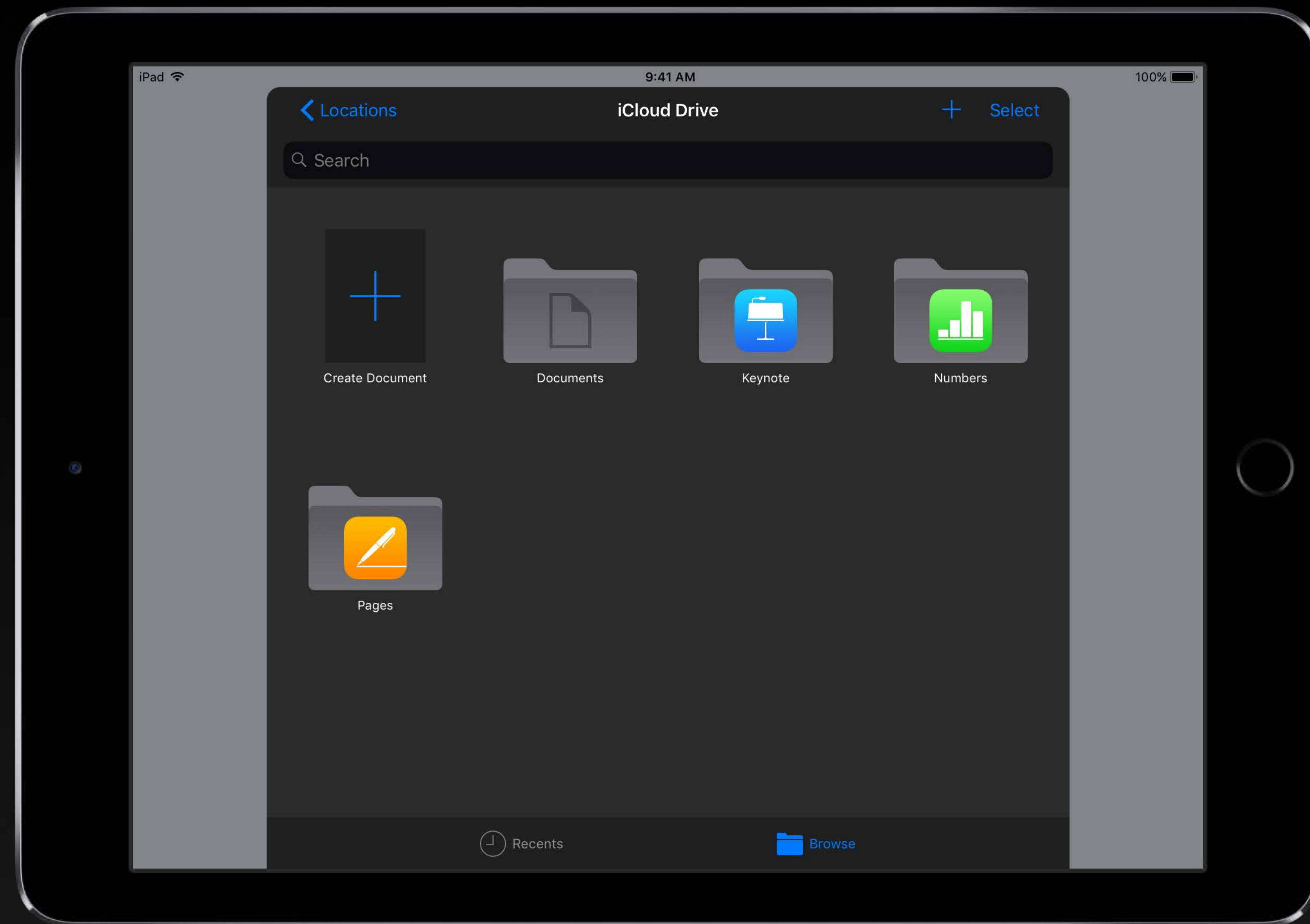


Type filtering

Custom actions

Document creation

Appearance customization

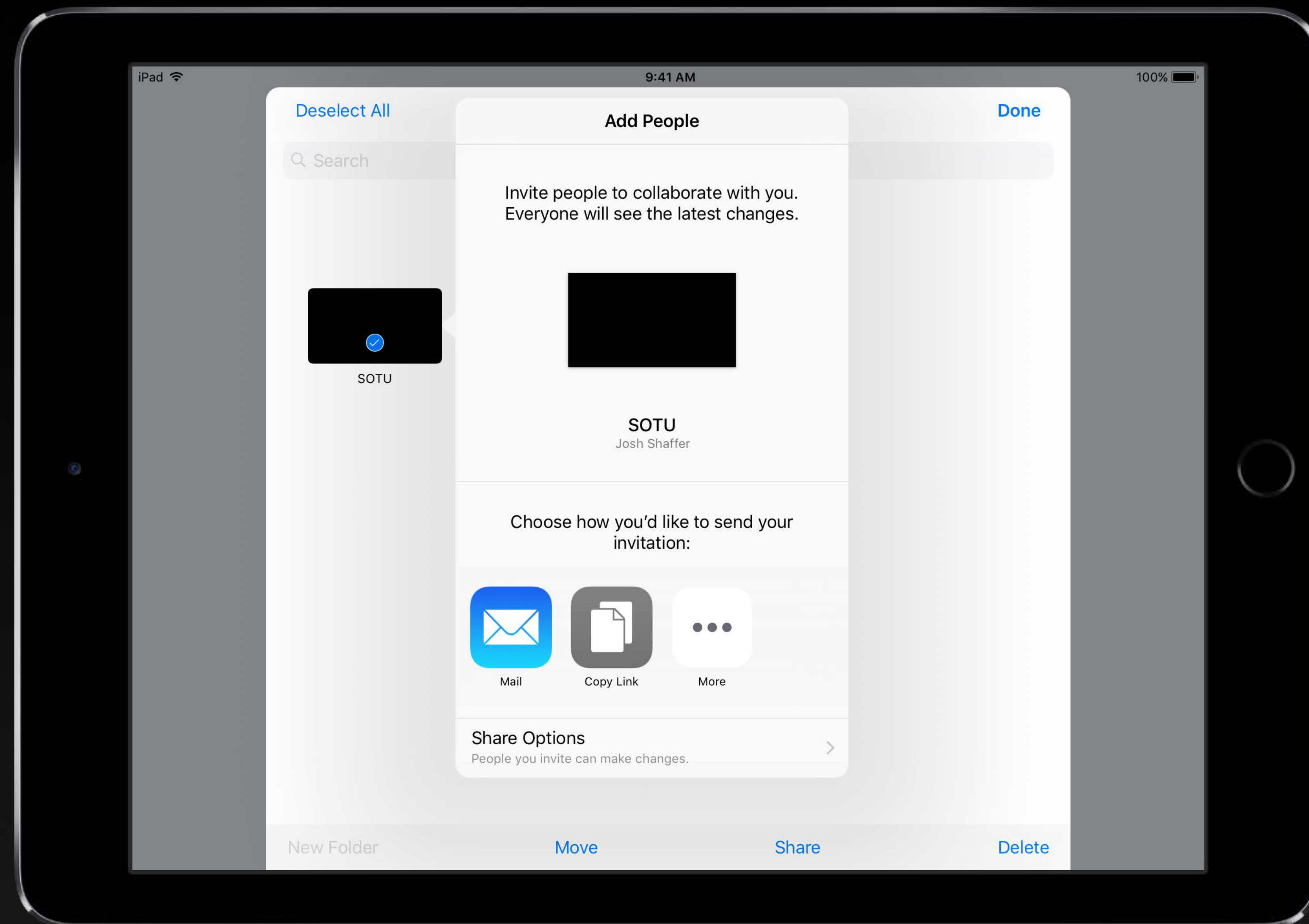


Type filtering

Custom actions

Document creation

Appearance customization

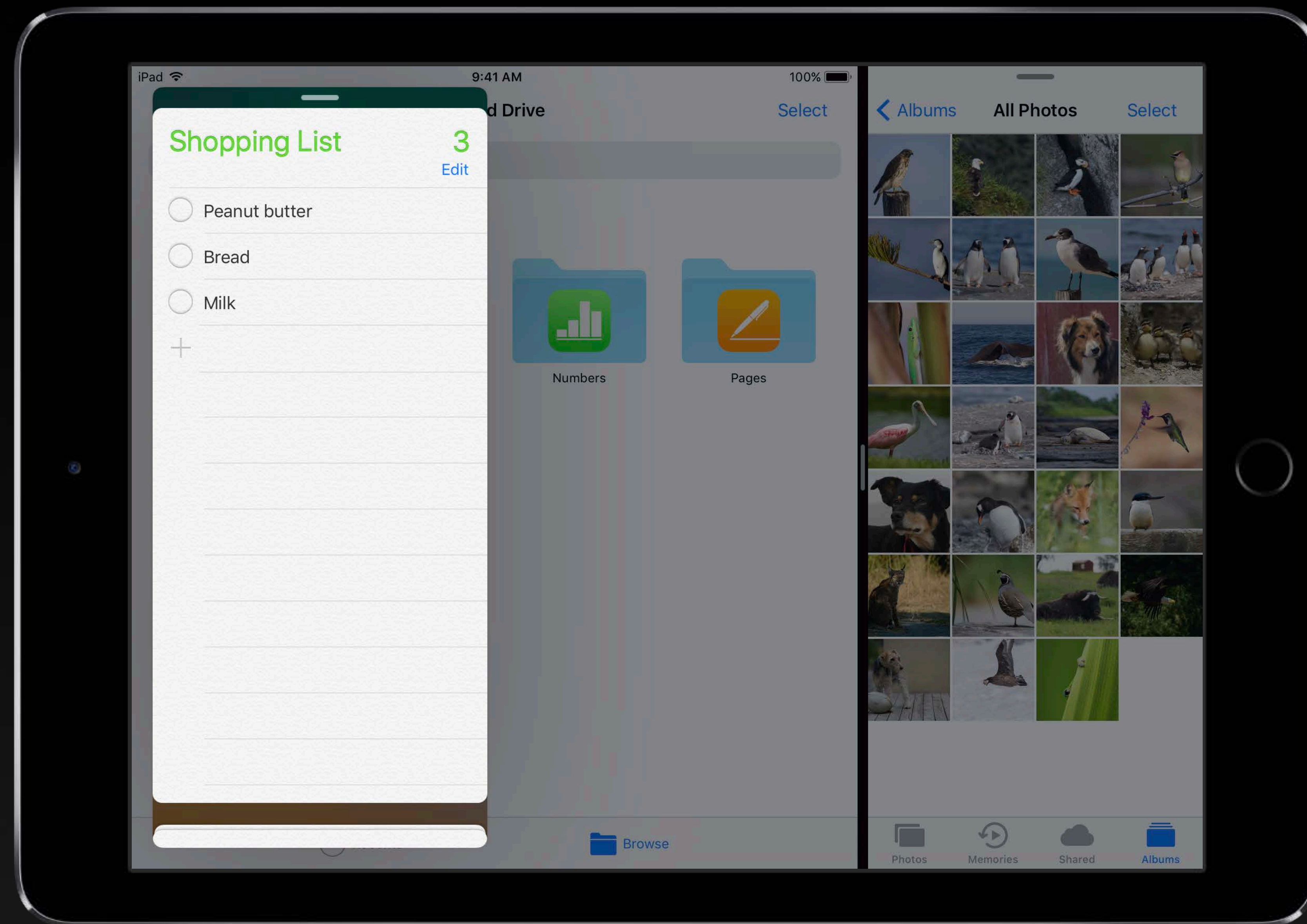


In-app document sharing

Supports iOS and macOS

Leverages NSFileCoordinator

NSDocument support



Size classes

Auto layout

Default storyboards

HomeKit Event Triggers

Password Autofill

QuickLook Thumbnails

Screen Edge Protect

ReplayKit with Screen Broadcast

Named Color Assets

HEVC/HEIF

Property Animator Enhancements

FileProvider for File Management

Swift Archival and Serialization

Vector Assets

CoreSpotlight

Focus Debugger

UIFontMetrics

PDFKit

Collection View RTL

Paste Configurations

Export to PDF

Smart Quotes

AV Route Picker

SceneKit Animations

Document Picker

Photos AutoLoop

CALayer Corner Masking





App strip

Apps in the transcript

Direct send



Payment accounts

Lists

Notes

QR codes



40 million songs

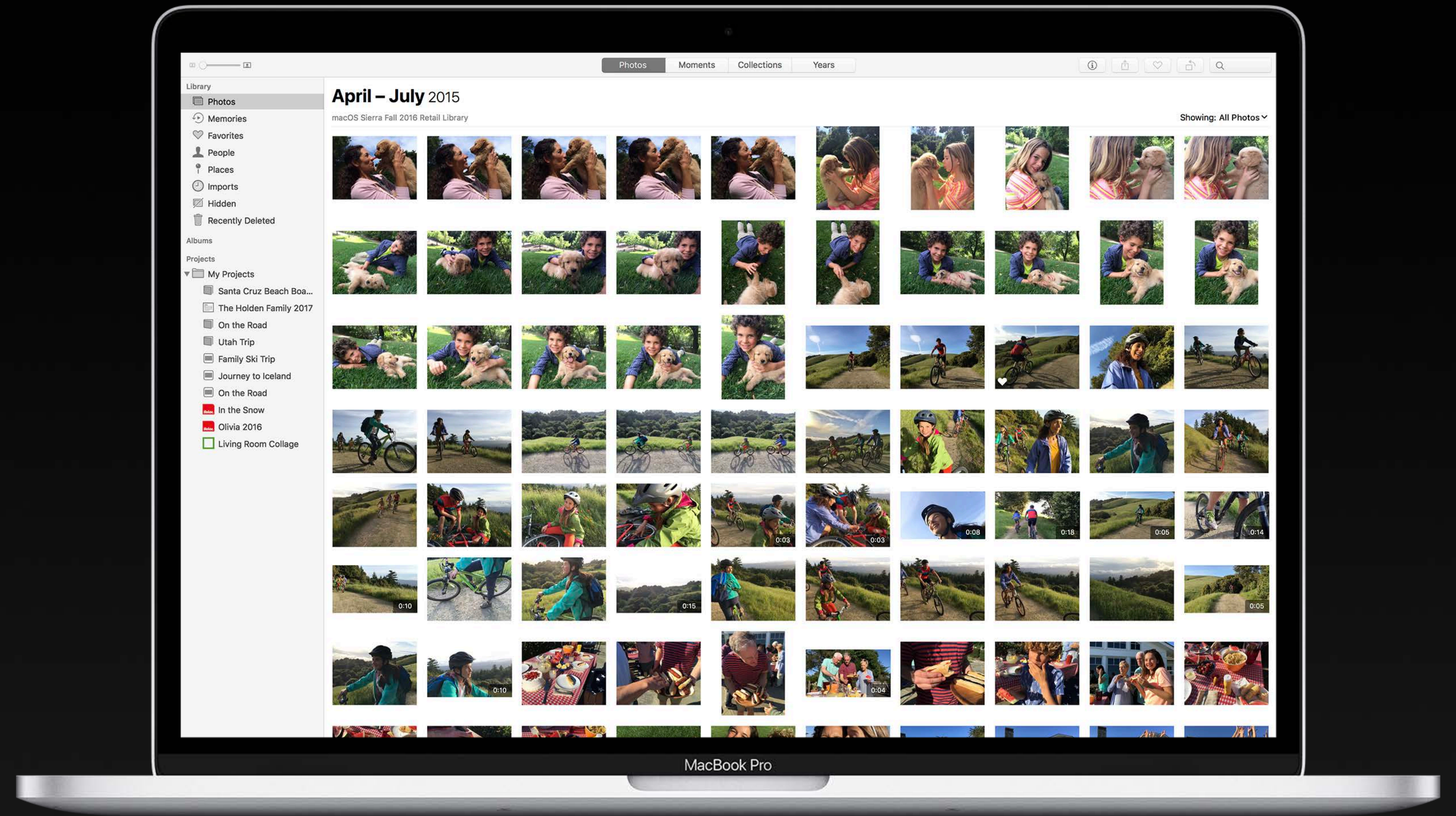
Play songs and playlists

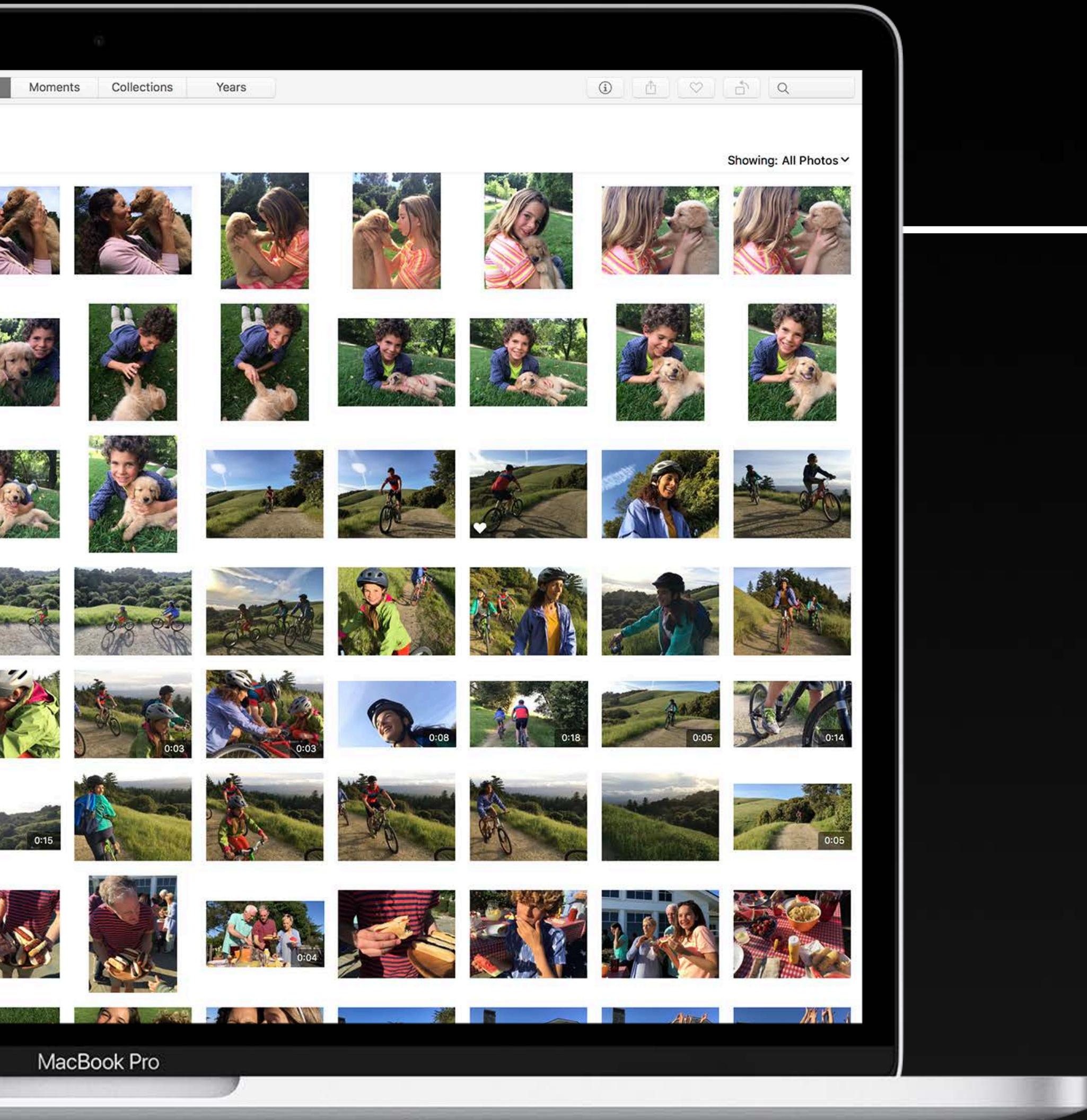
Listen to radio stations



Photos

Photos Project Extensions





Photos

Project Metadata

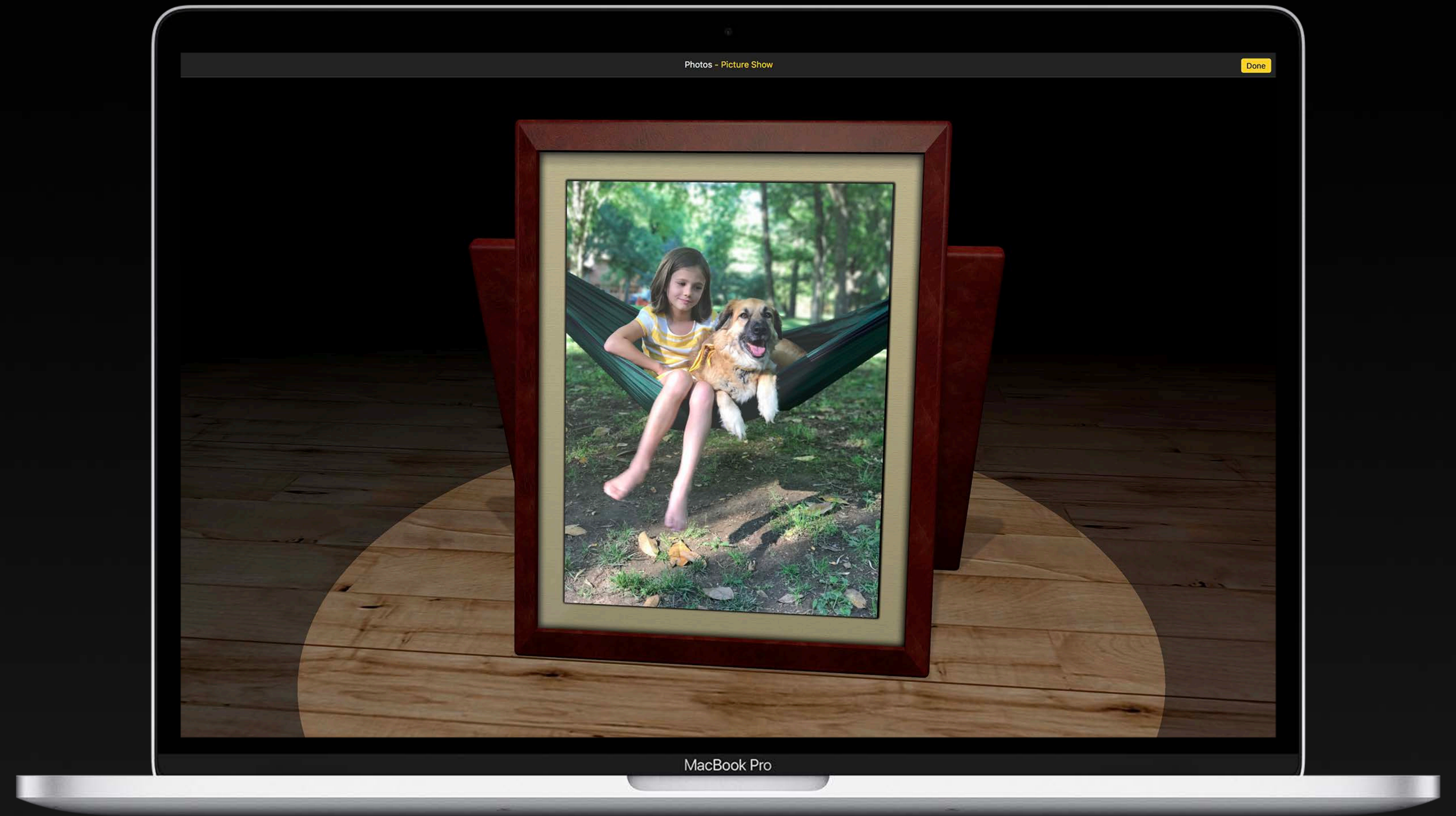


Extension Private Data



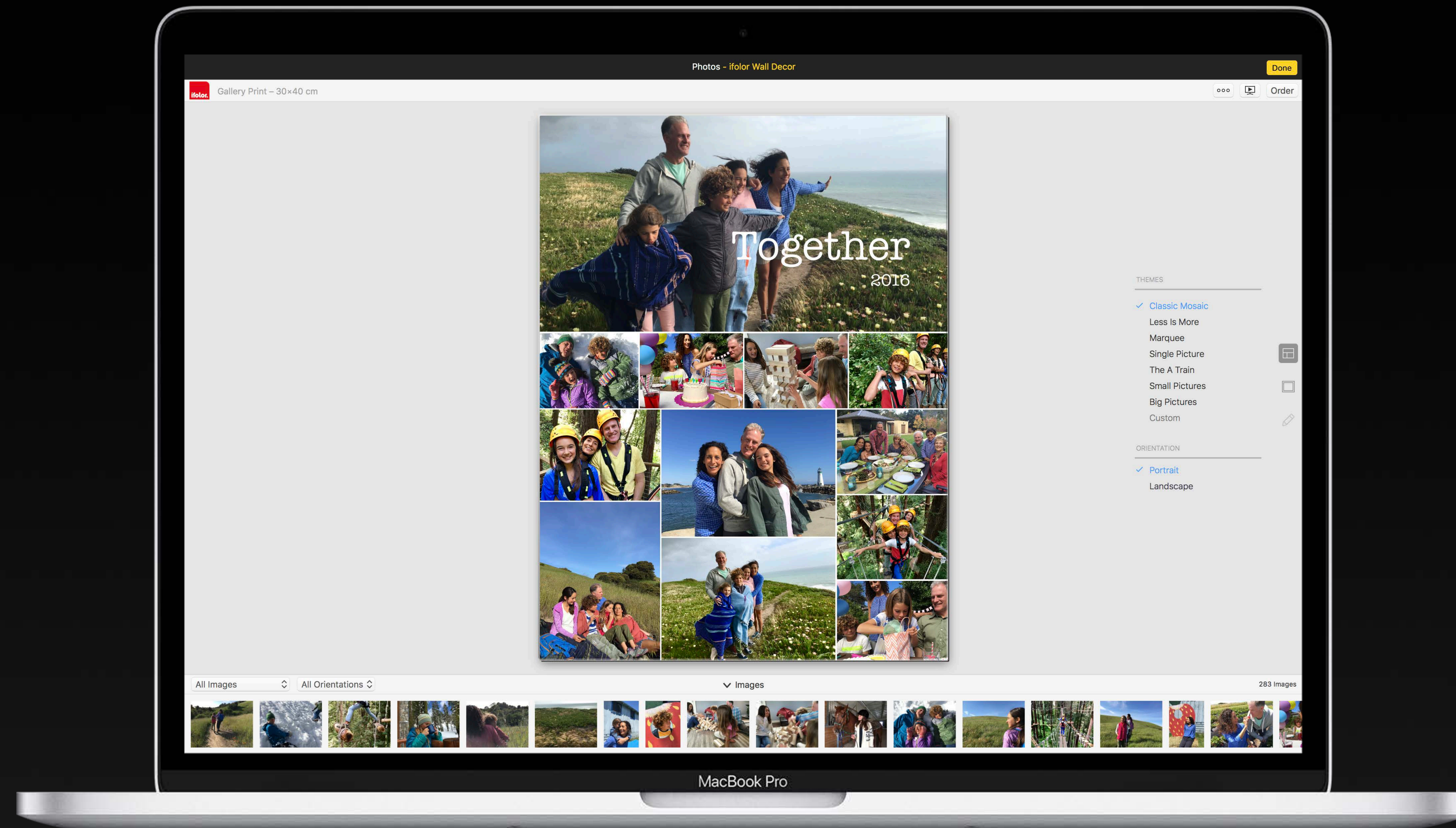
Extension

Extension UI Hosted in Photos



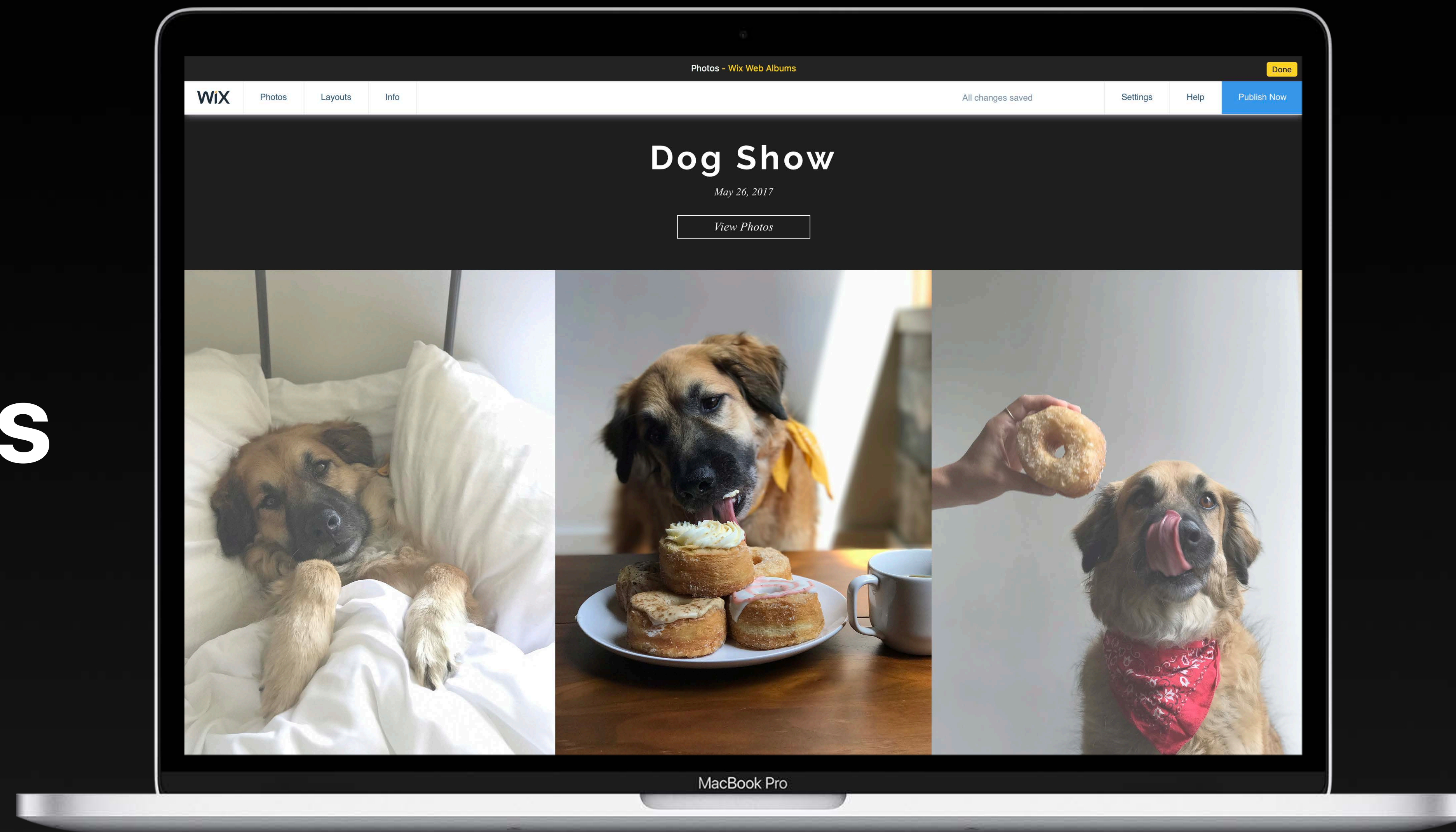


Print Products



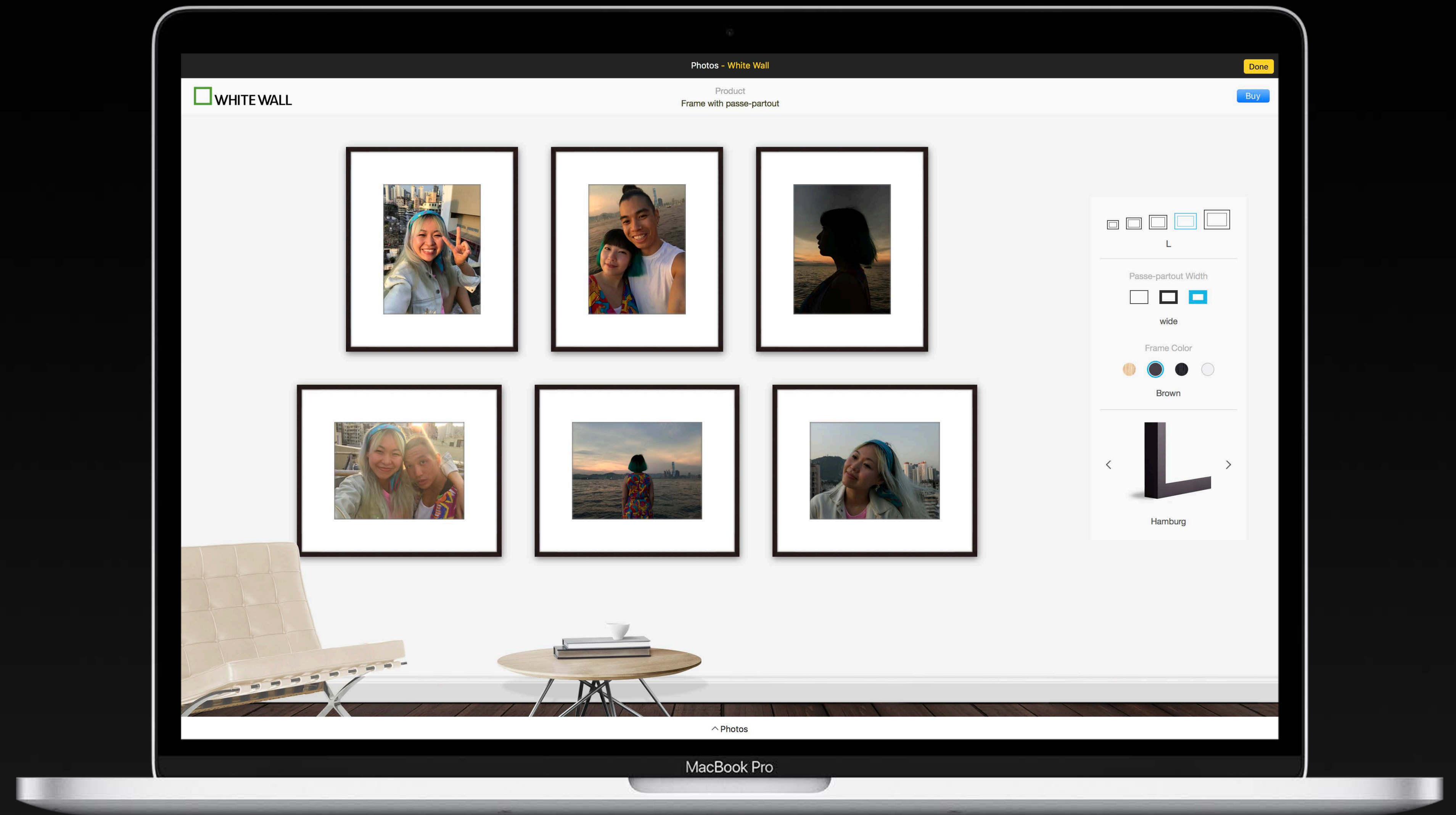
WIX.com

Web Photo Albums



 WHITE WALL

Framed Prints





Camera



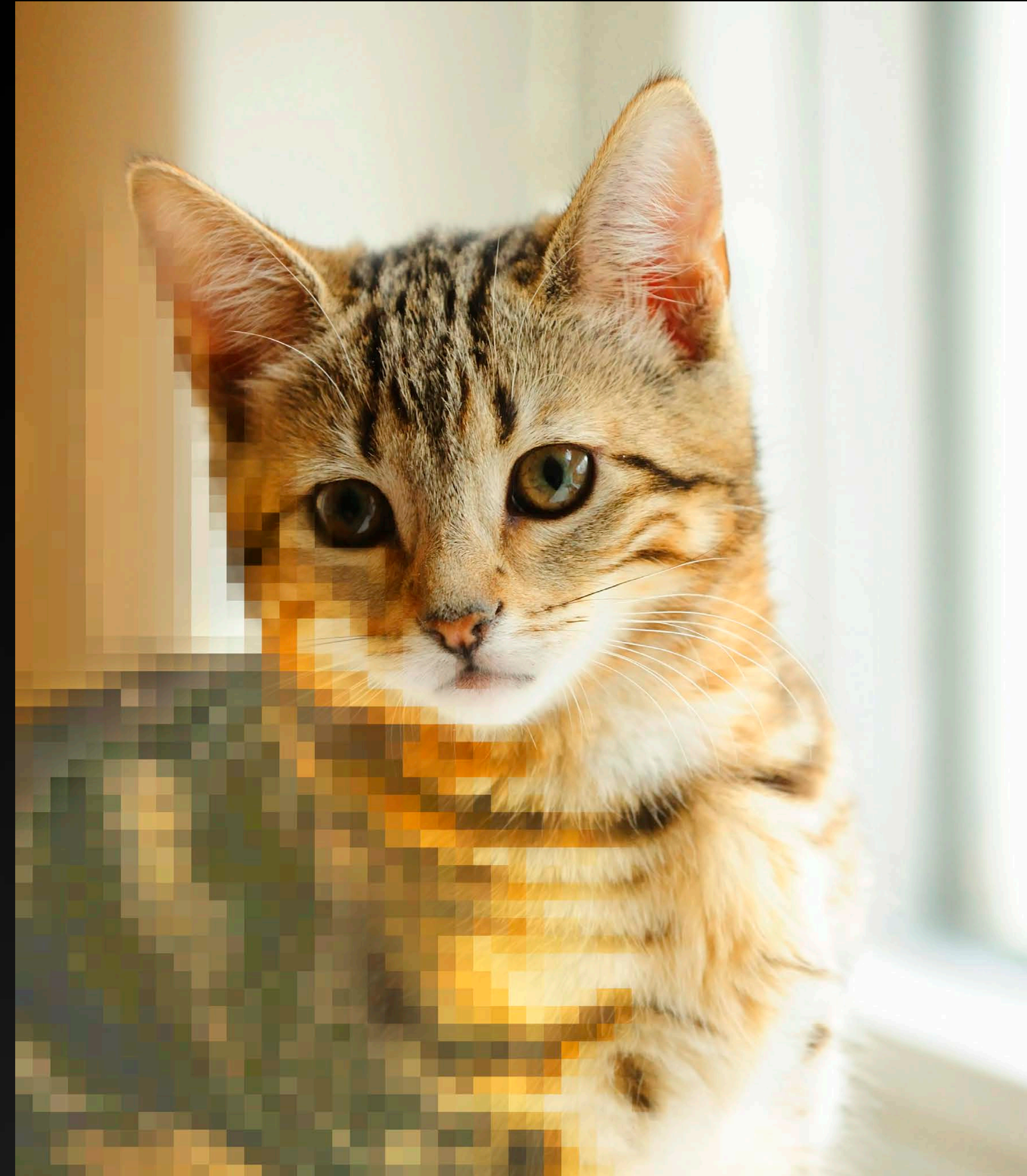
Camera detects QR codes

Link into apps via Universal Links



HEVC and HEIF

JPEG – 1992





H.264 – 2003

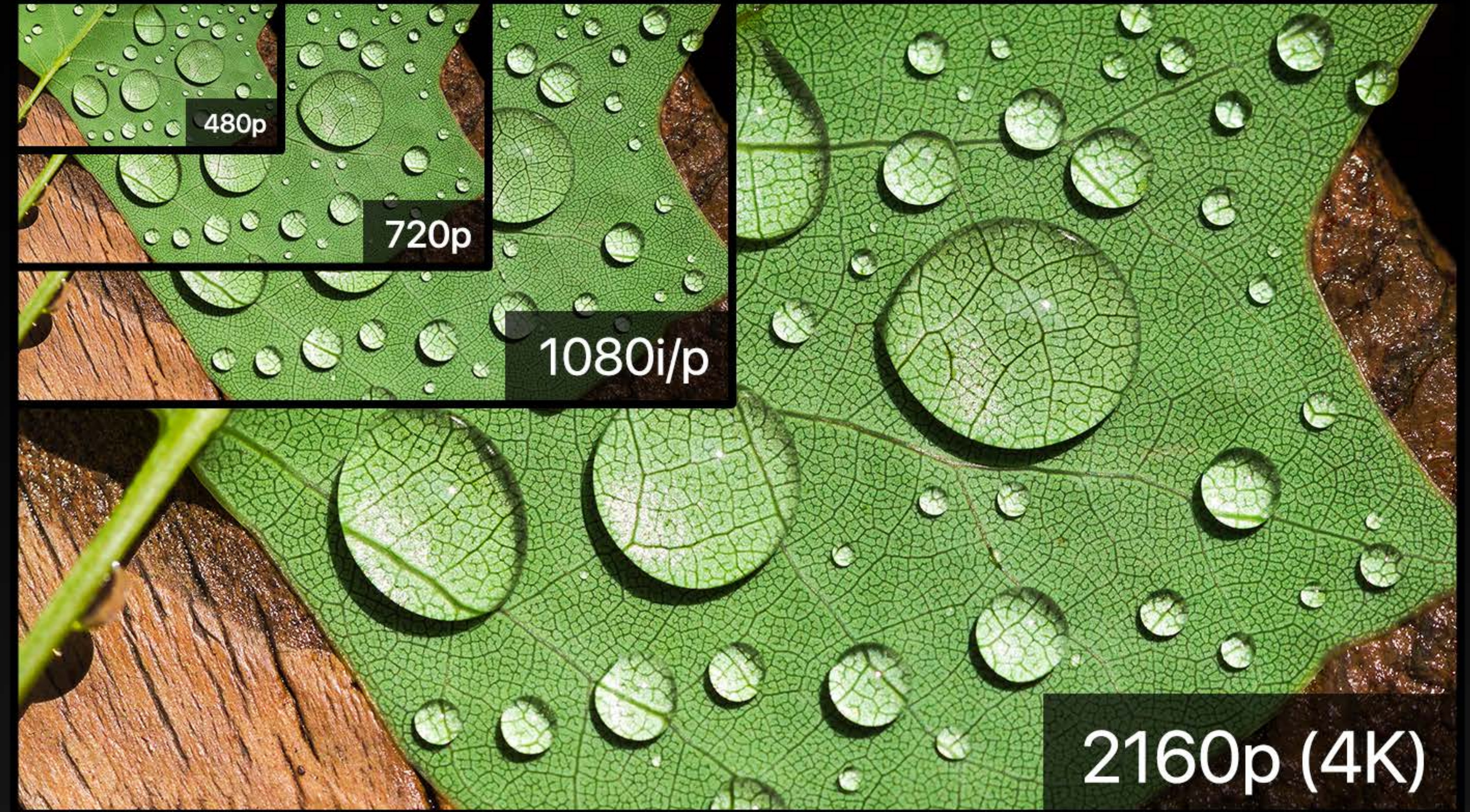
Complex Assets



**High Dynamic Range
Wide Color**



High Resolution



HEVC



Compression



Performance



Scalability

HEVC



Compression



Performance

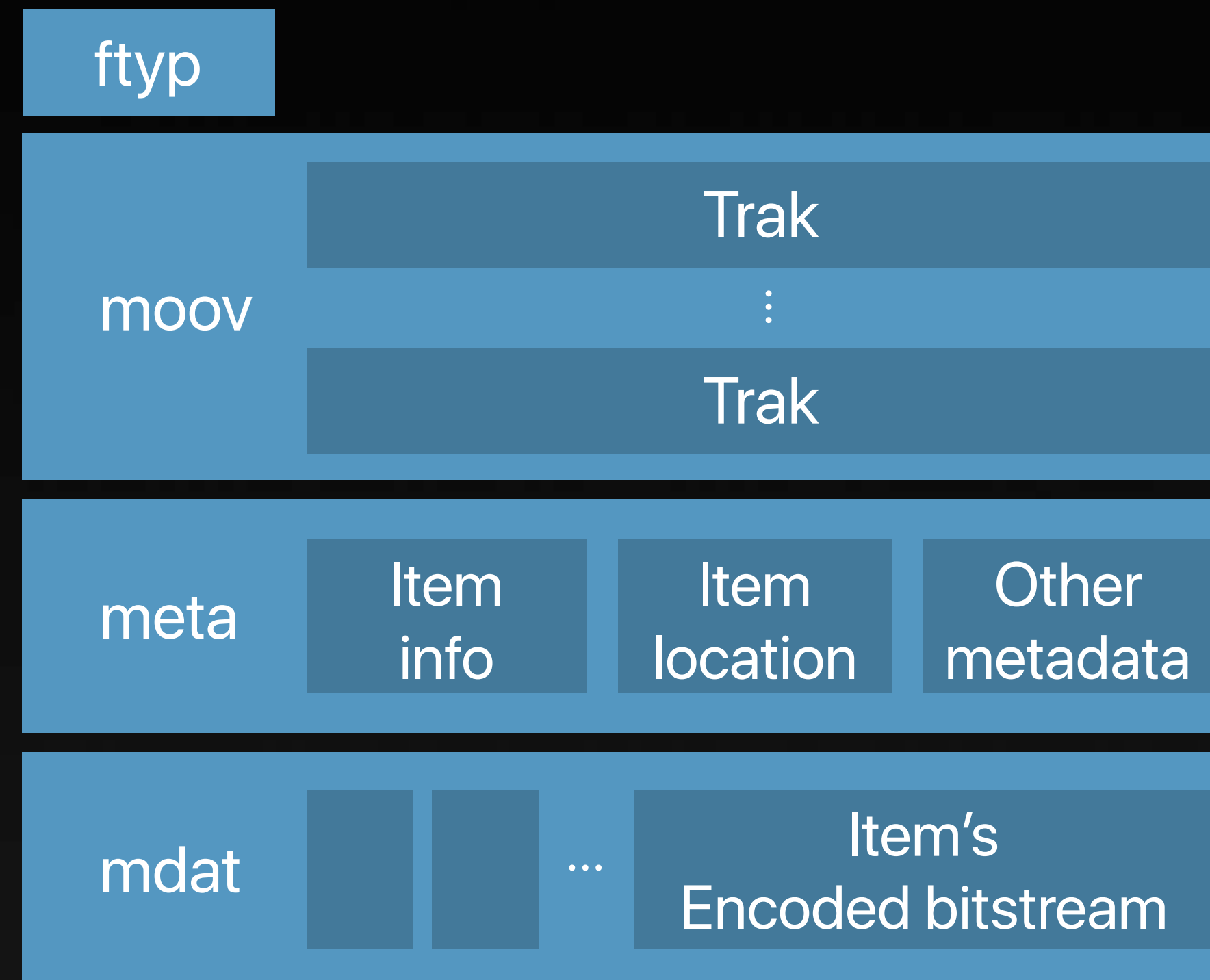


Scalability

HEIF



Compound Assets



Extensible and Rich



Modern Container



ImageIO

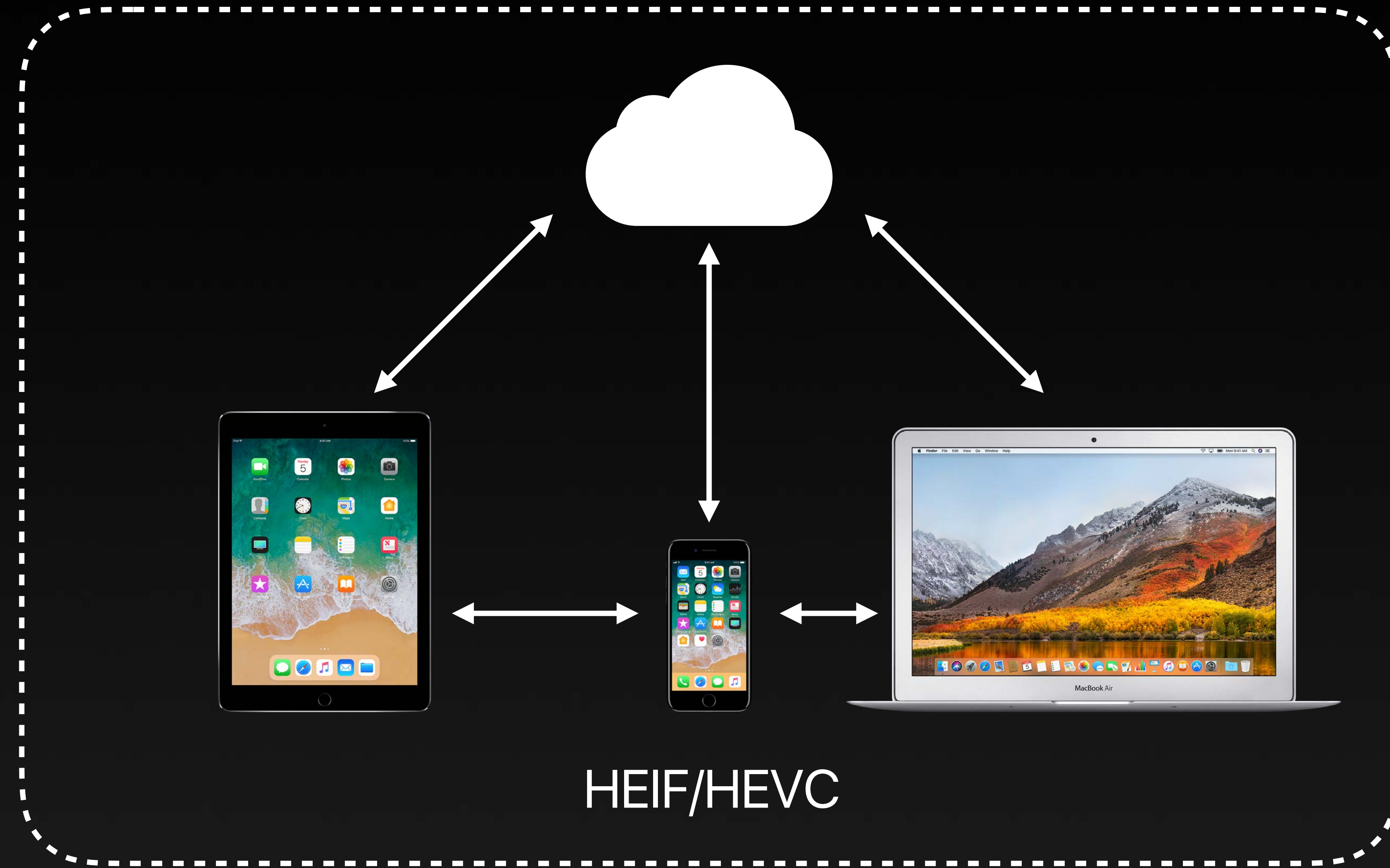
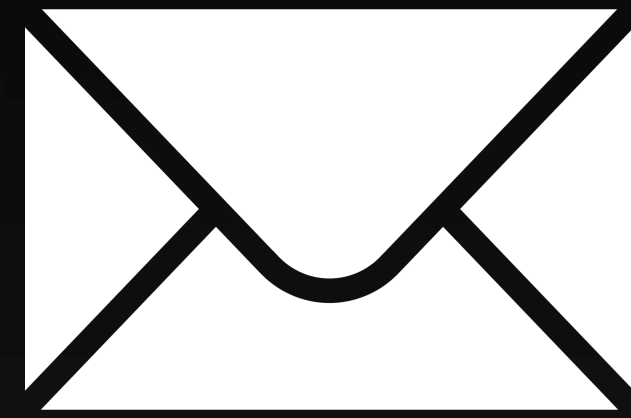
Core Image

AVFoundation

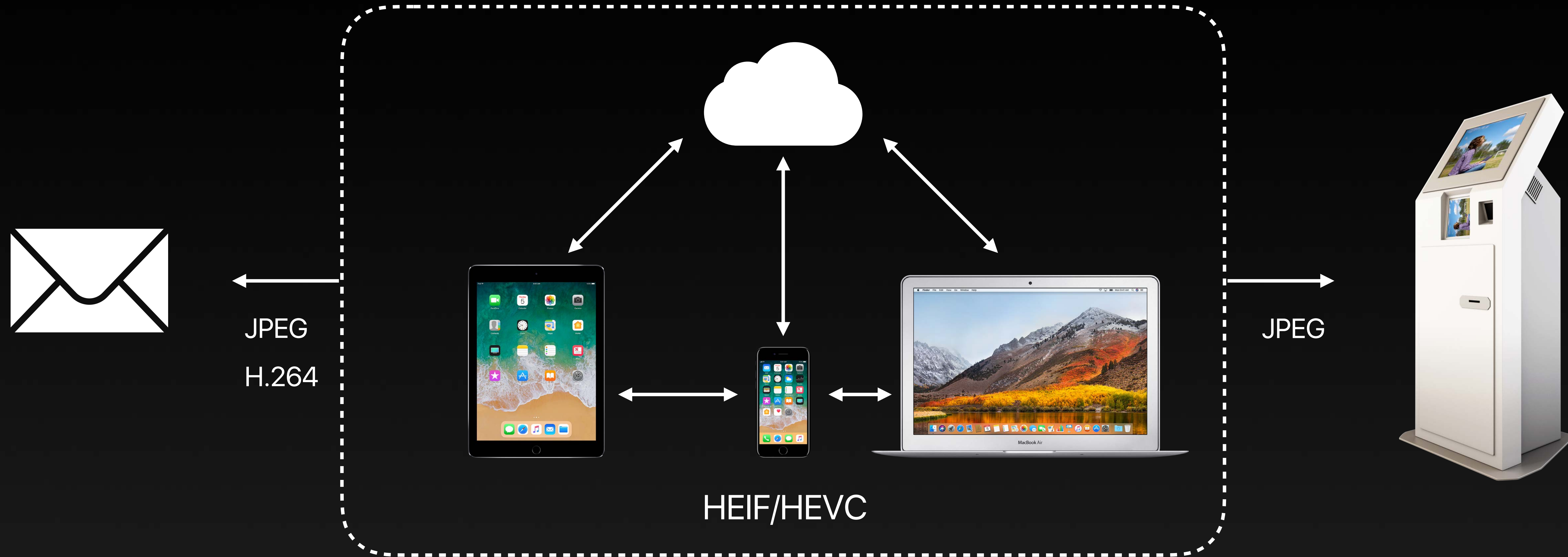
PhotoKit

```
var asset = AVAsset(url: URL(fileURLWithPath: "hevc.mov"))
if !asset.isPlayable {
    asset = AVAsset(url: URL(fileURLWithPath: "h264.mov"))
}
```


Ecosystem



Ecosystem





Depth





Wide Camera

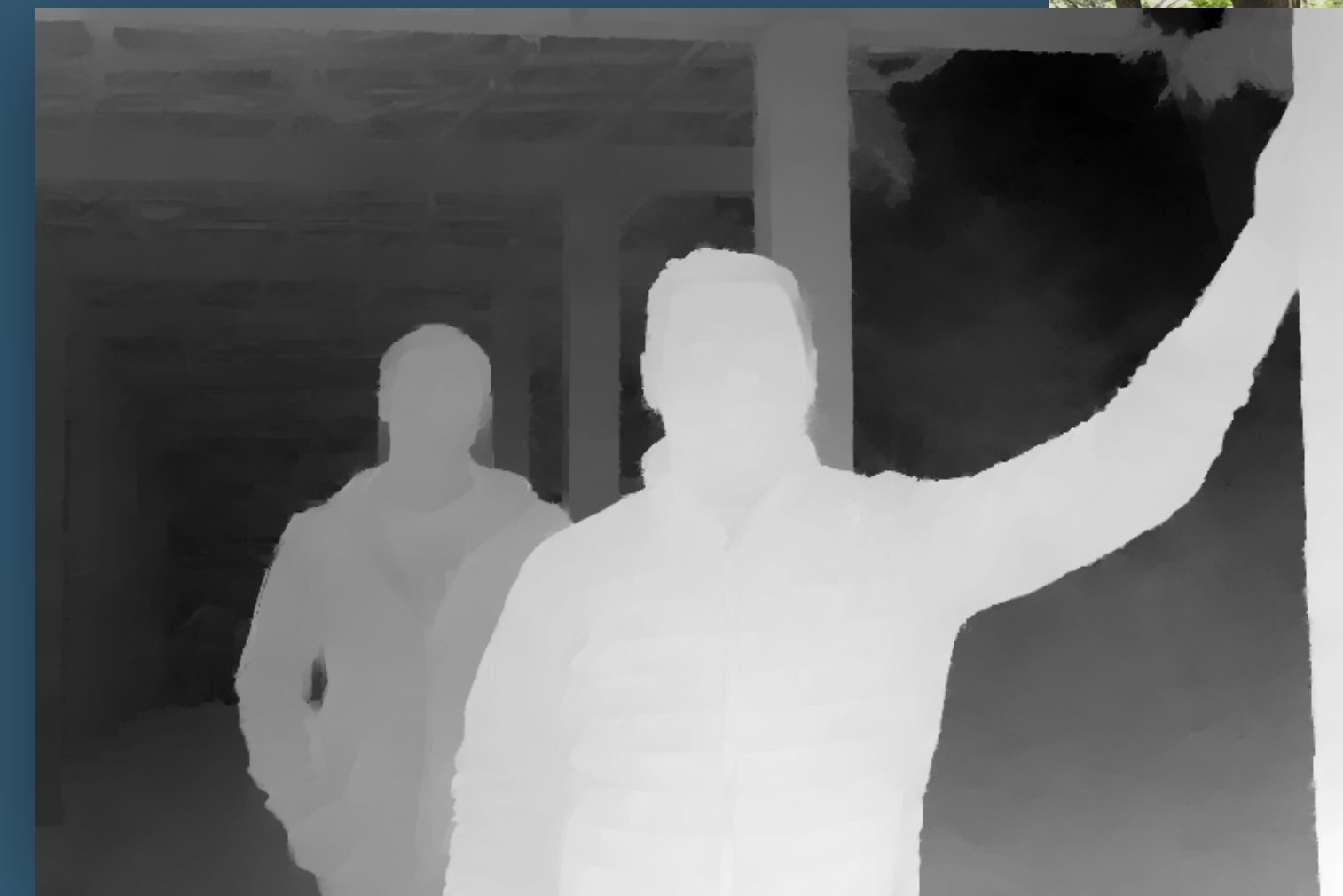
Tele Camera







Depth

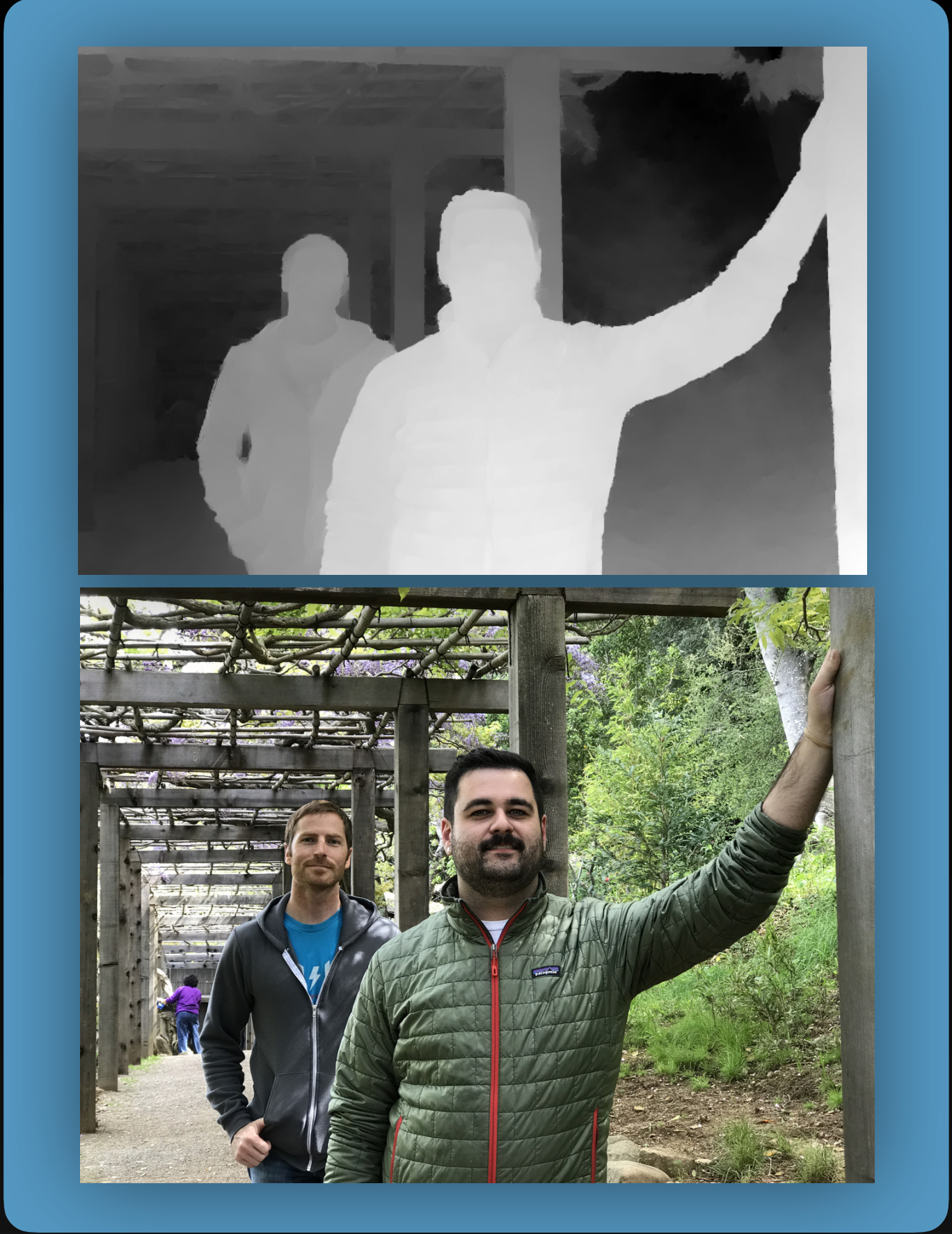








Depth of Field Effect



HEIF



HEIF





HEIF

```
CIImage(contents:)
```

```
CIImage(contents:  
options:[kCIImageAuxiliaryDepth:YES])
```



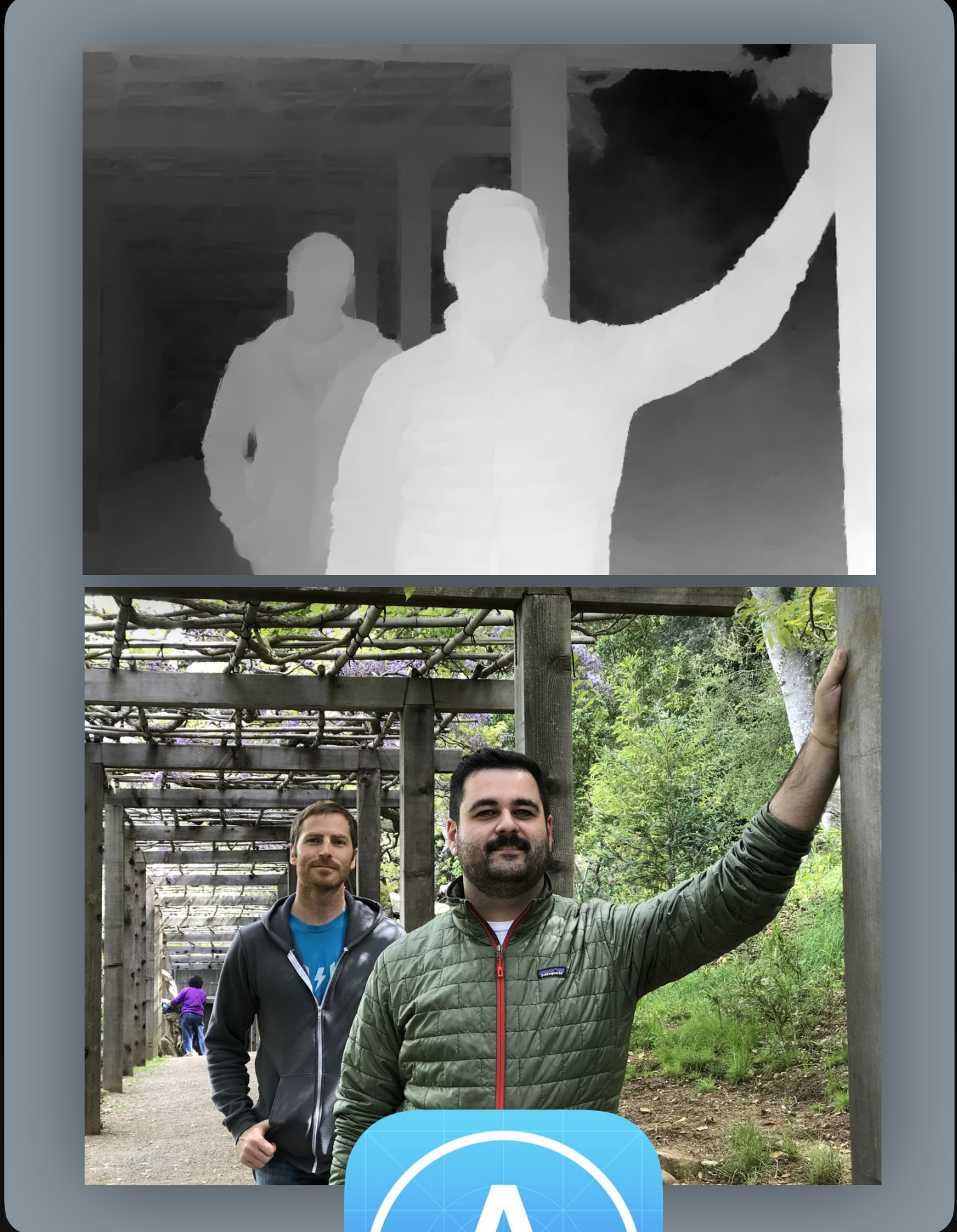


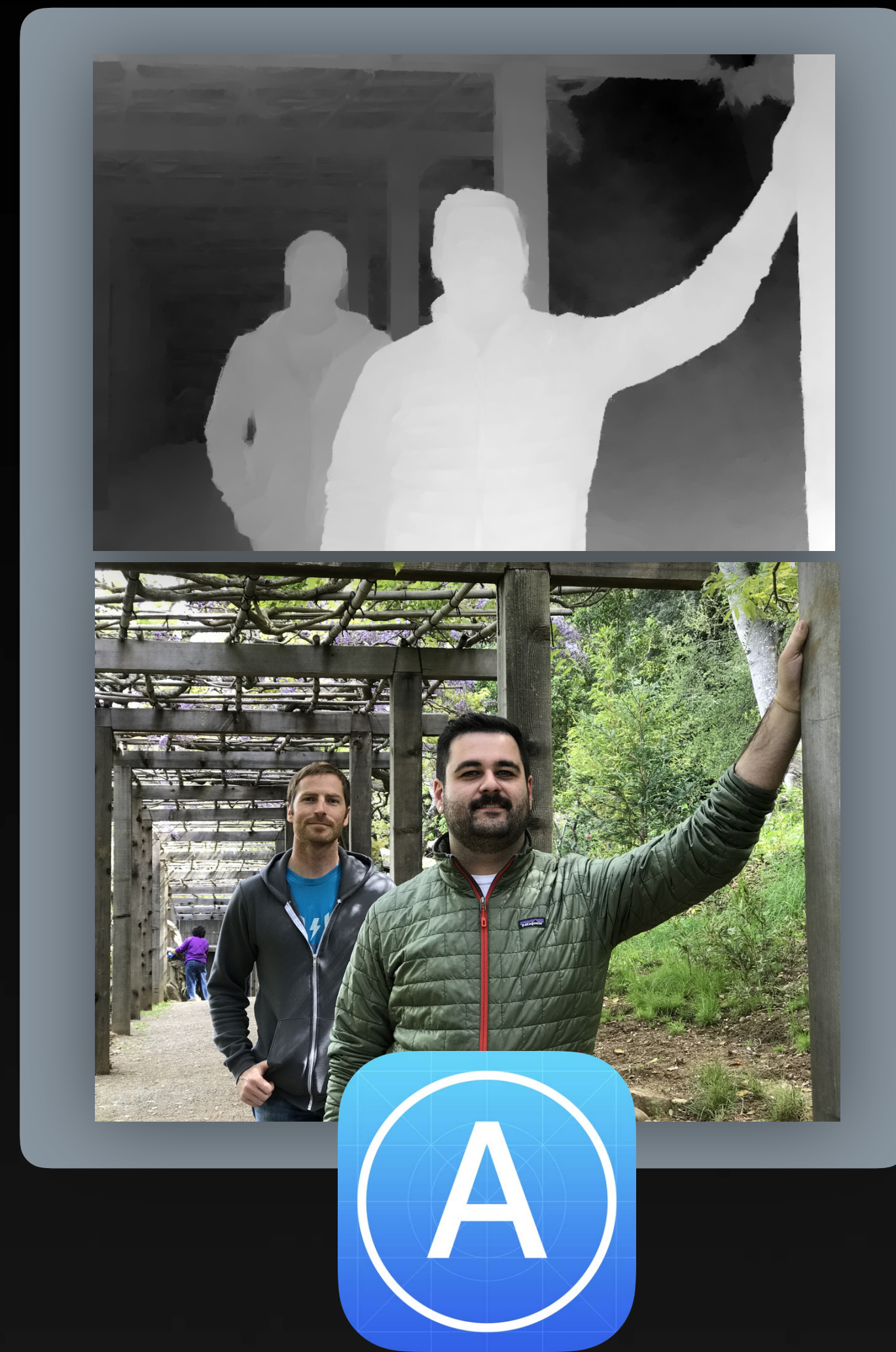
HEIF

```
CIImage(contents:)
```

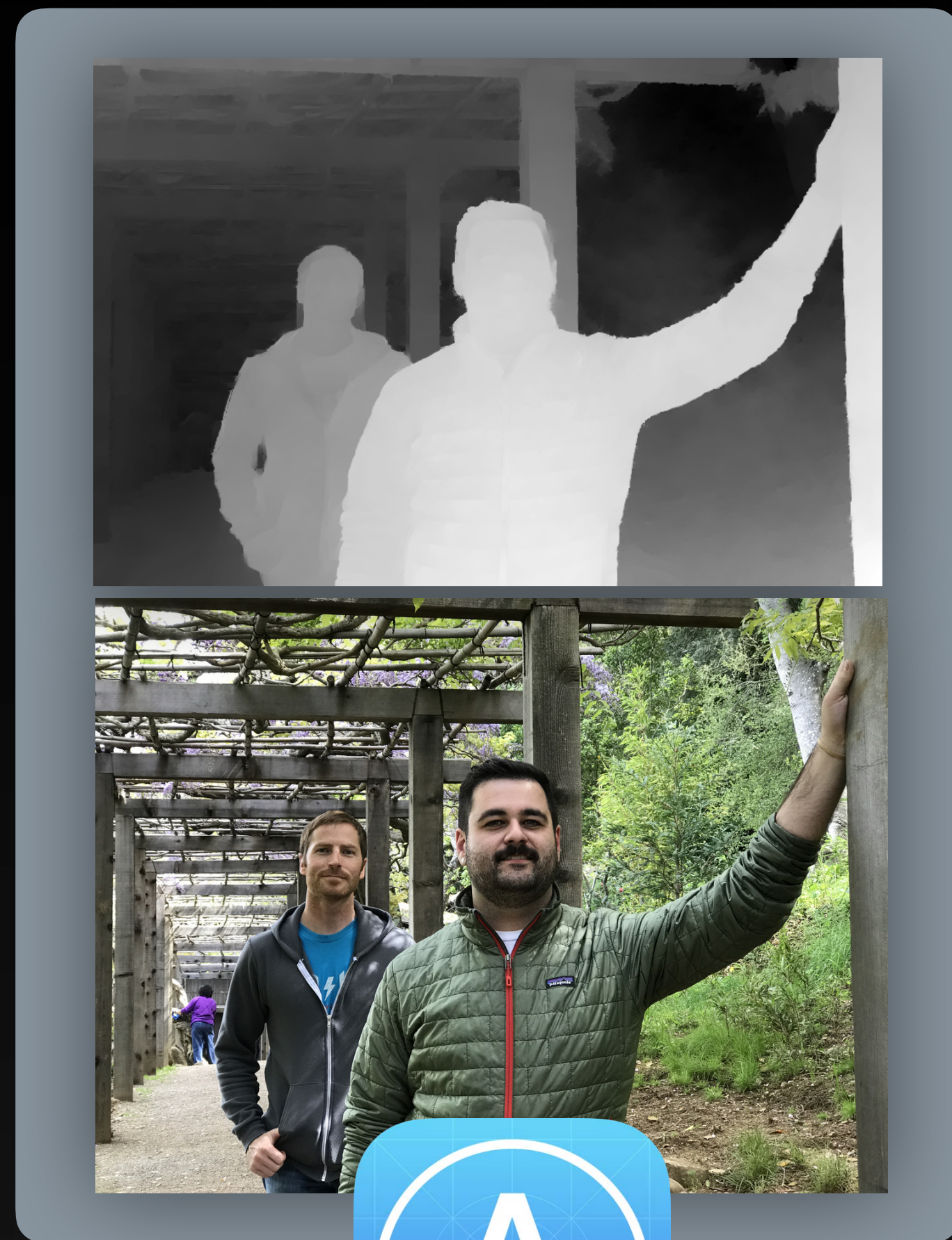
```
CIImage(contents:  
options:[kCIImageAuxiliaryDepth:YES])
```



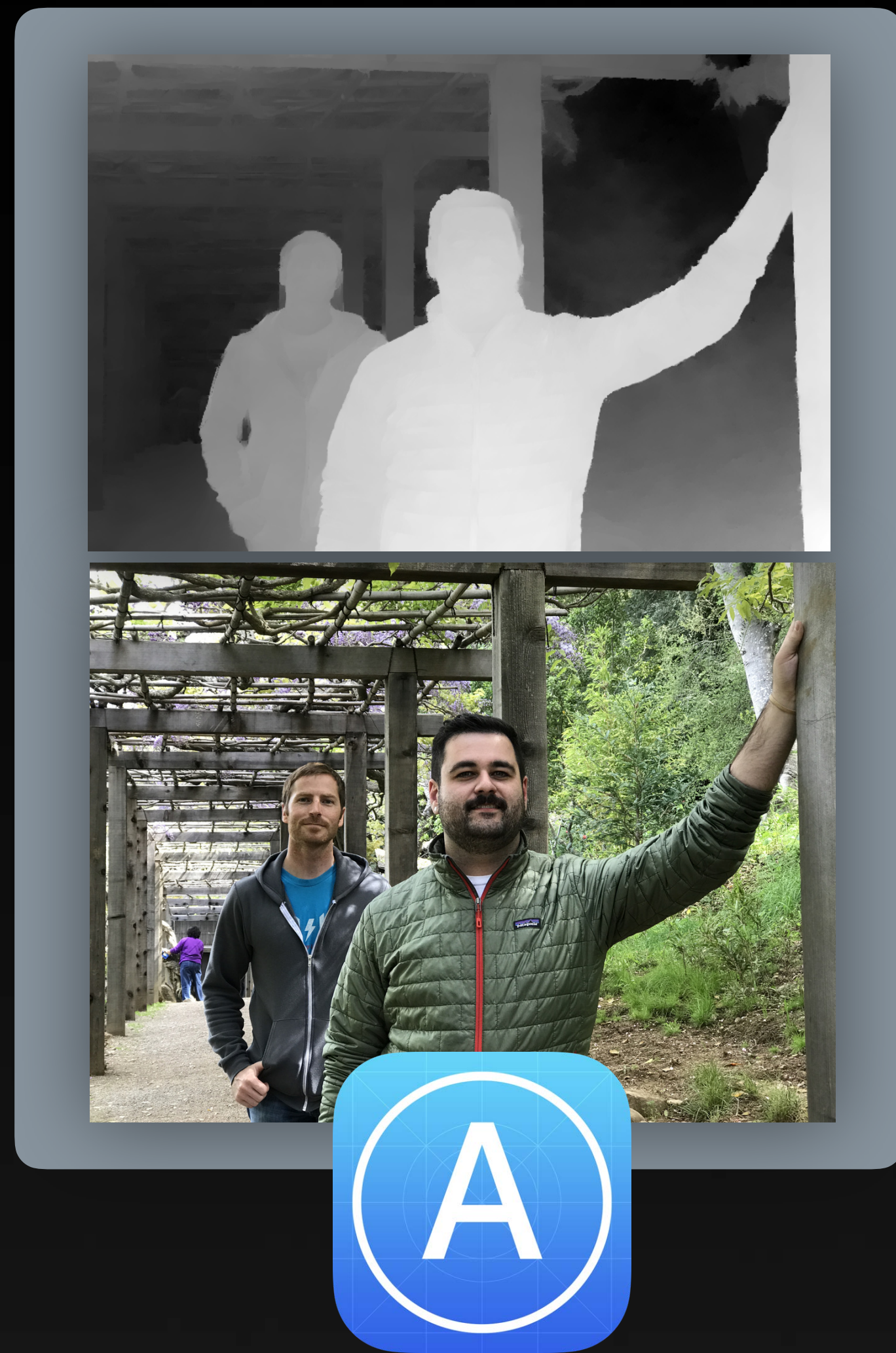




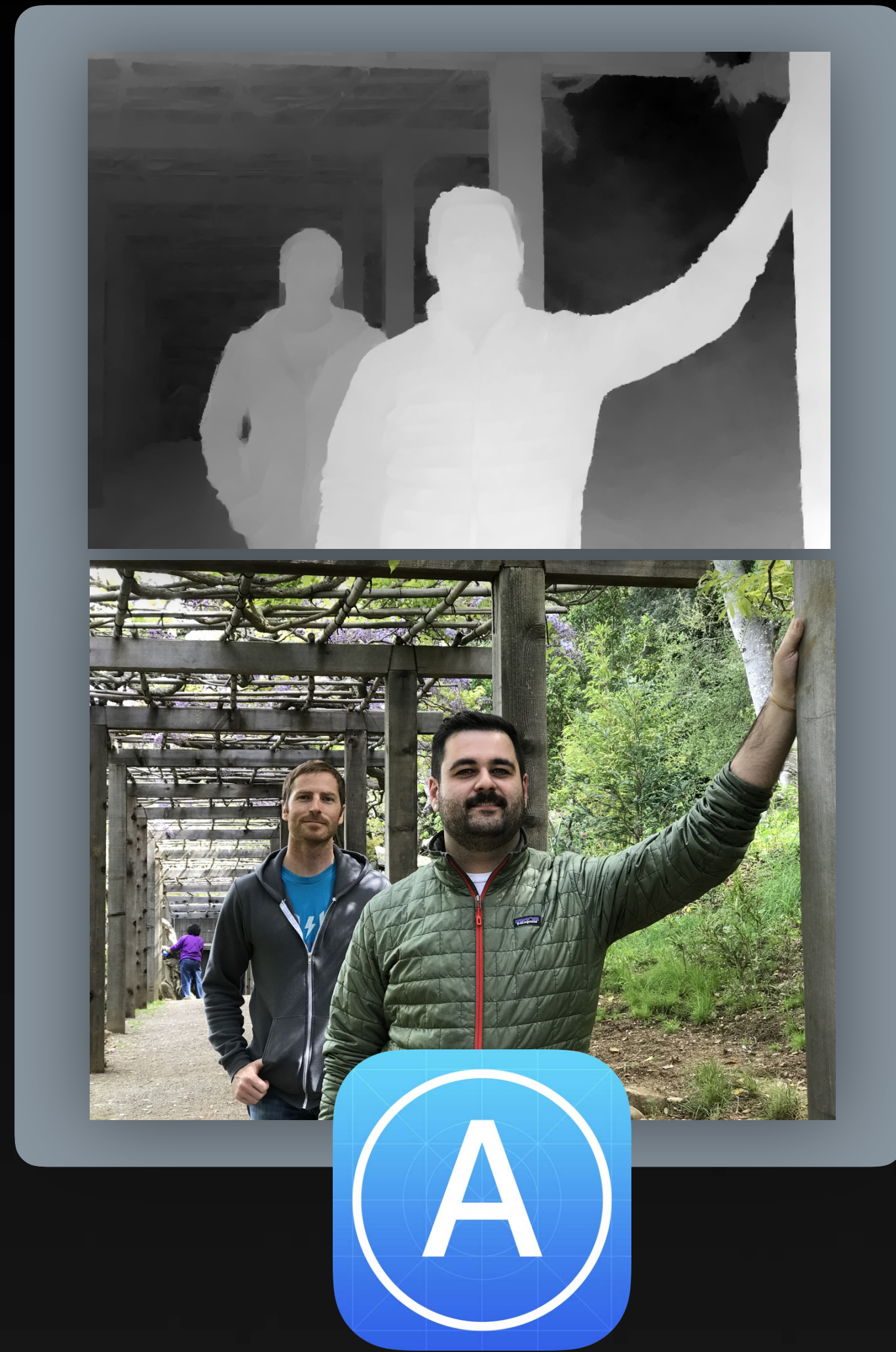
Use Depth to create custom filters



Use Depth to create custom filters



Use Depth to create custom filters



Use Depth to create custom filters

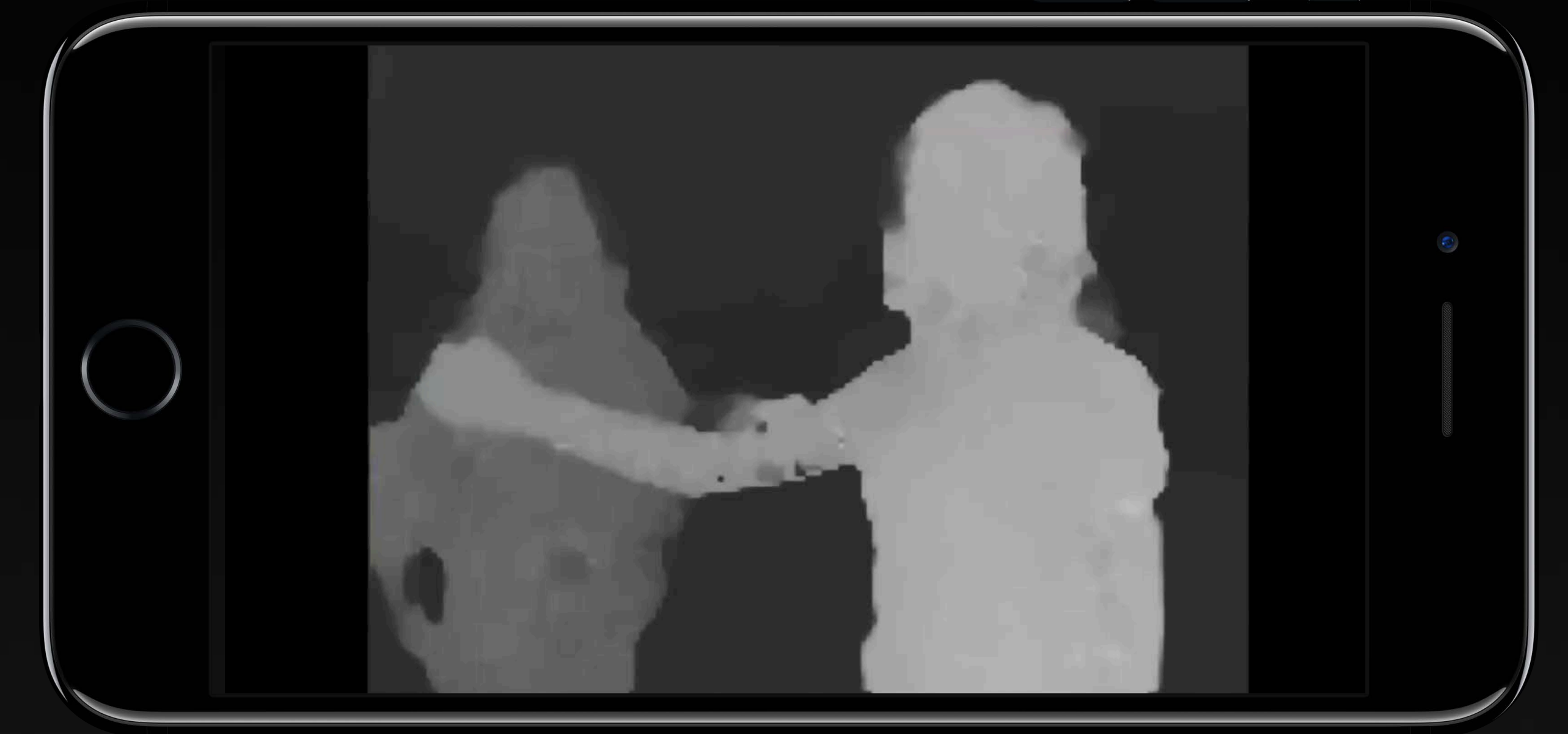
Streaming Depth API



Camera Stream



Hardware Depth Engine



AVDepthData

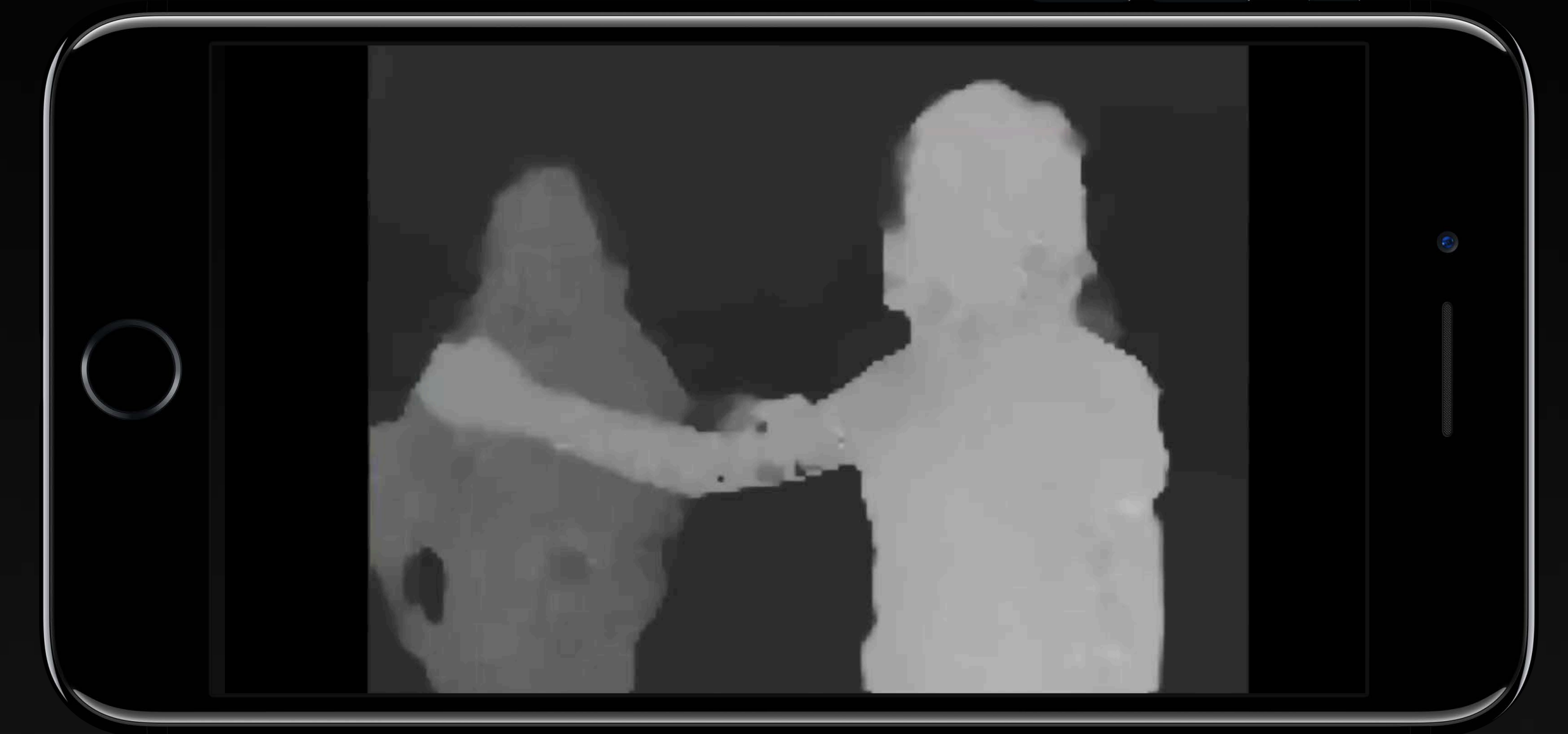
Streaming Depth API



Camera Stream



Hardware Depth Engine



AVDepthData



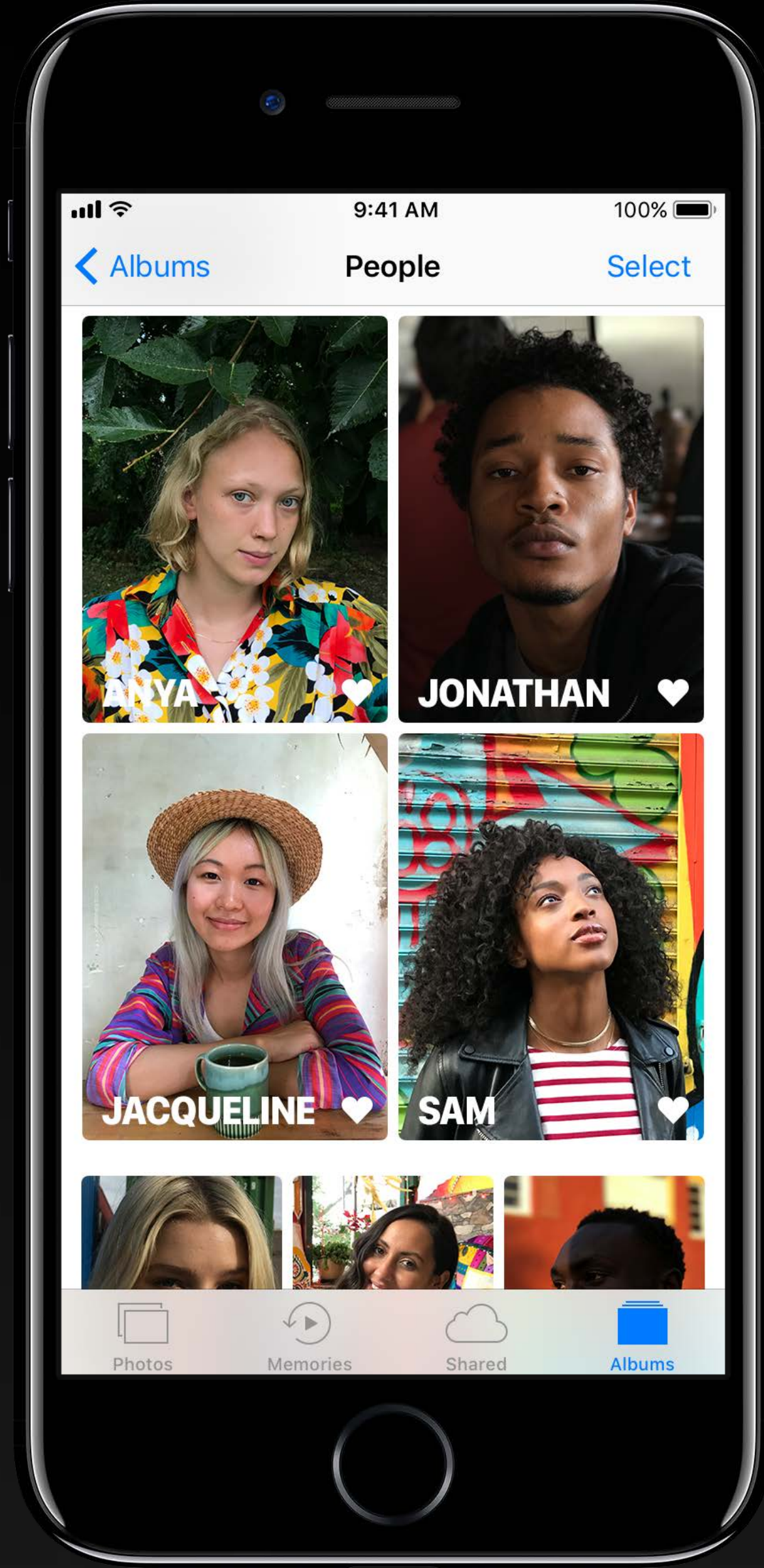
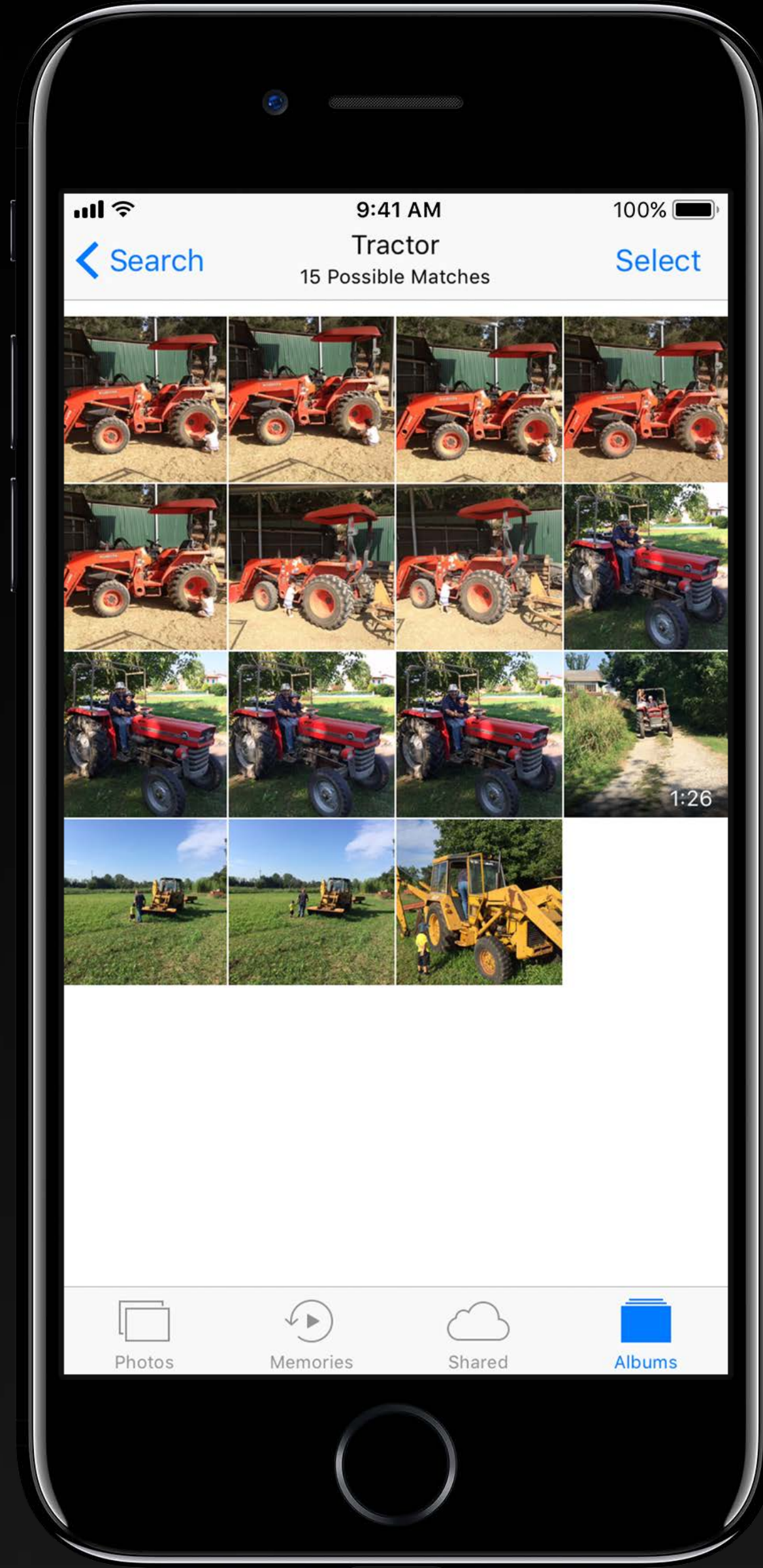
AVCapture

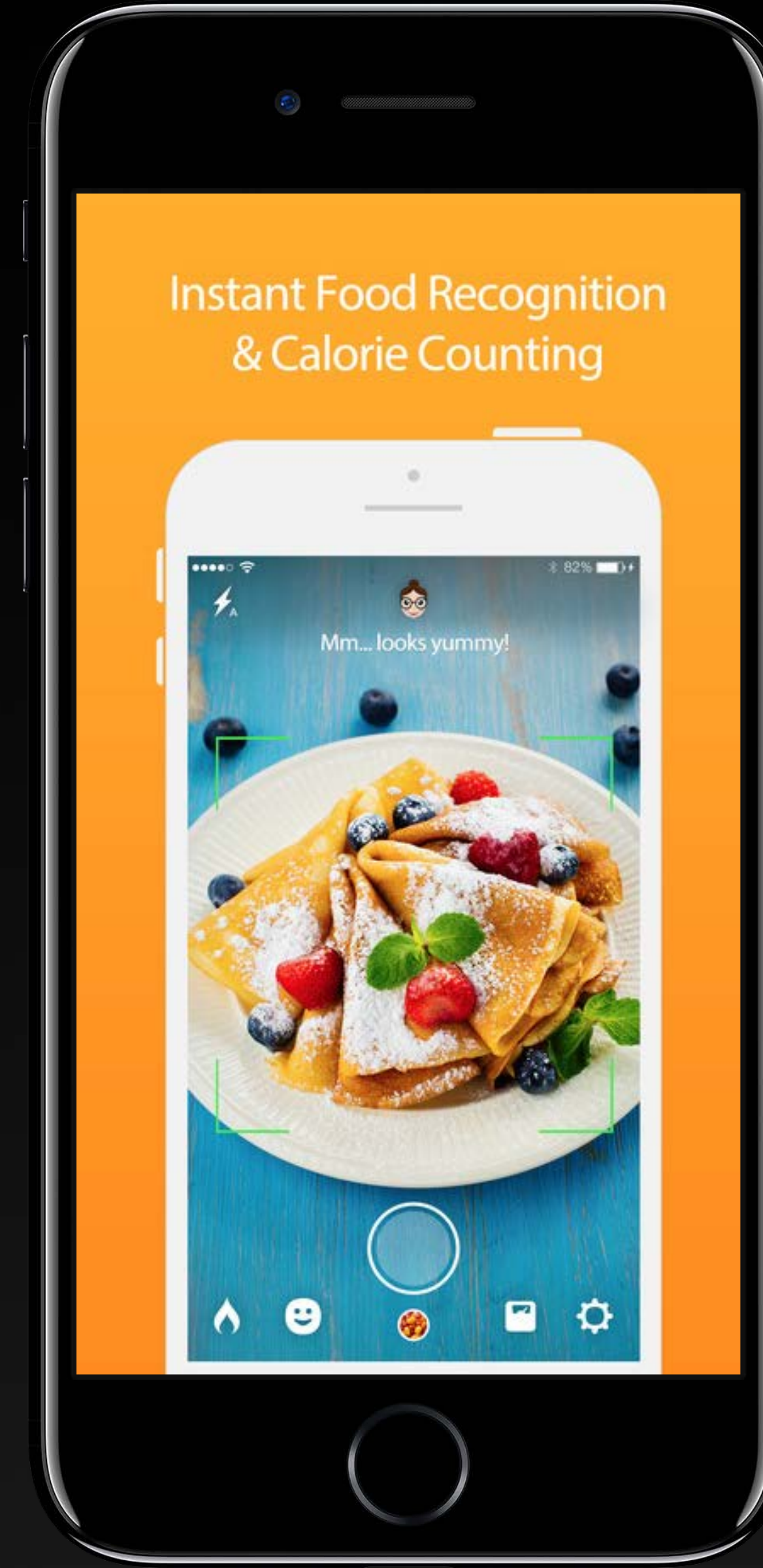
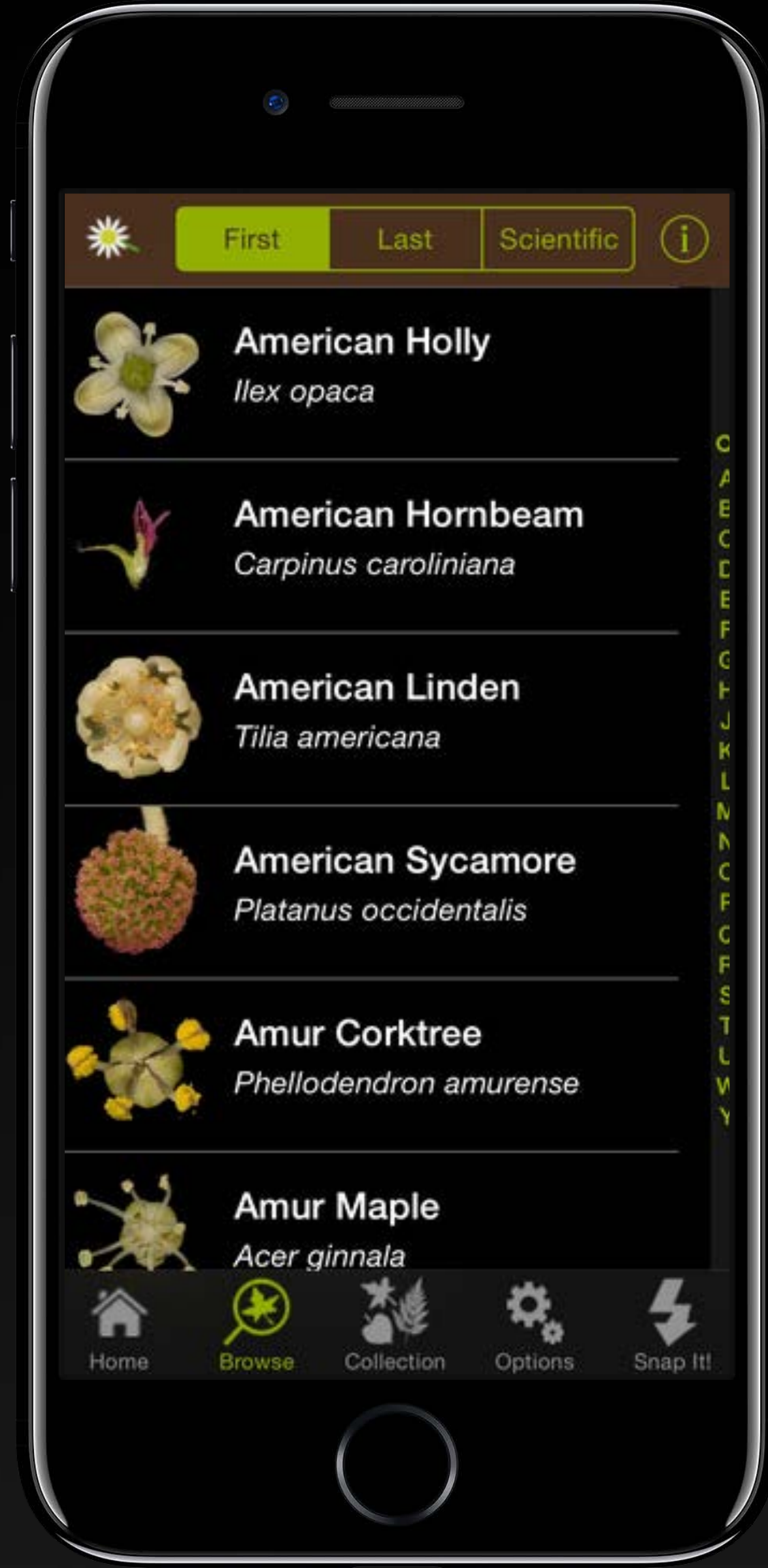
AVDepthData

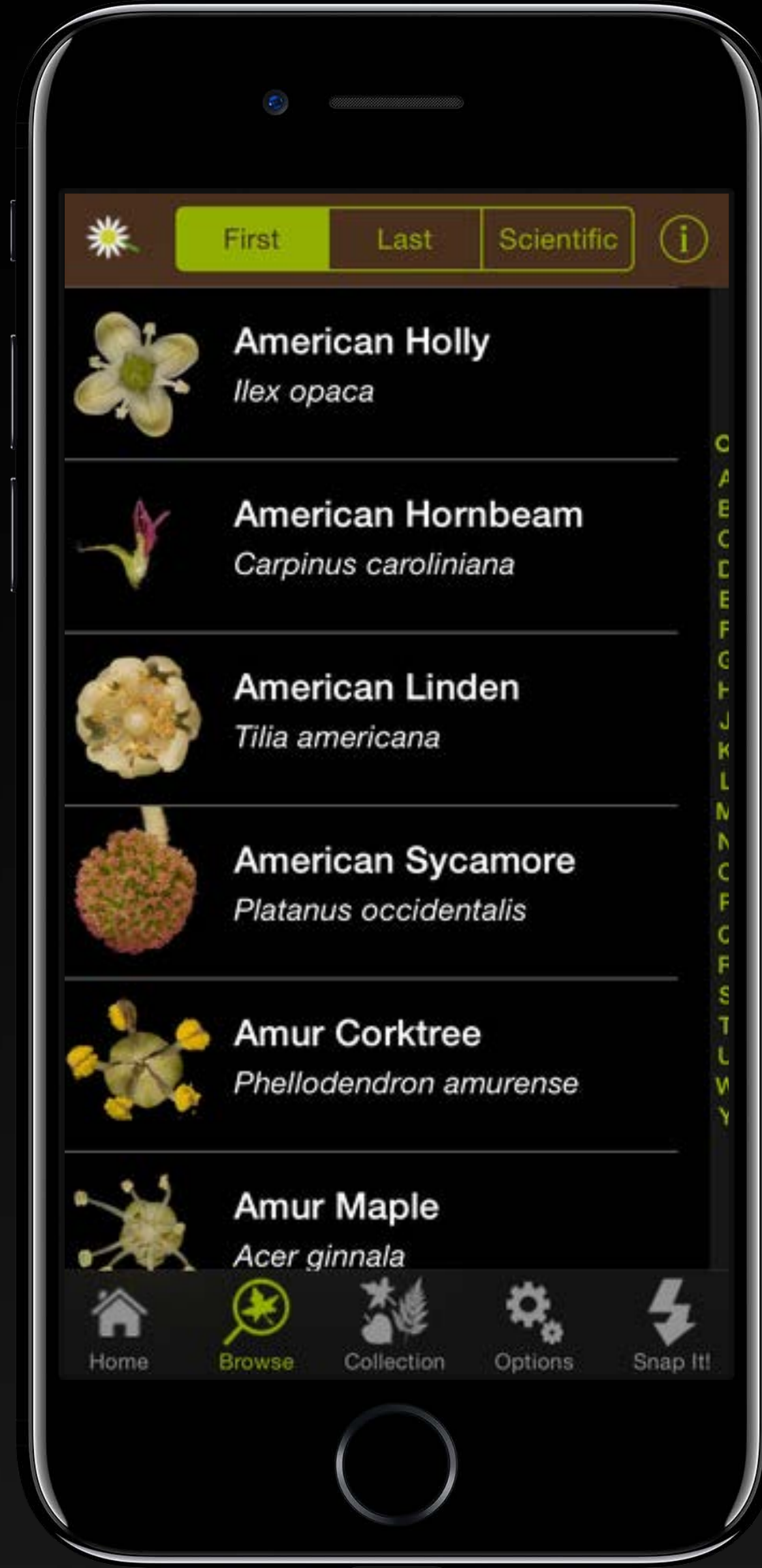
ImageIO

Core Image

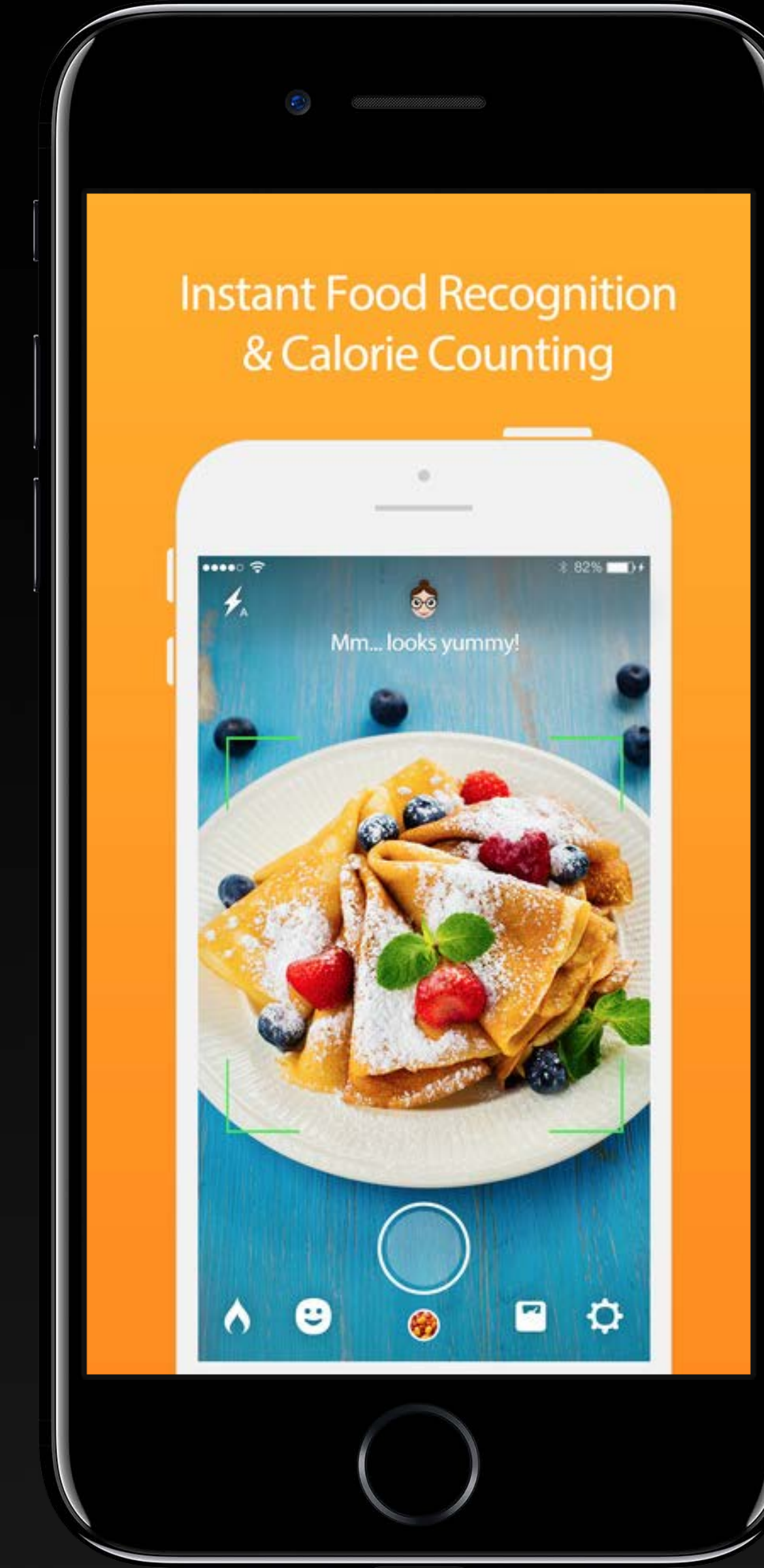
Vision



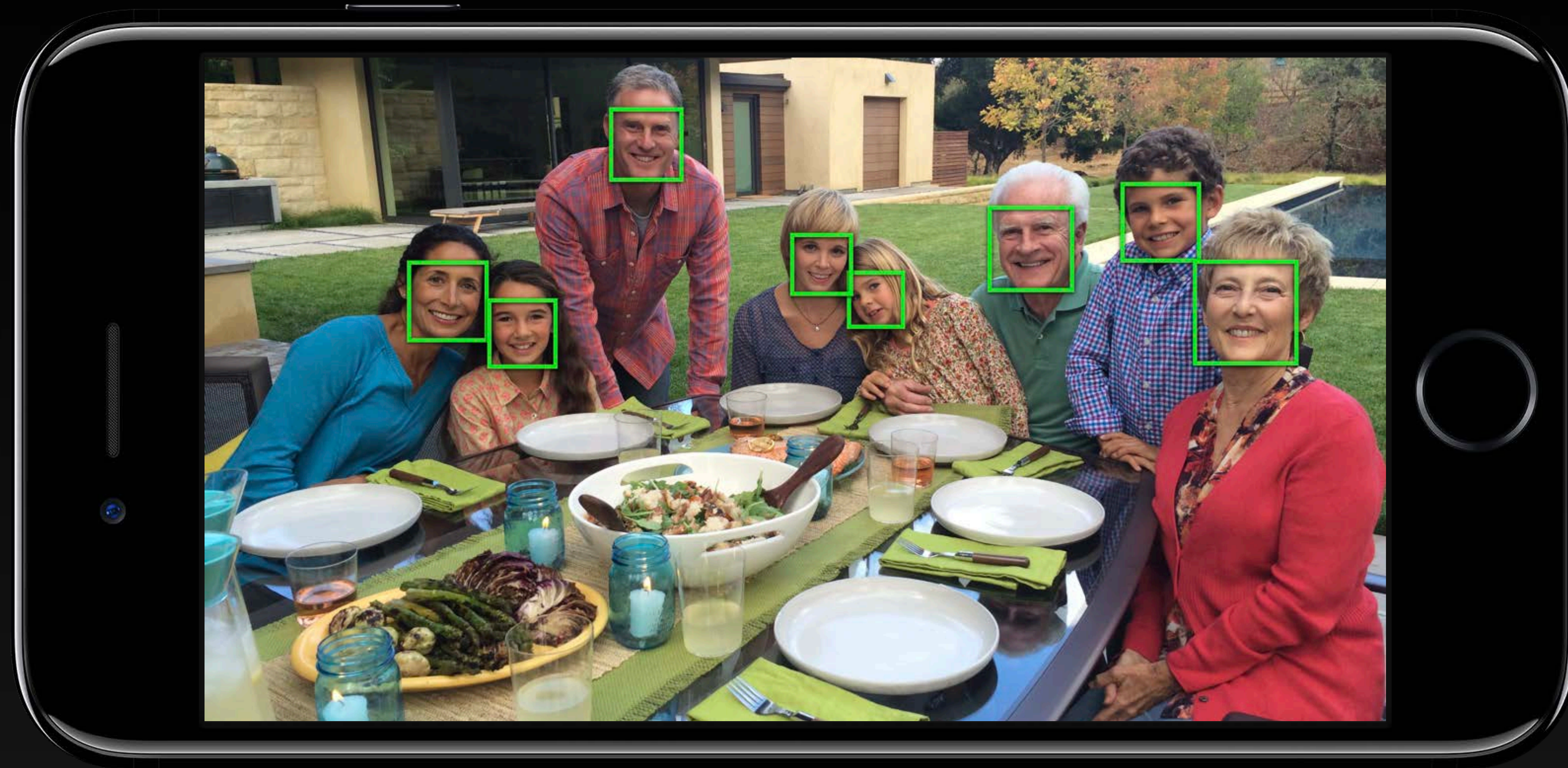




Bring powerful
computer
vision capabilities
to your apps



Face and Landmark Detection



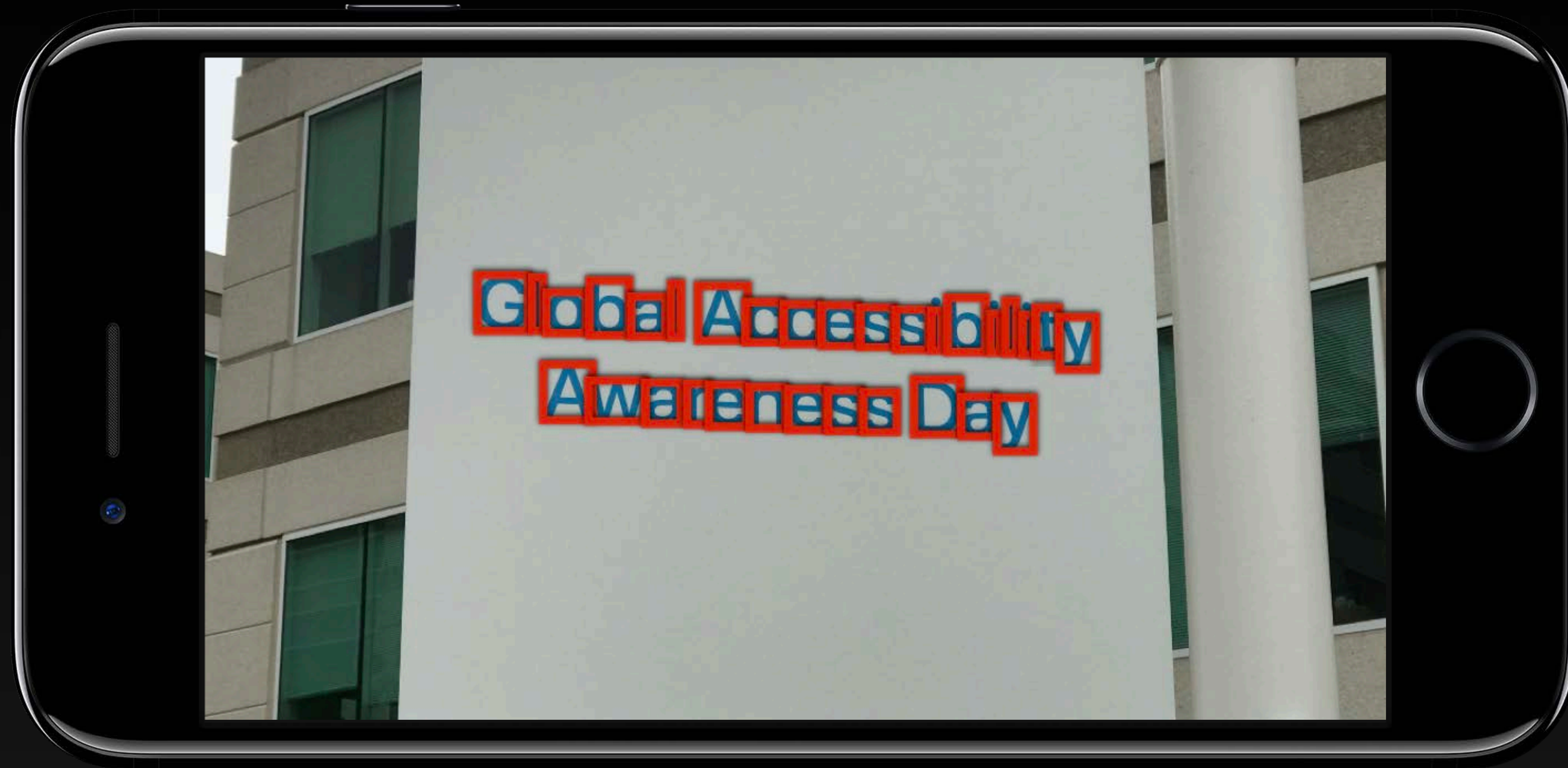
Rectangle Detection



Text Detection



Text Detection



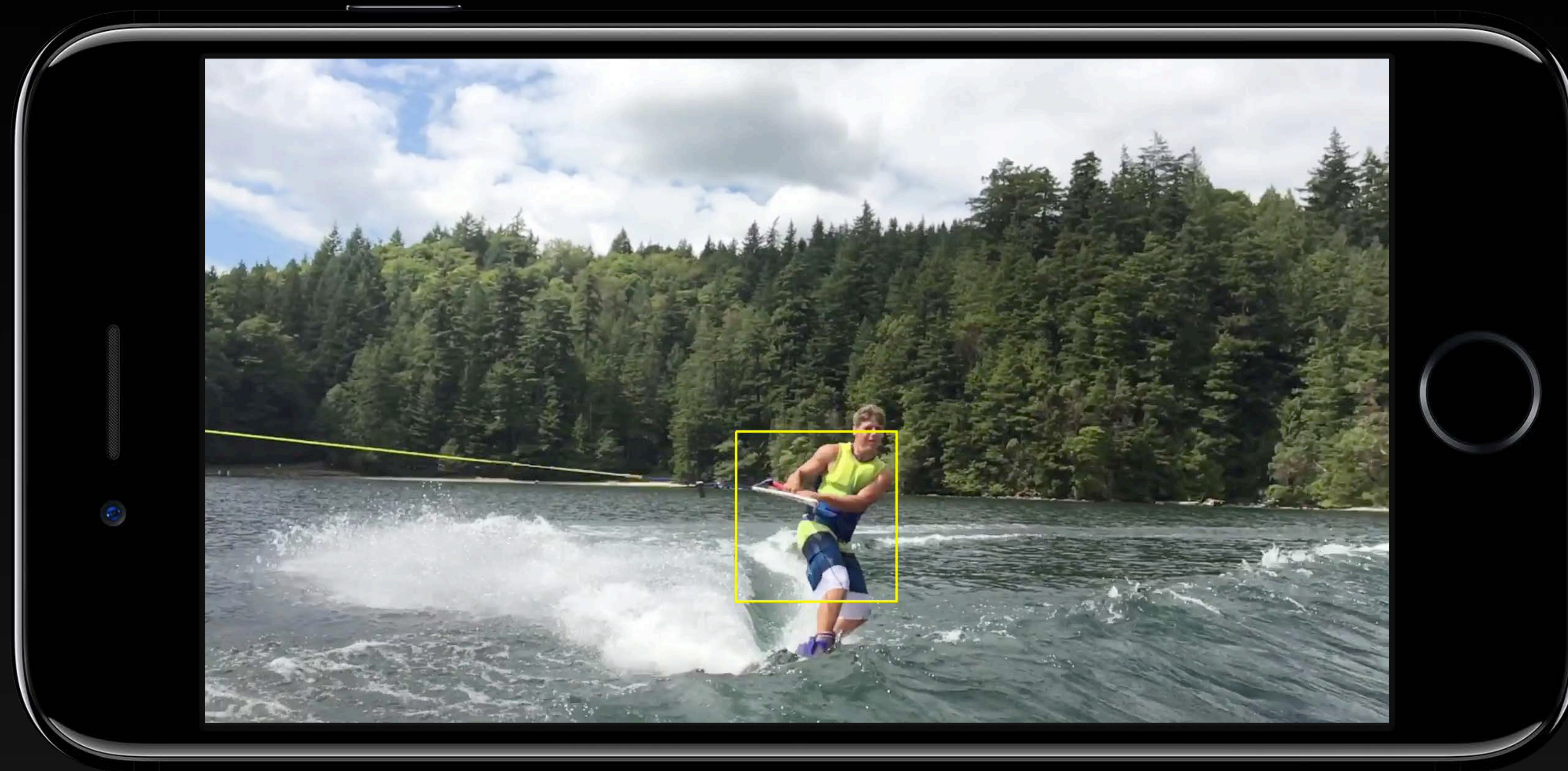
Barcode Detection



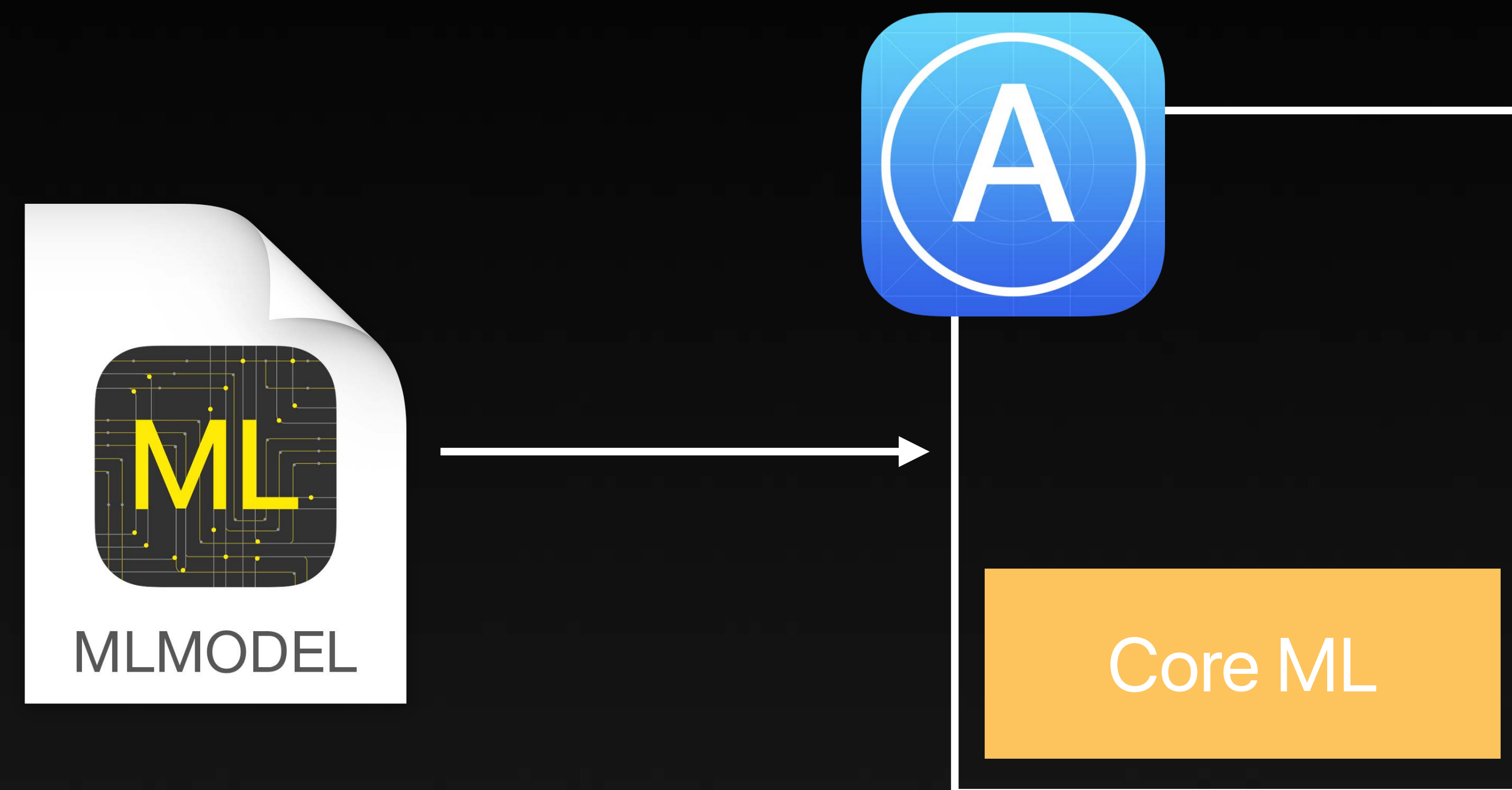
Object Tracking



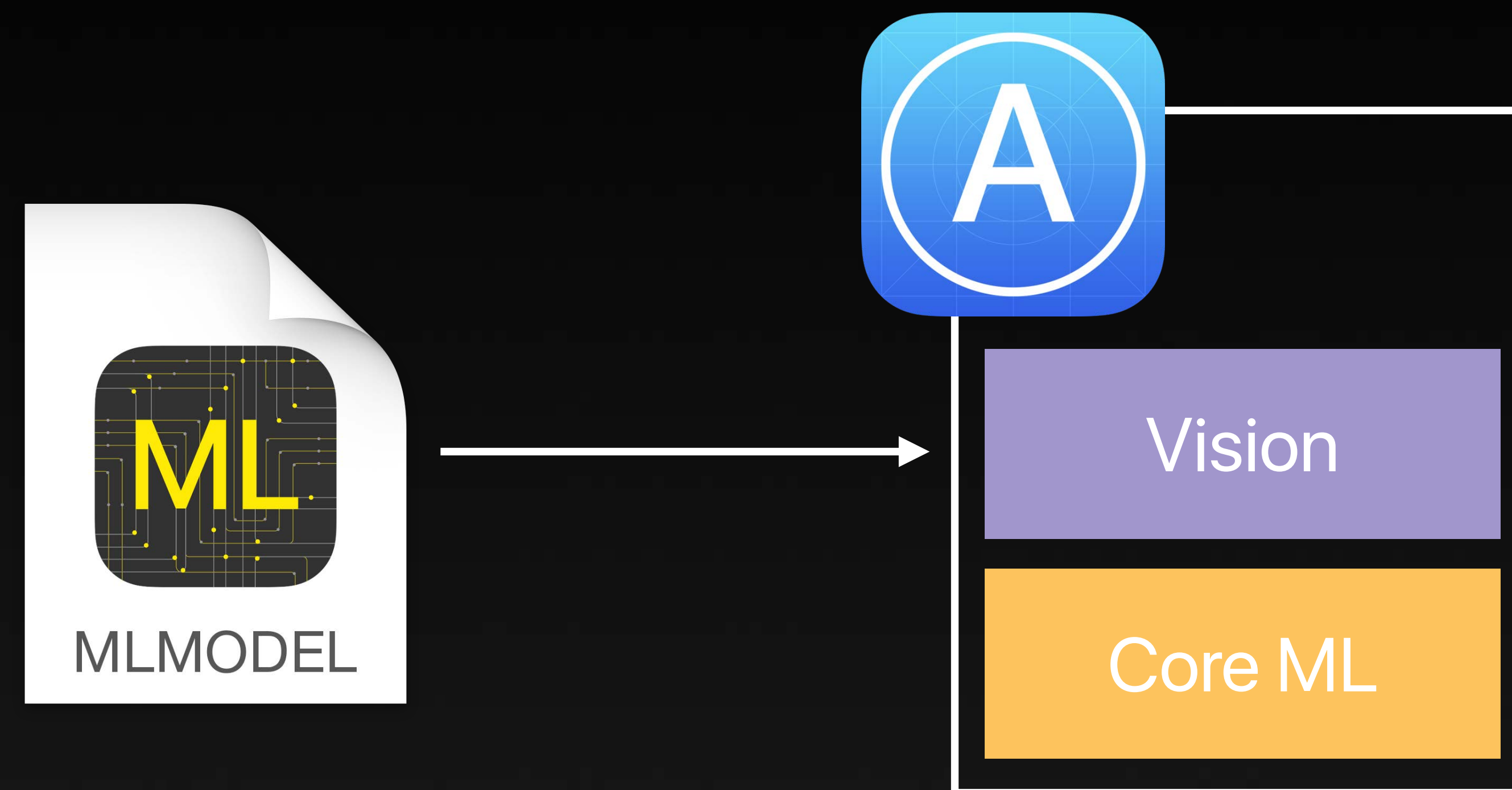
Object Tracking



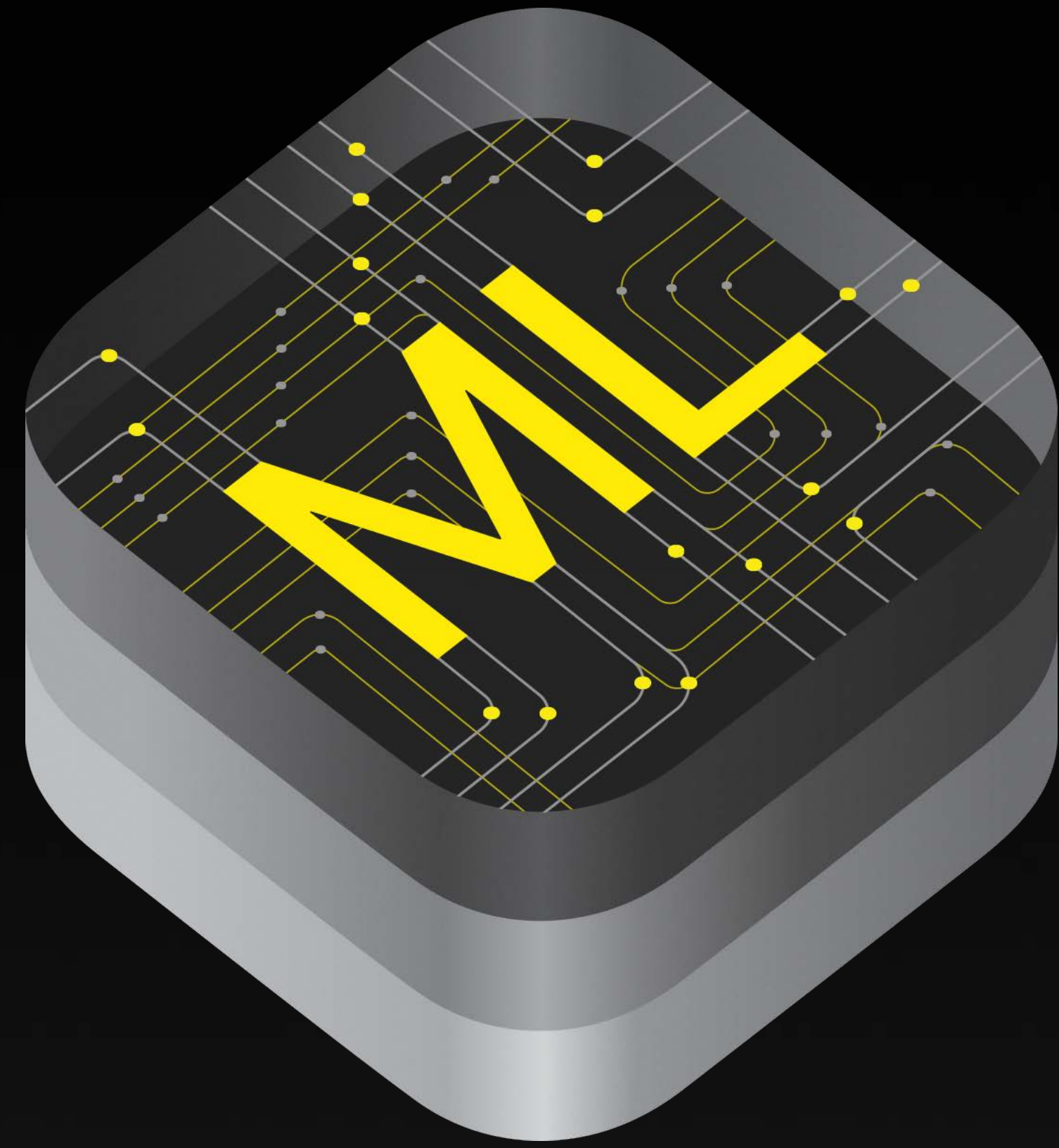
Vision and Core ML



Vision and Core ML



Demo

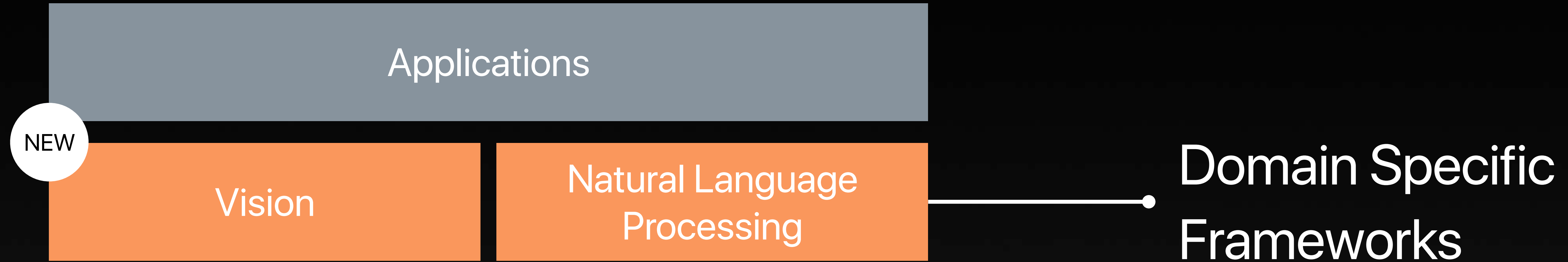


Core ML

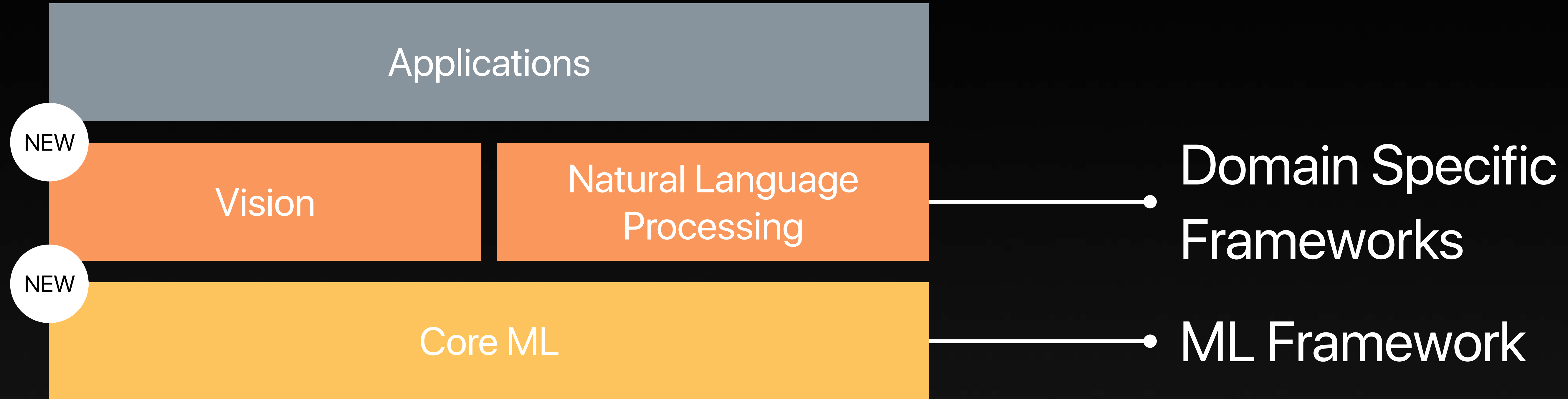
Architecture

Applications

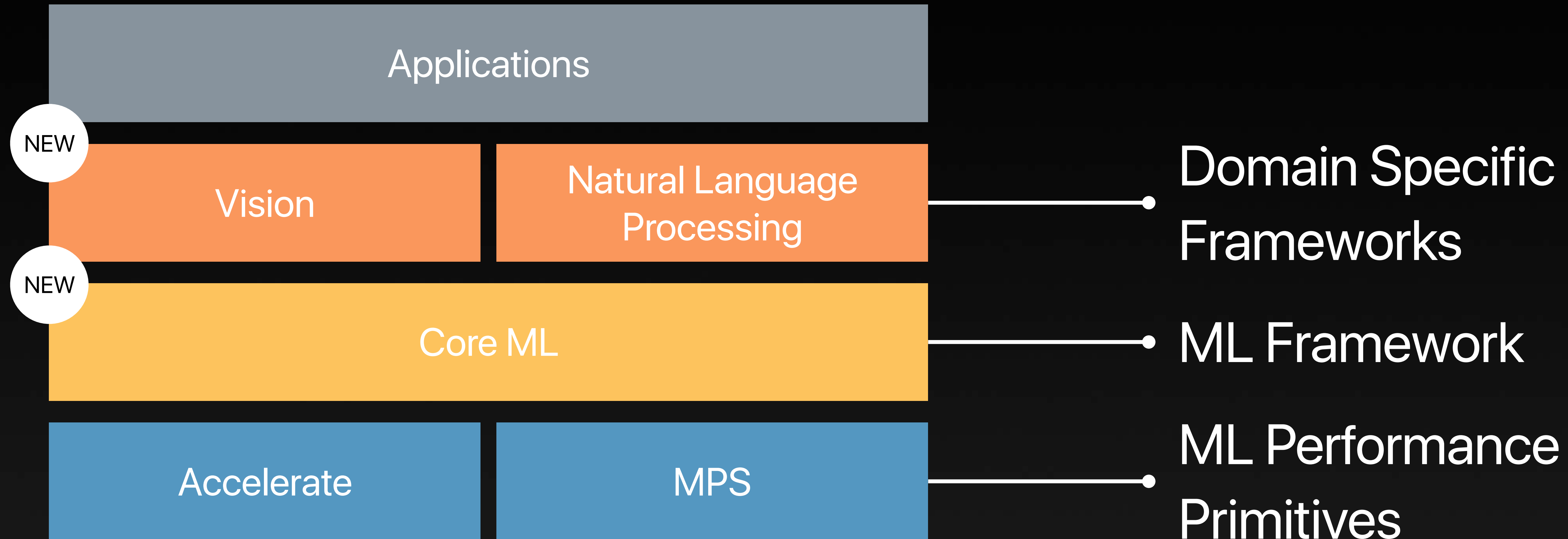
Architecture



Architecture



Architecture



Core ML

Feed Forward
Neural
Networks

Convolutional
Neural
Networks

Recurrent
Neural
Networks

Tree
Ensembles

Support
Vector
Machines

Generalized
Linear
Models

Caffe

 Keras

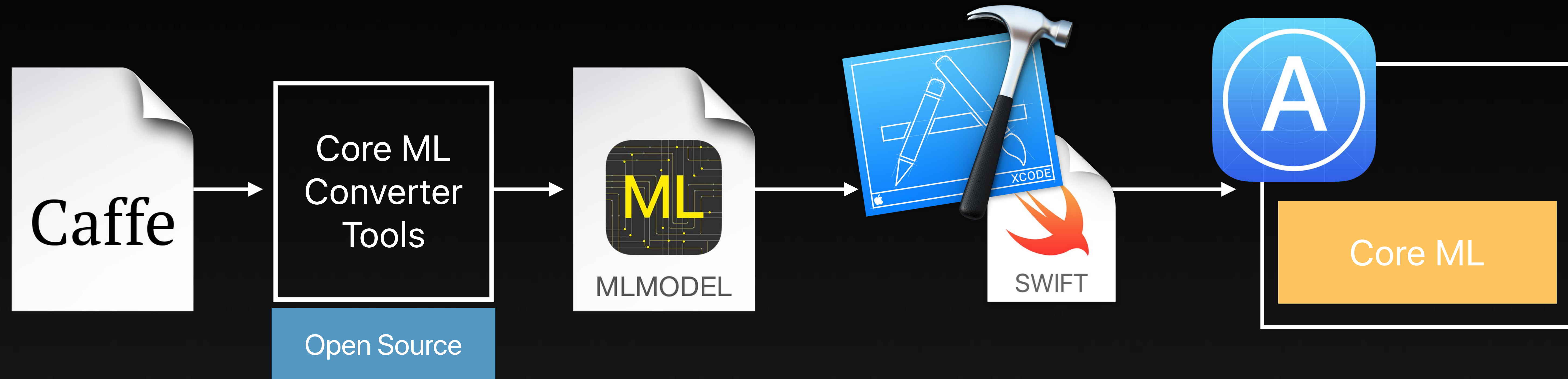
  scikit
learn

dmlc
XGBoost

turi 

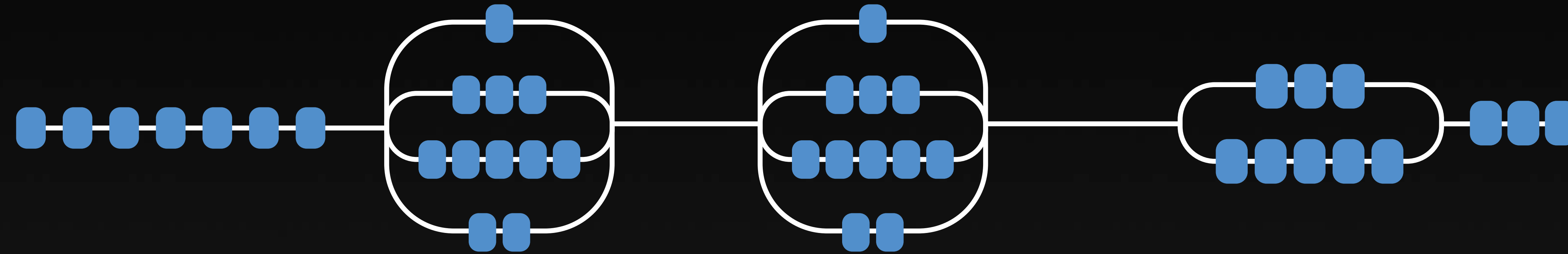
LIBSVM

Core ML Workflow



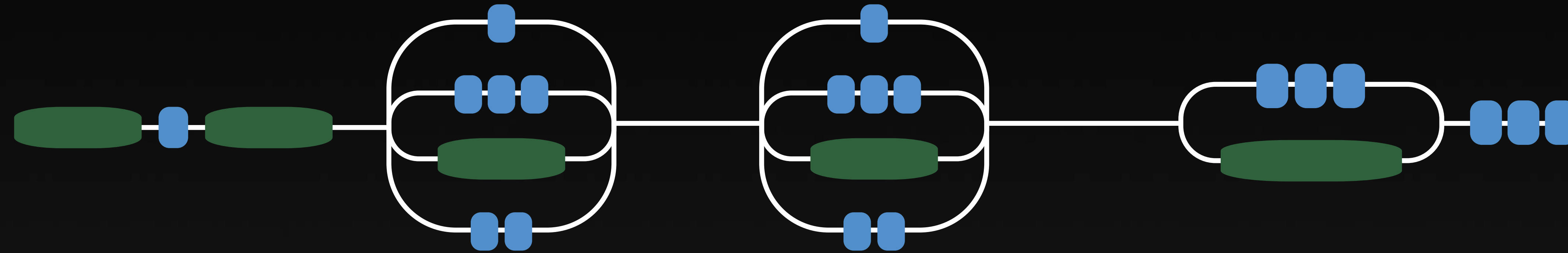
Performance

Neural
Network



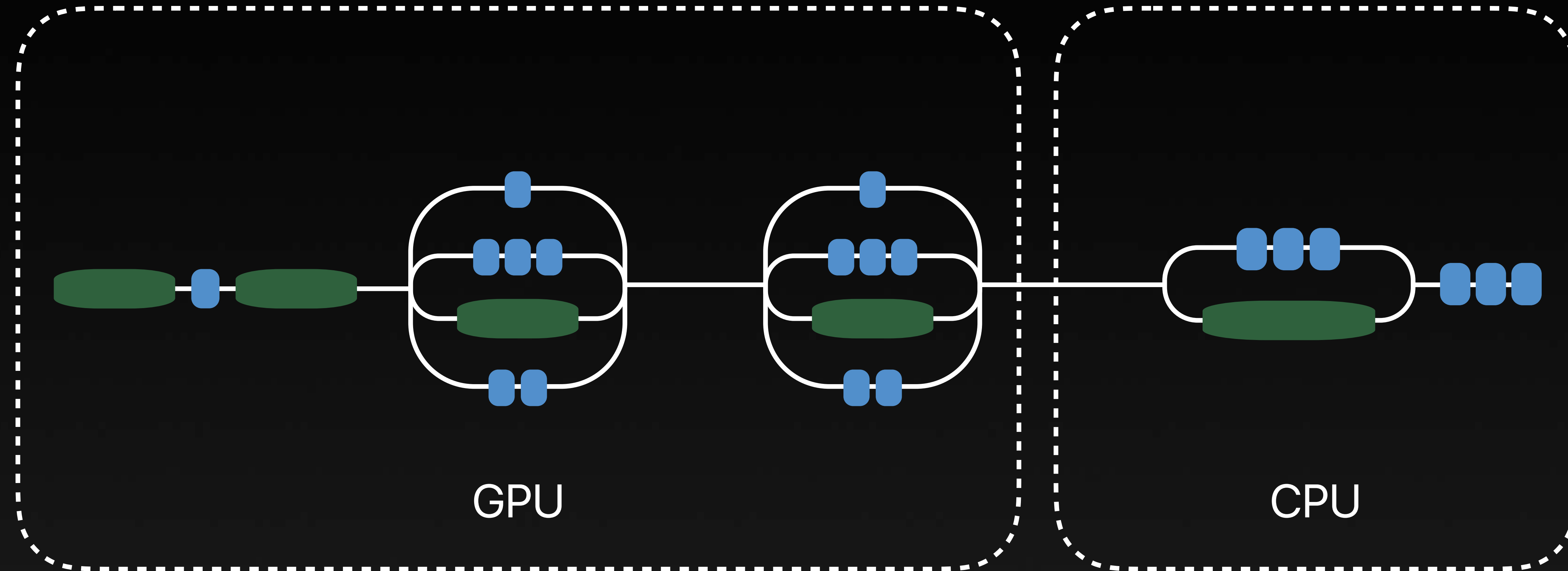
Performance

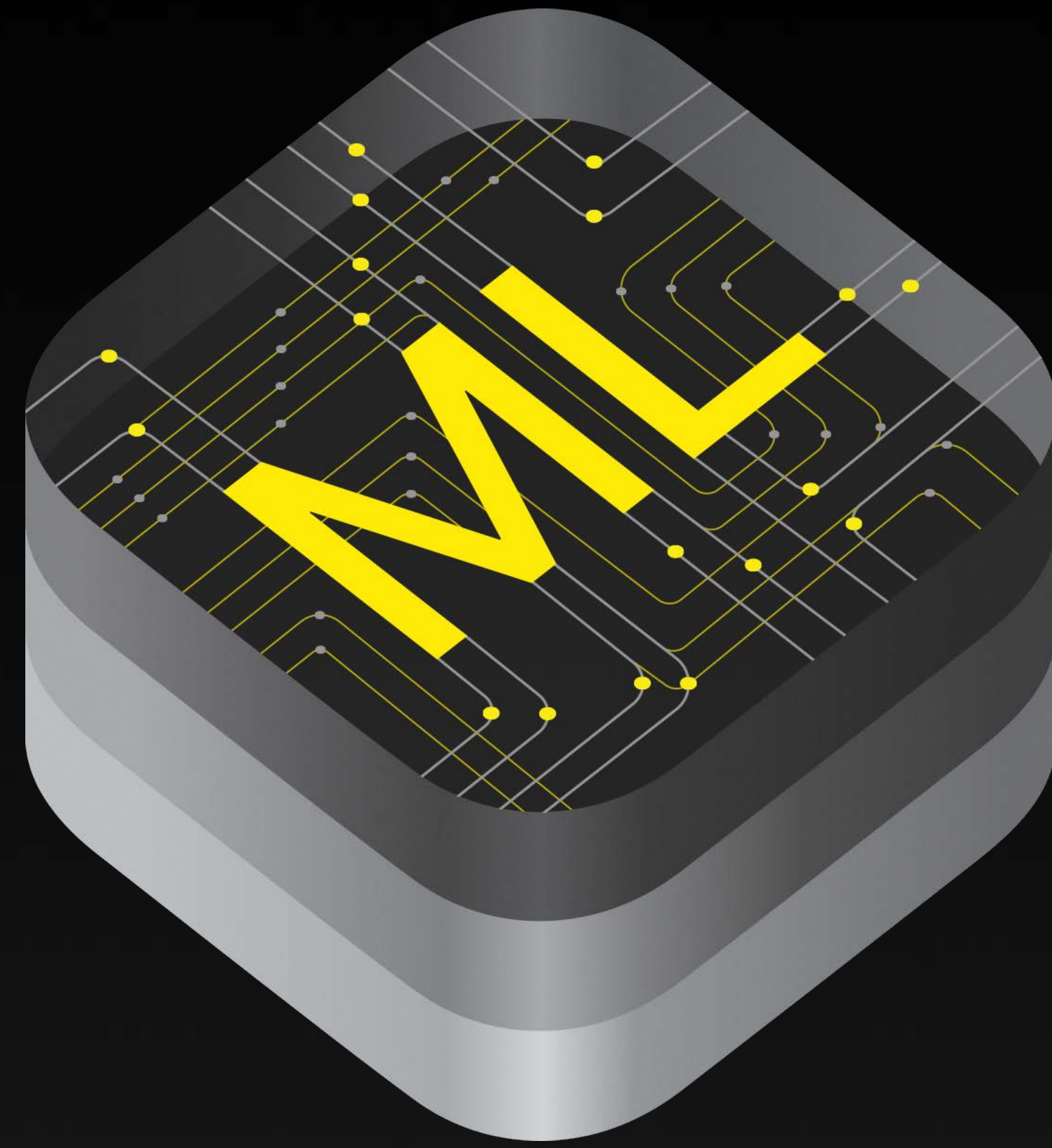
Graph
Optimization



Performance

Automatic
Hardware
Selection



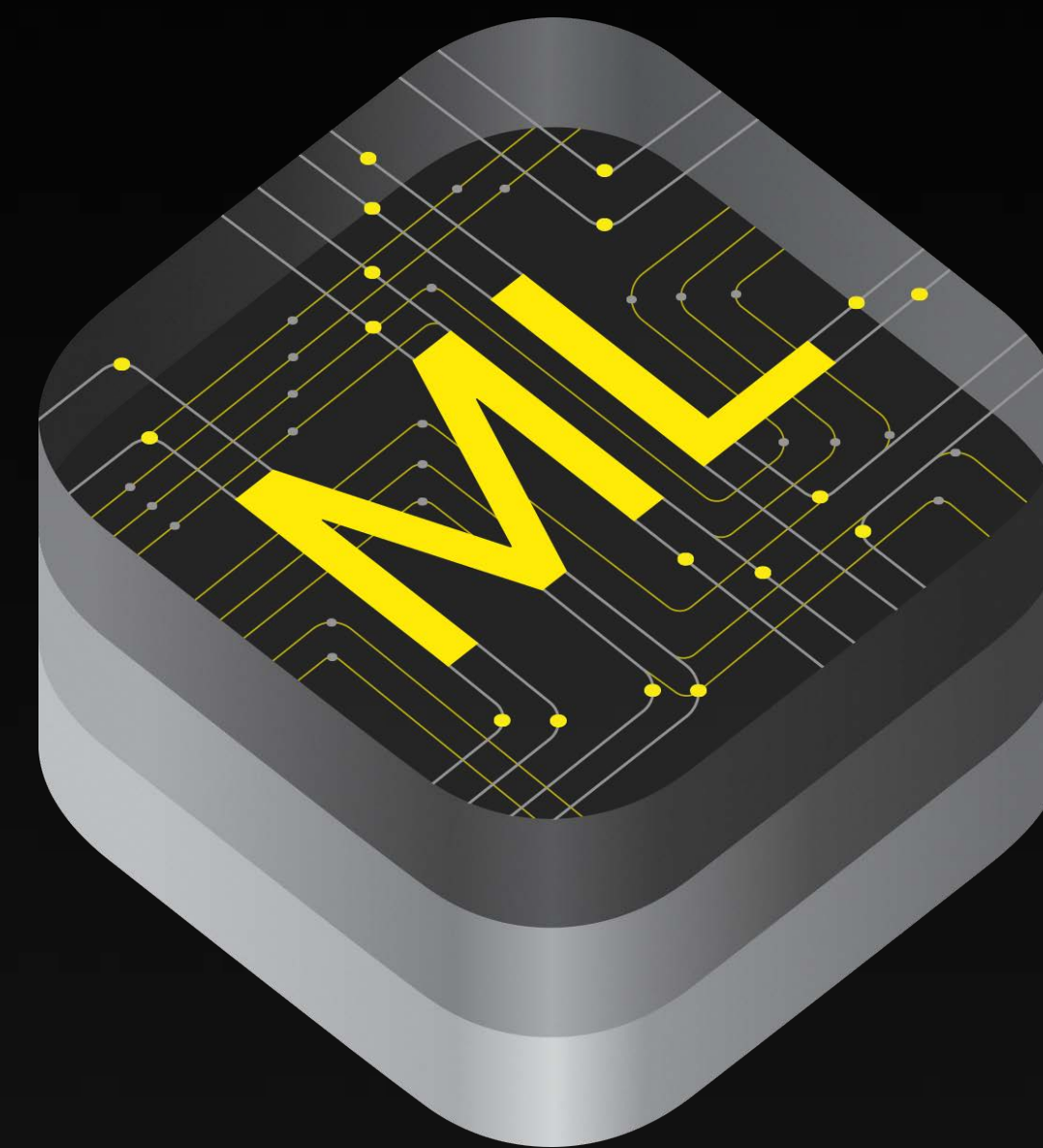


Simple

Fast

Energy efficient

iOS, macOS, watchOS, tvOS





Metal



Metal Performance Shaders Memoryless Render Targets C++14 shading language Stencil Texture Views

Separate textures and samplers Wide Color Efficient multithreading Shader profiler Indirect drawing/dispatch

Visual frame debugger Array of Textures

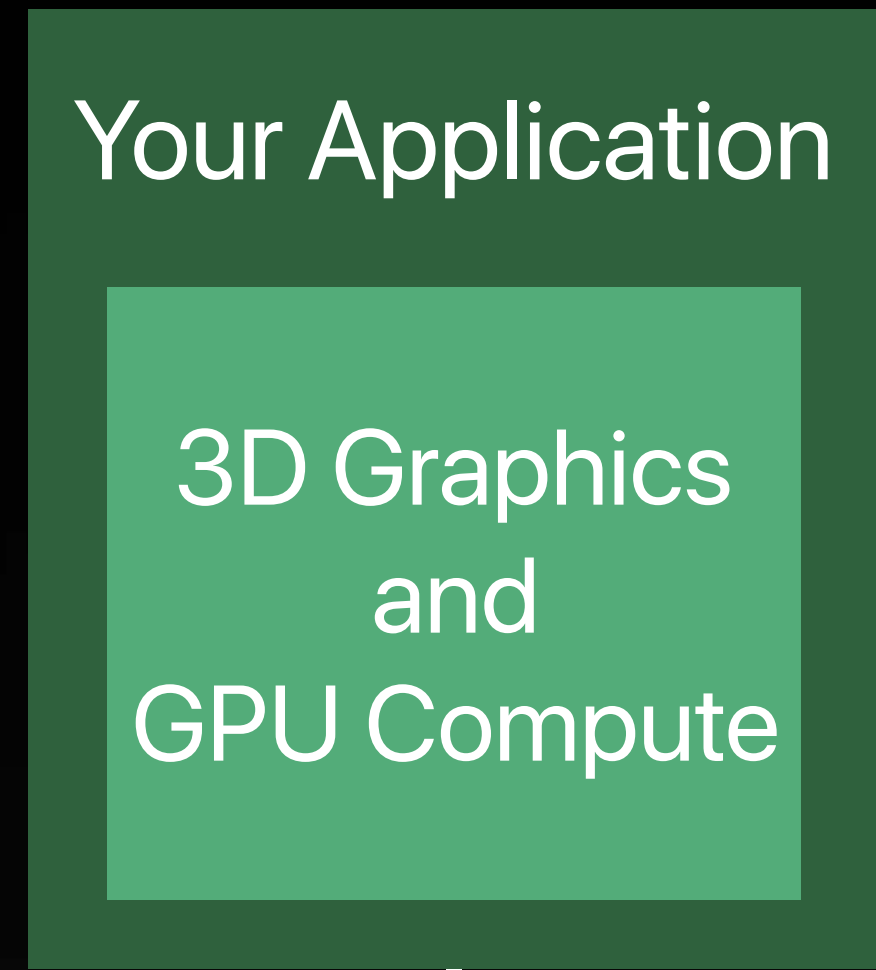
MSSA Blits Unified graphics and compute

Deferred Store Actions Parallel encoding

Threadgroup Synchronization MetalKit Dramatically reduced overhead sRGB Writes Convolution Neural Nets

Function Specialization Metal Trace Systems Persistent buffer mappings Function Resource Read-Writes





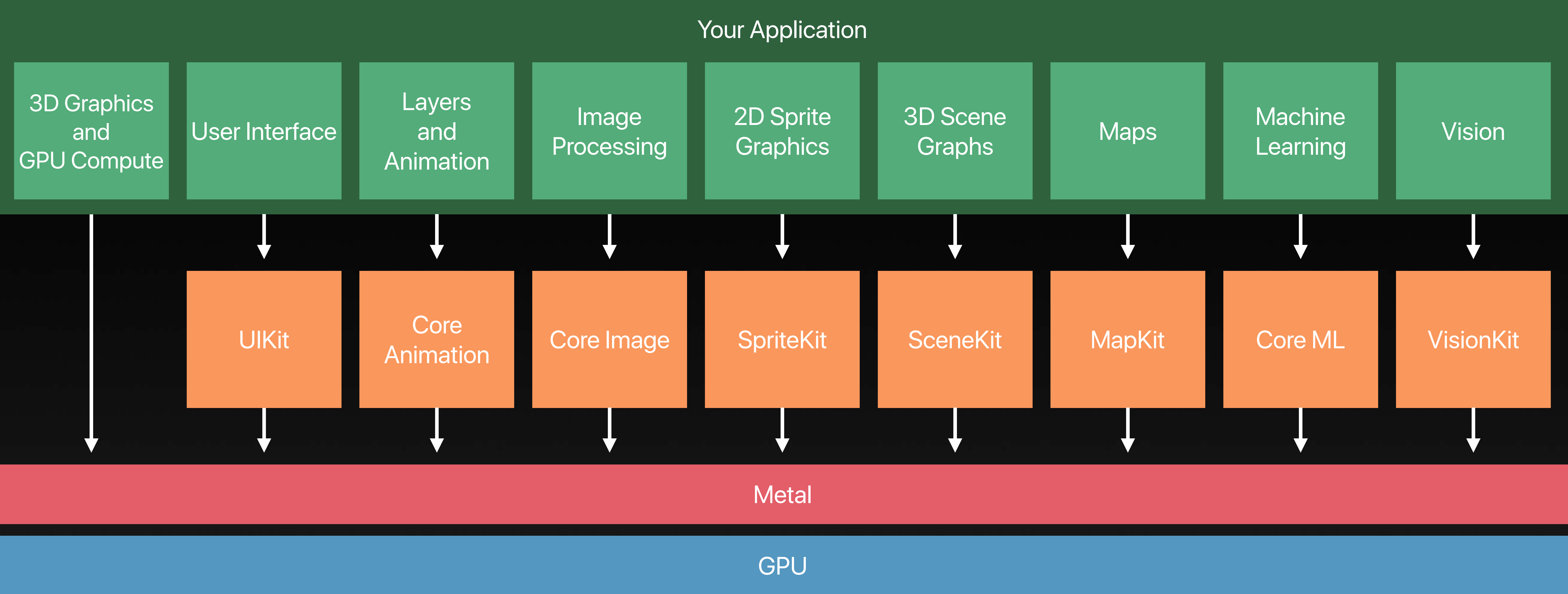
Your Application

3D Graphics
and
GPU Compute



Metal

GPU



1.7 Million

iOS apps using system frameworks calling Metal



Colour



Colour Wheel



Opacity 100 %



Noise 0 %



Recent Colours



Swatches

379 px
Width

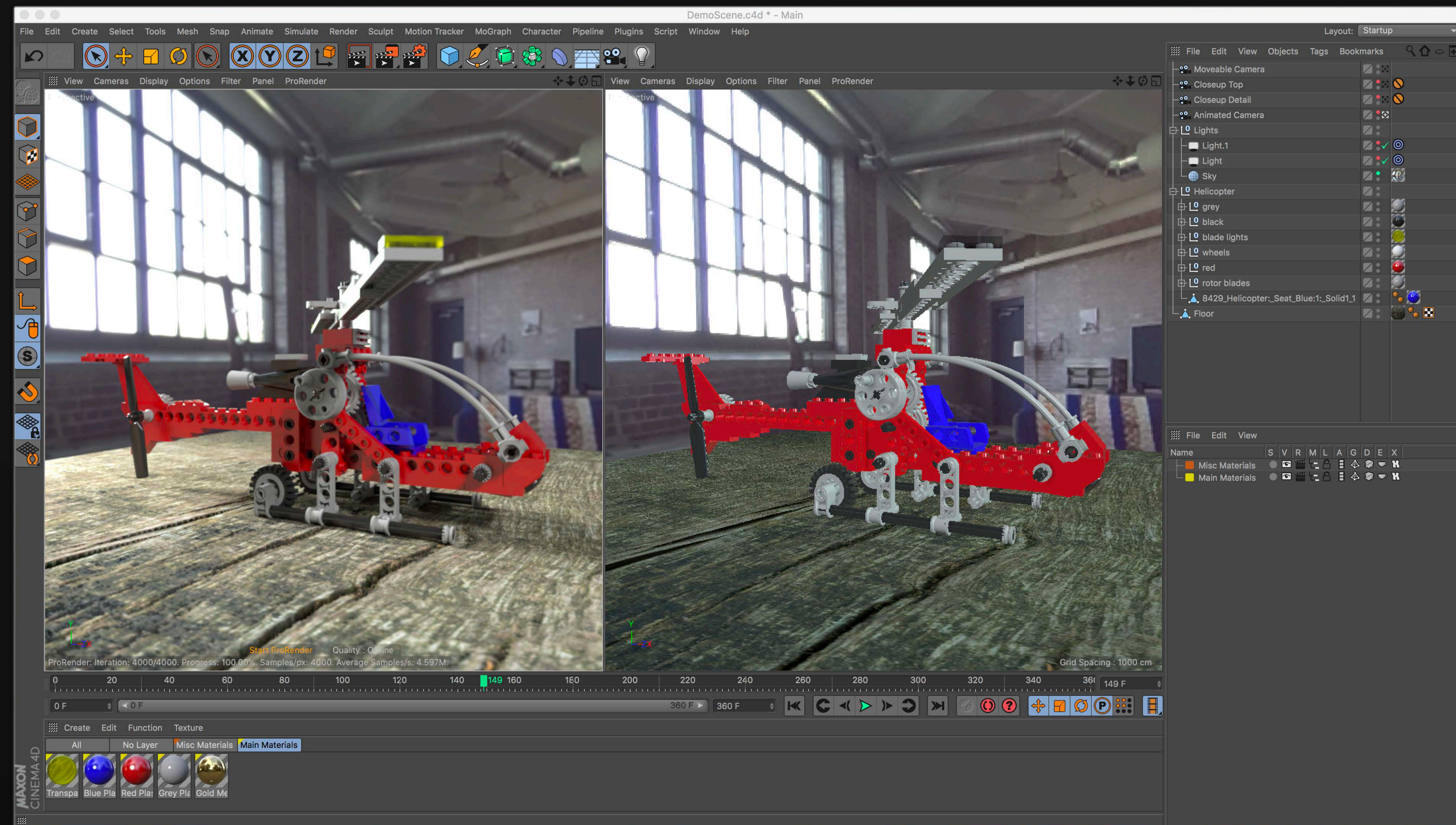
71 %
Opacity

67 %
Flow

16 %
Hardness

More

Colour



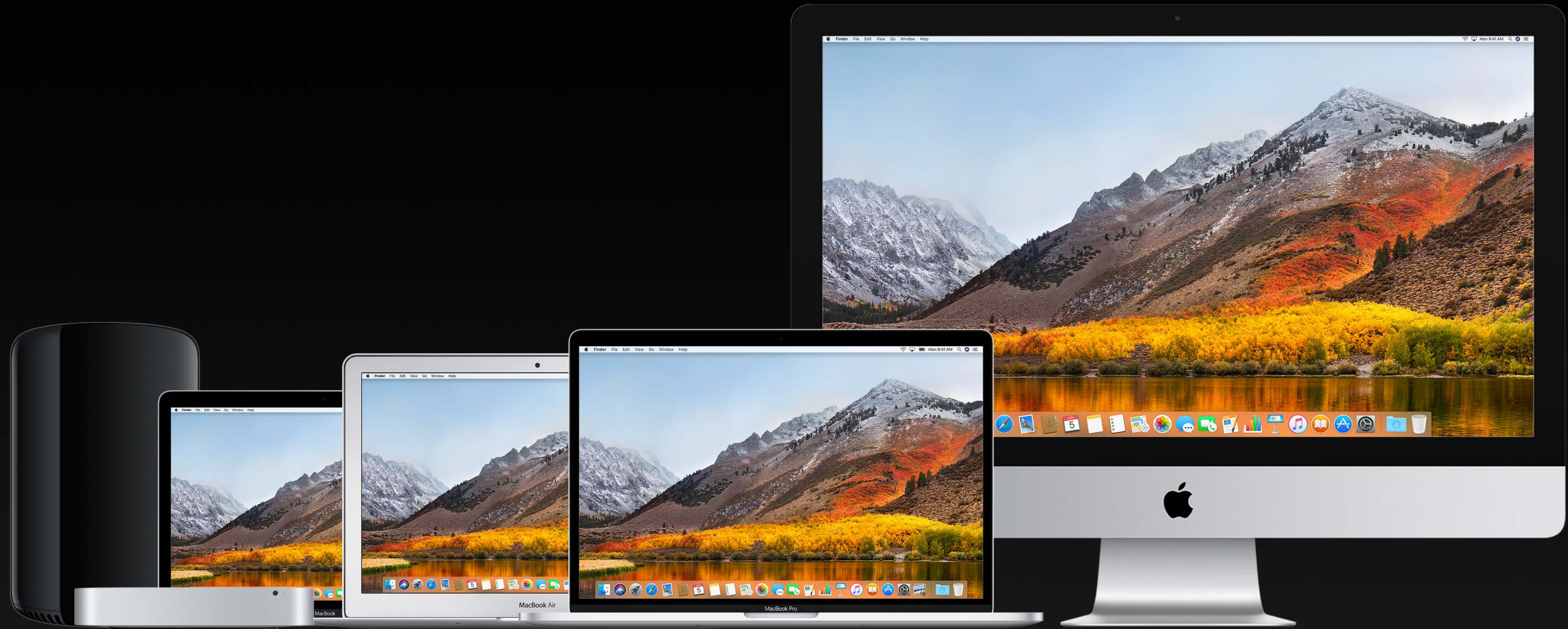




“Metal’s richer feature set and lower overhead have allowed us to bring cutting edge games to the Mac with frame rates and effects that simply weren’t possible before.”

Ian Bullock, Head of Technology
Feral Interactive





900 Million

Apple products supporting Metal



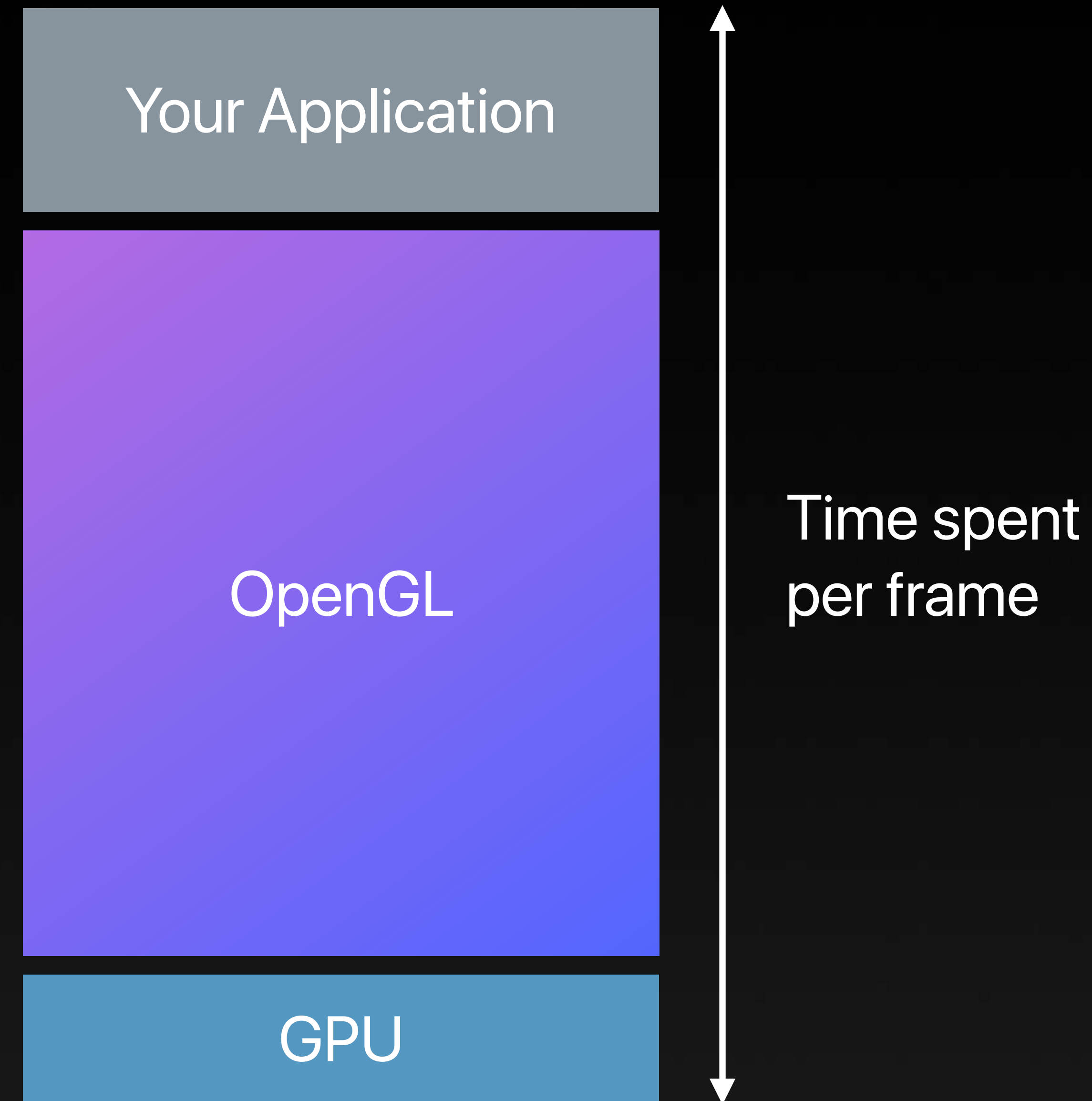
Metal 2



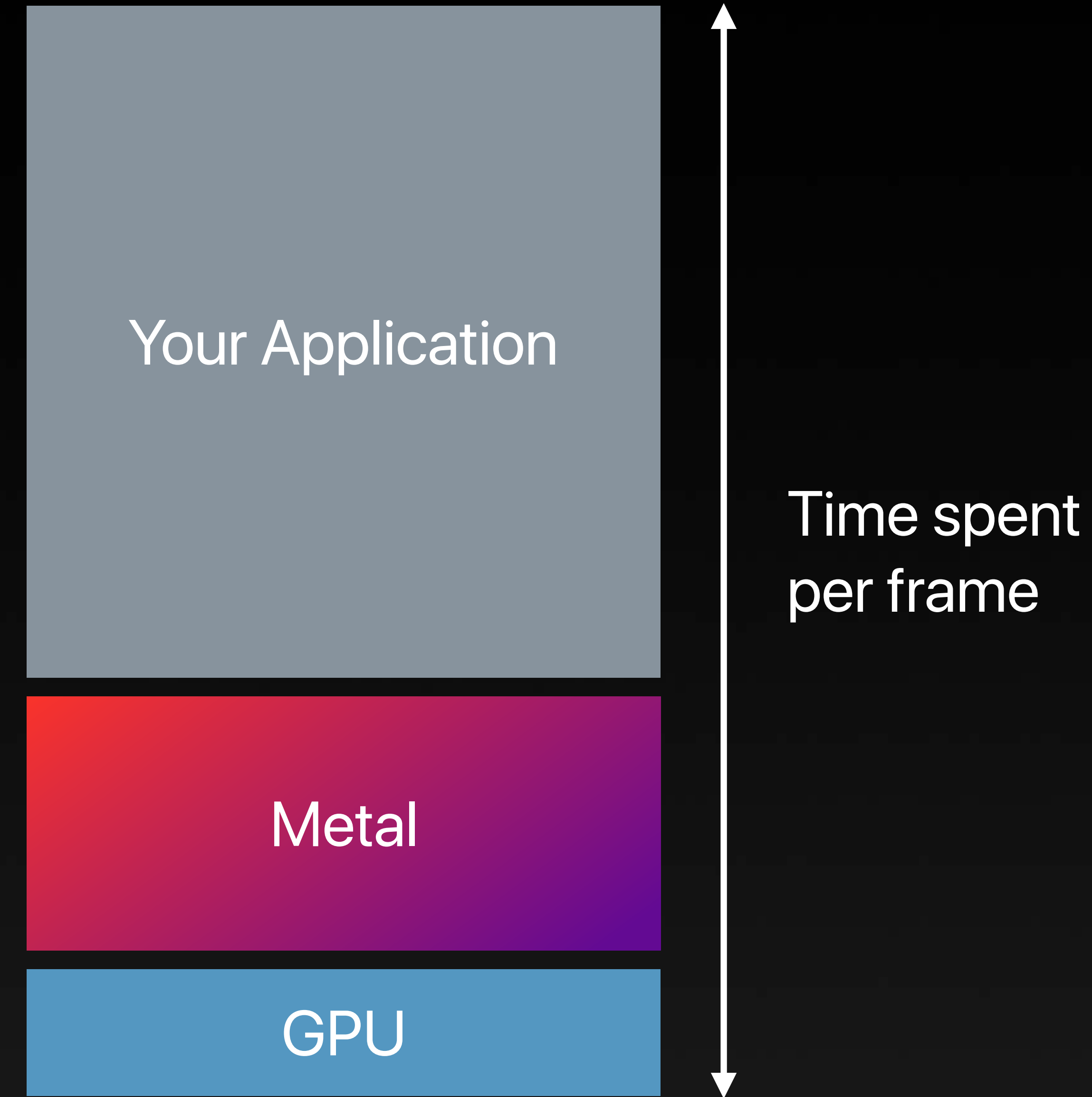


GPU-Driven Rendering

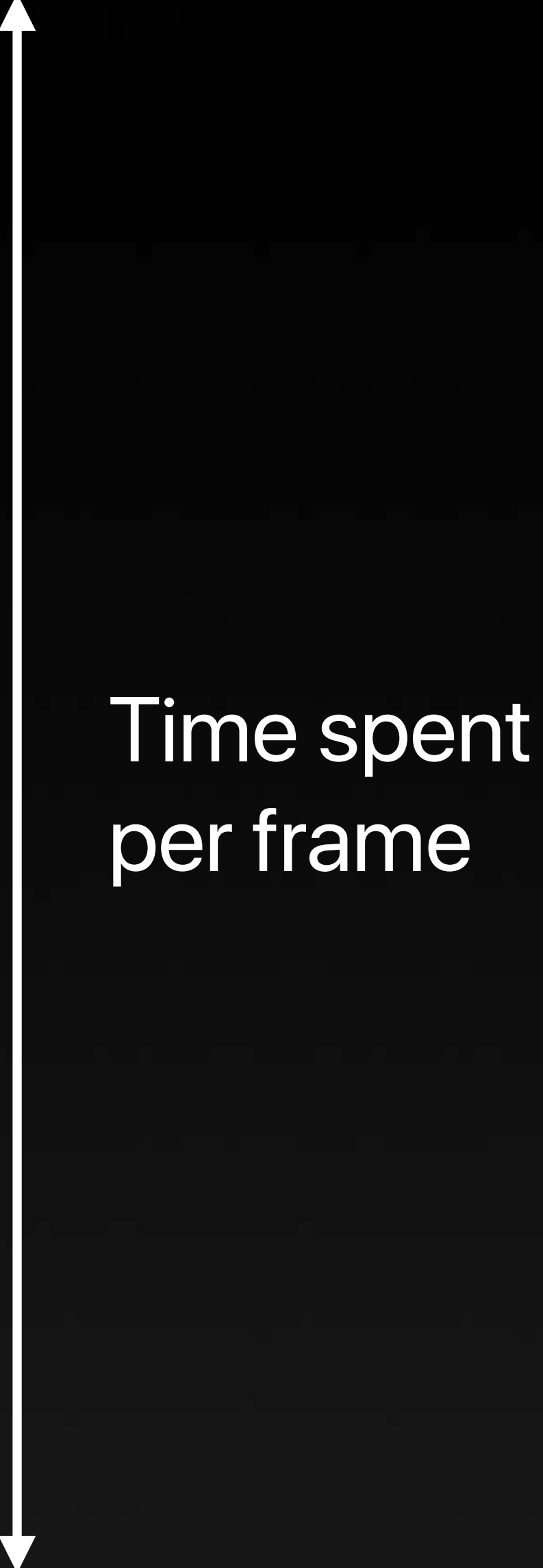
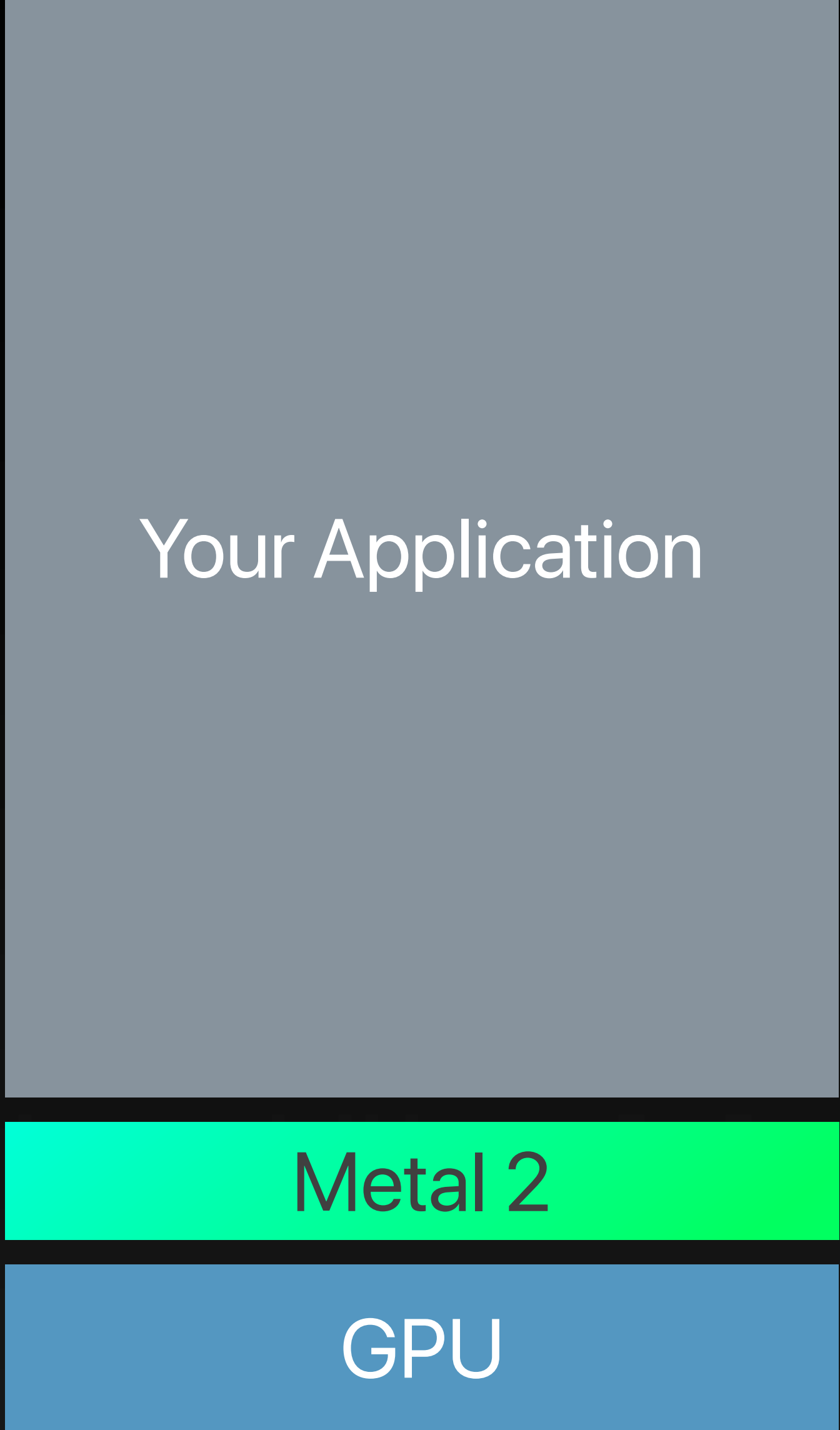
Significant Overhead Per Frame



Dramatically Reduced CPU Costs



Even More Efficiency





Indirect argument buffers

Raster order groups

Quad and SIMDGroup data exchange

Arrays of samplers

Uniform variables

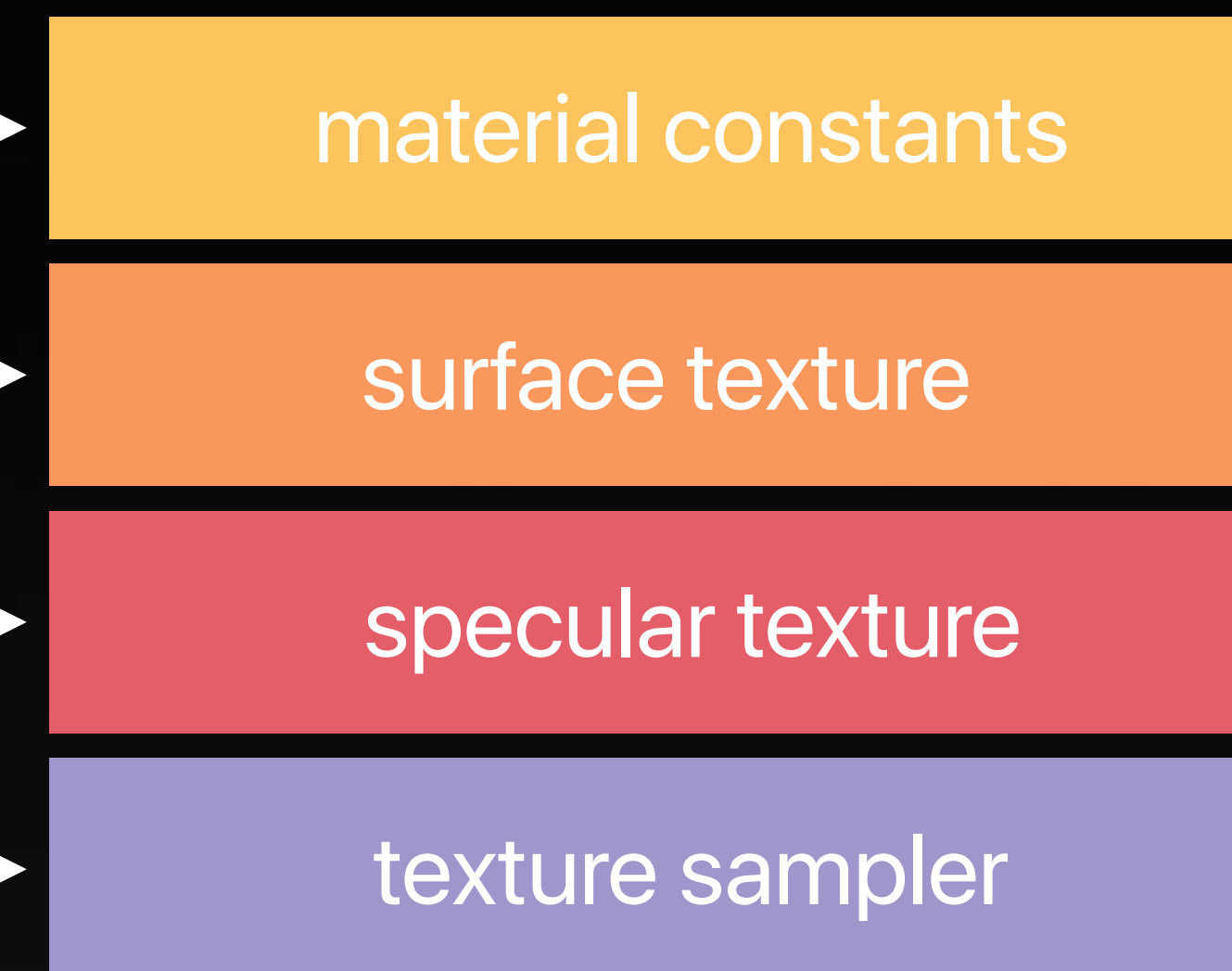
Metal Render Pass



Metal Render Pass



Metal Resources



Setting all arguments every draw call

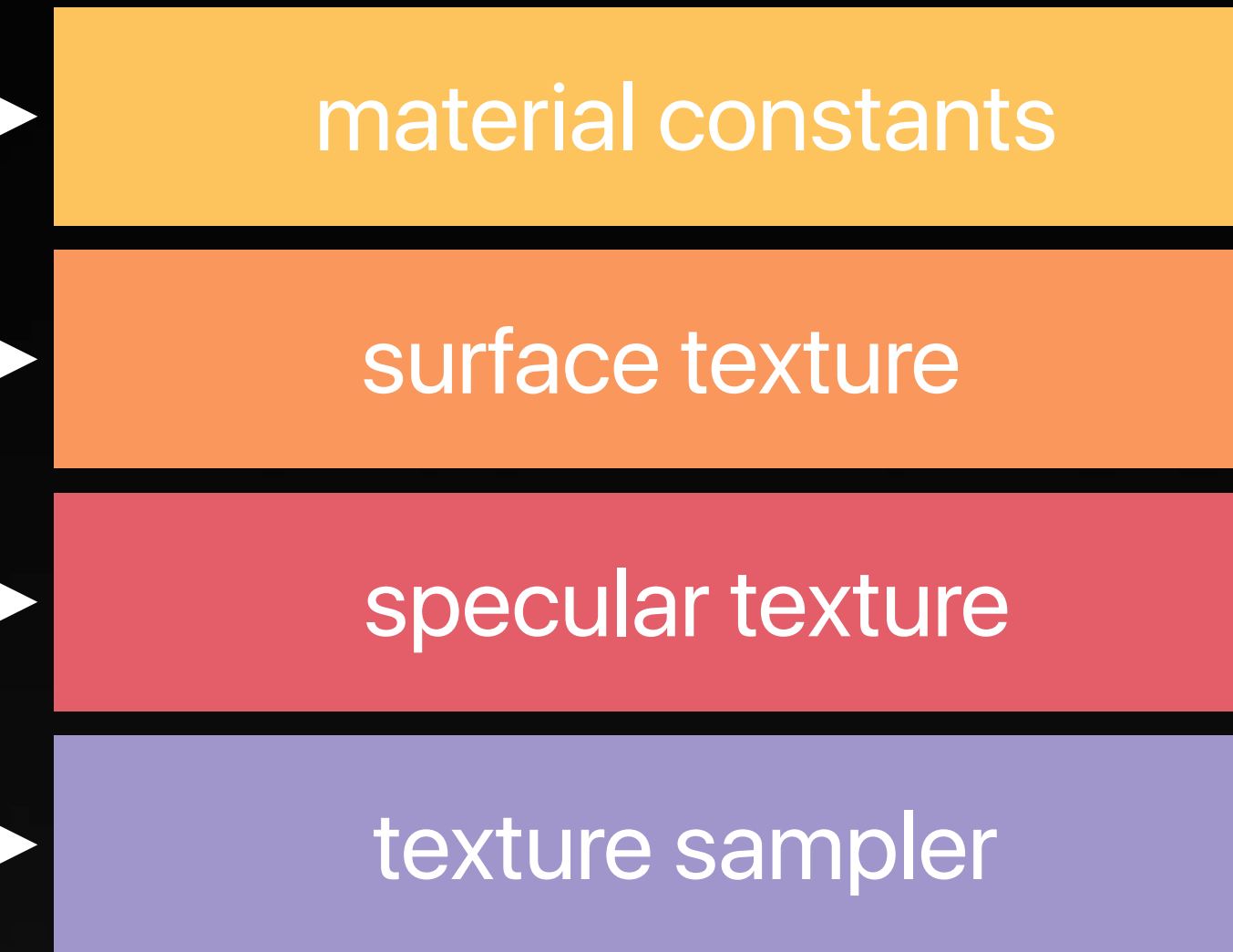
Metal Render Pass



Argument Buffer



Metal Resources



Metal Render Pass



Set argument buffer per draw call

Metal Resources



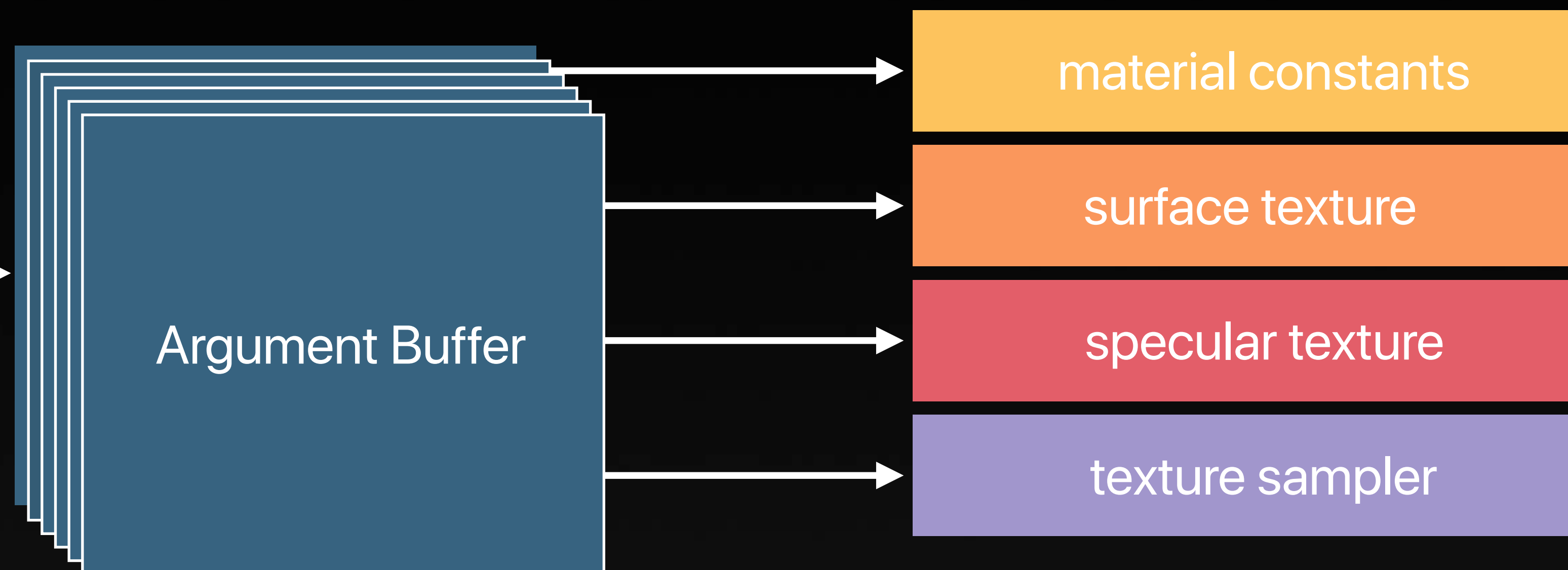
Initialize argument buffers once during set up

Metal Render Pass



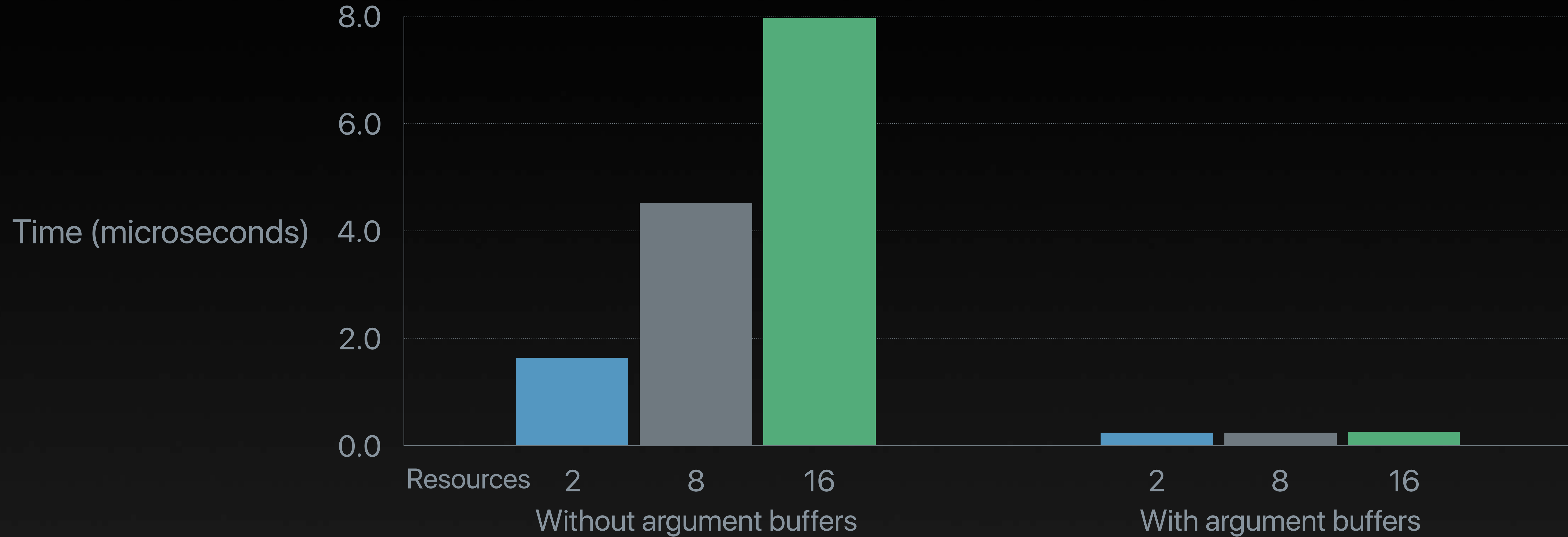
Set argument buffer per draw call

Metal Resources

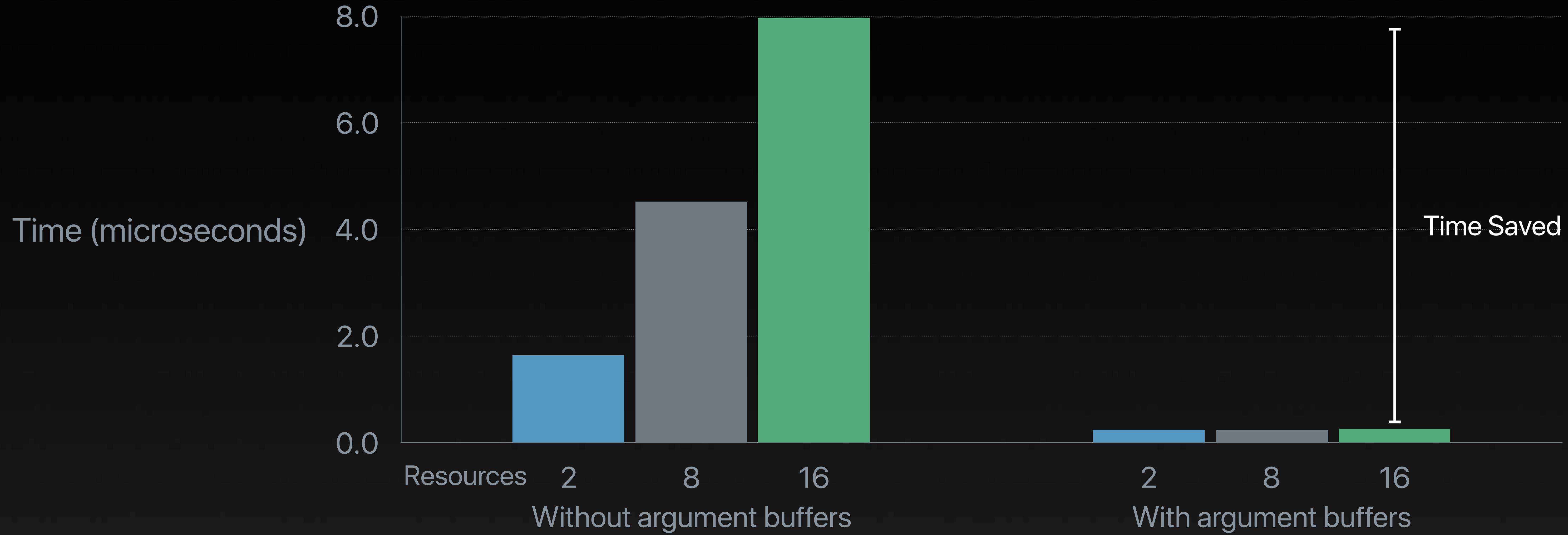


Initialize argument buffers once during set up

CPU Time Per Draw Call



CPU Time Per Draw Call





Platform Feature Alignment



Unified API across macOS, iOS, and tvOS
Maintains compatibility wherever possible
Exposes key hardware features



Metal Performance Shaders

Resource Heaps

Linear Textures

Dual Source Blending





Allocate large buffers of memory

Ultra fast resource re-allocation

Abstract GPU hardware differences

Building block for MPS performance

Separate Allocations

Separate Allocations

Memory Allocation for A

Texture A

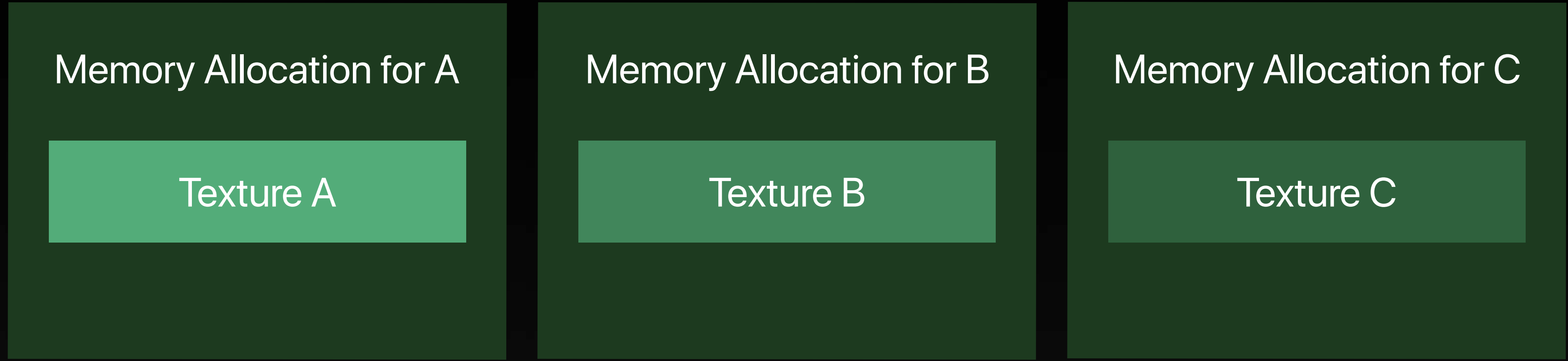
Memory Allocation for B

Texture B

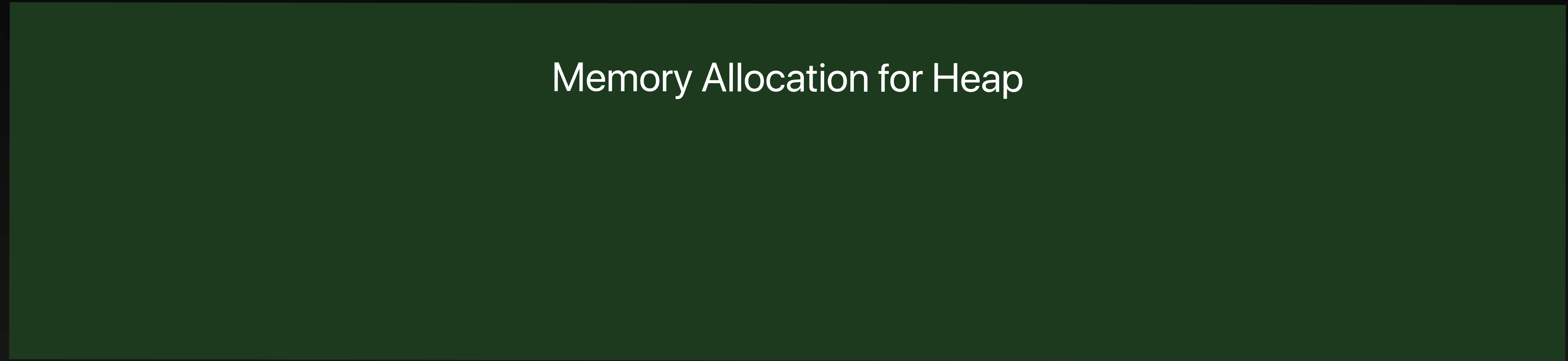
Memory Allocation for C

Texture C

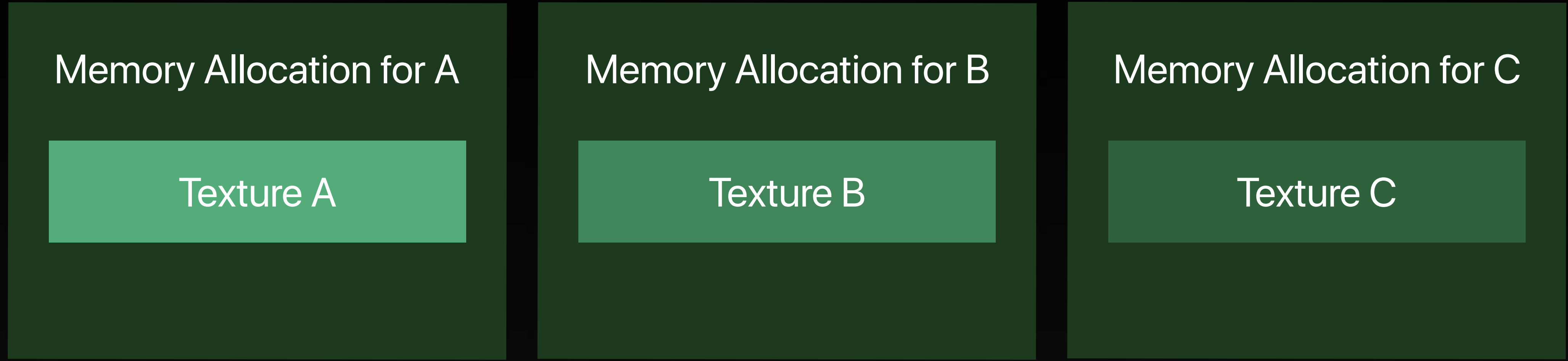
Separate Allocations



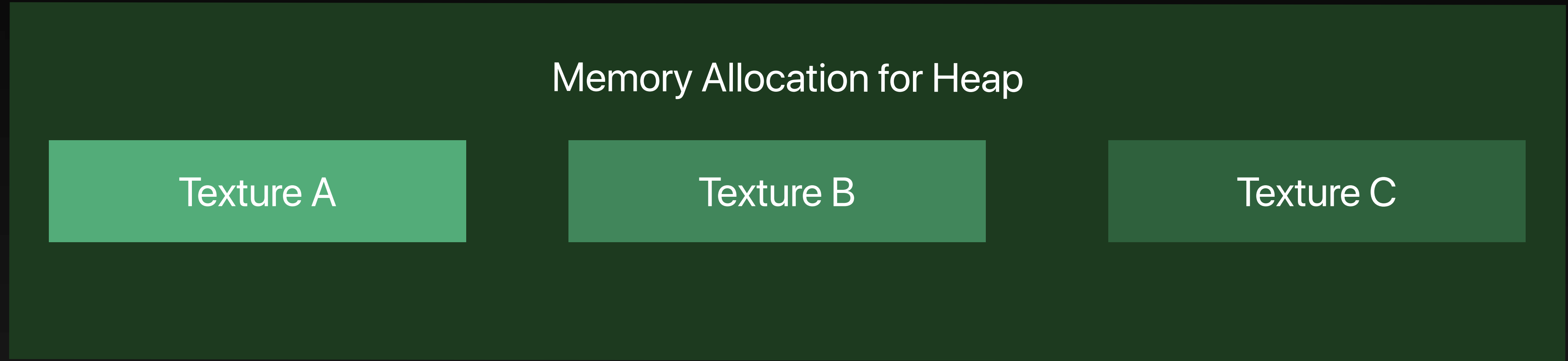
Using Heaps



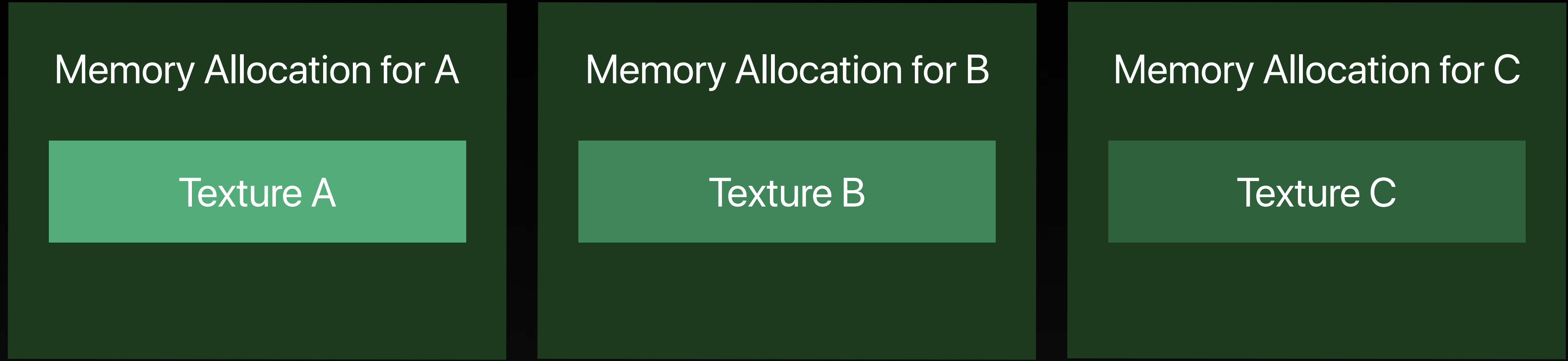
Separate Allocations



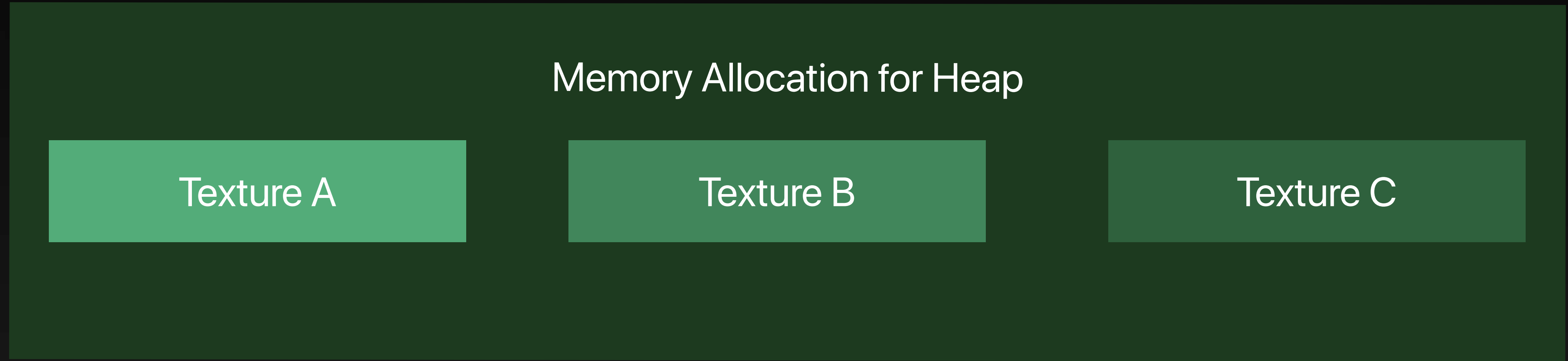
Using Heaps



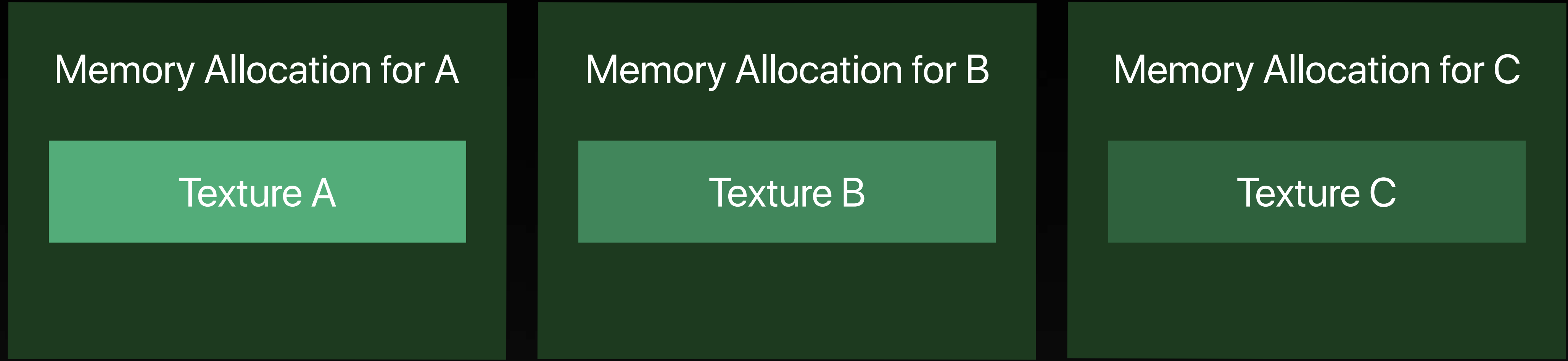
Separate Allocations



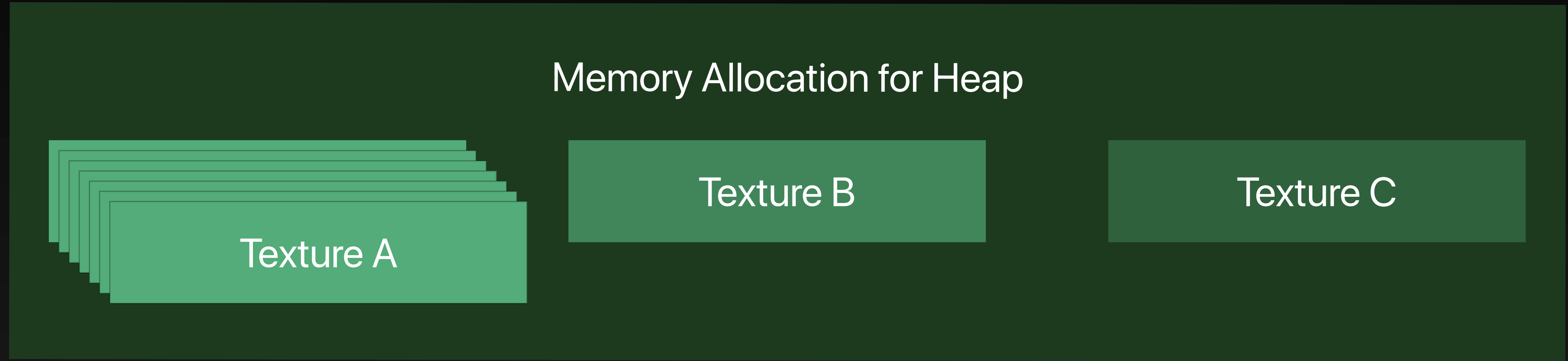
Using Heaps



Separate Allocations

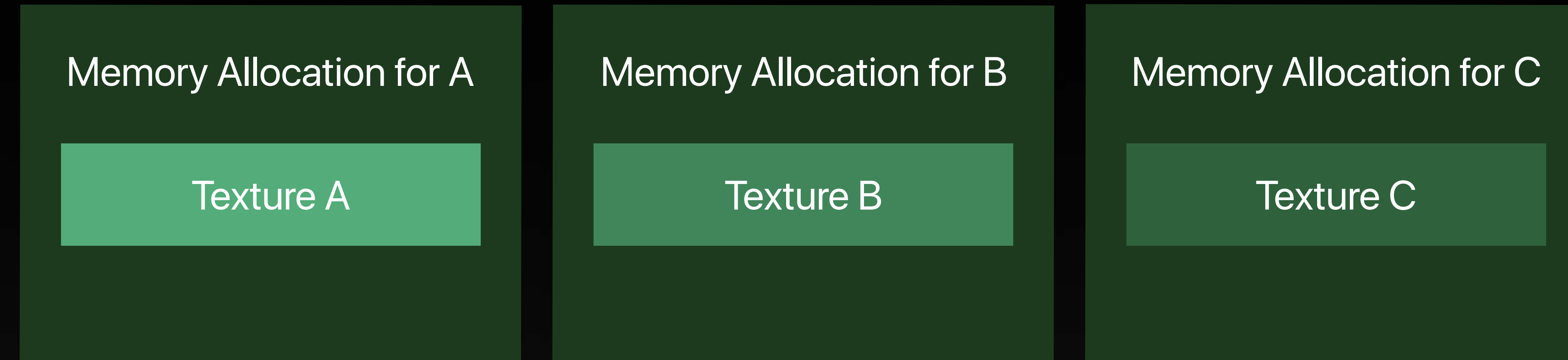


Using Heaps

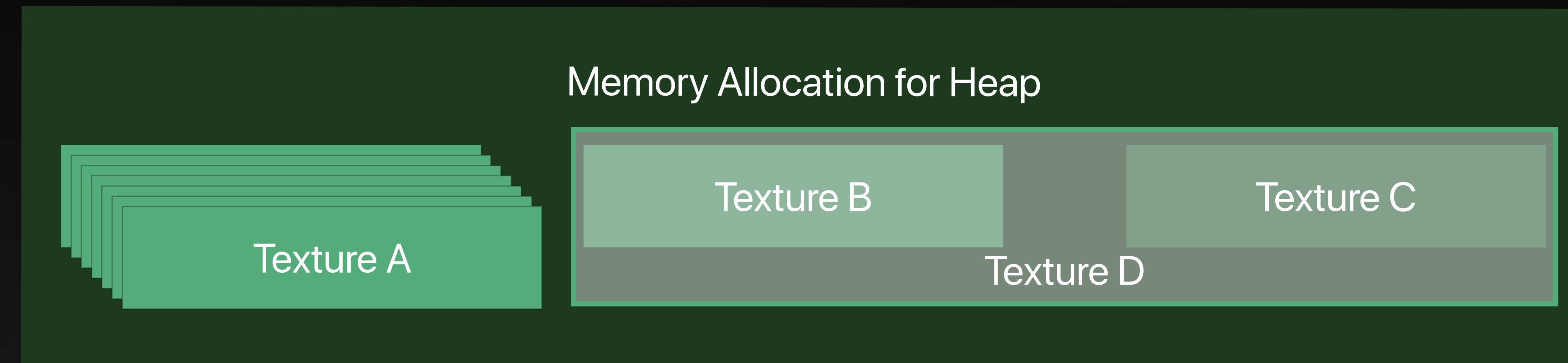


Super fast re-allocation within heap

Separate Allocations



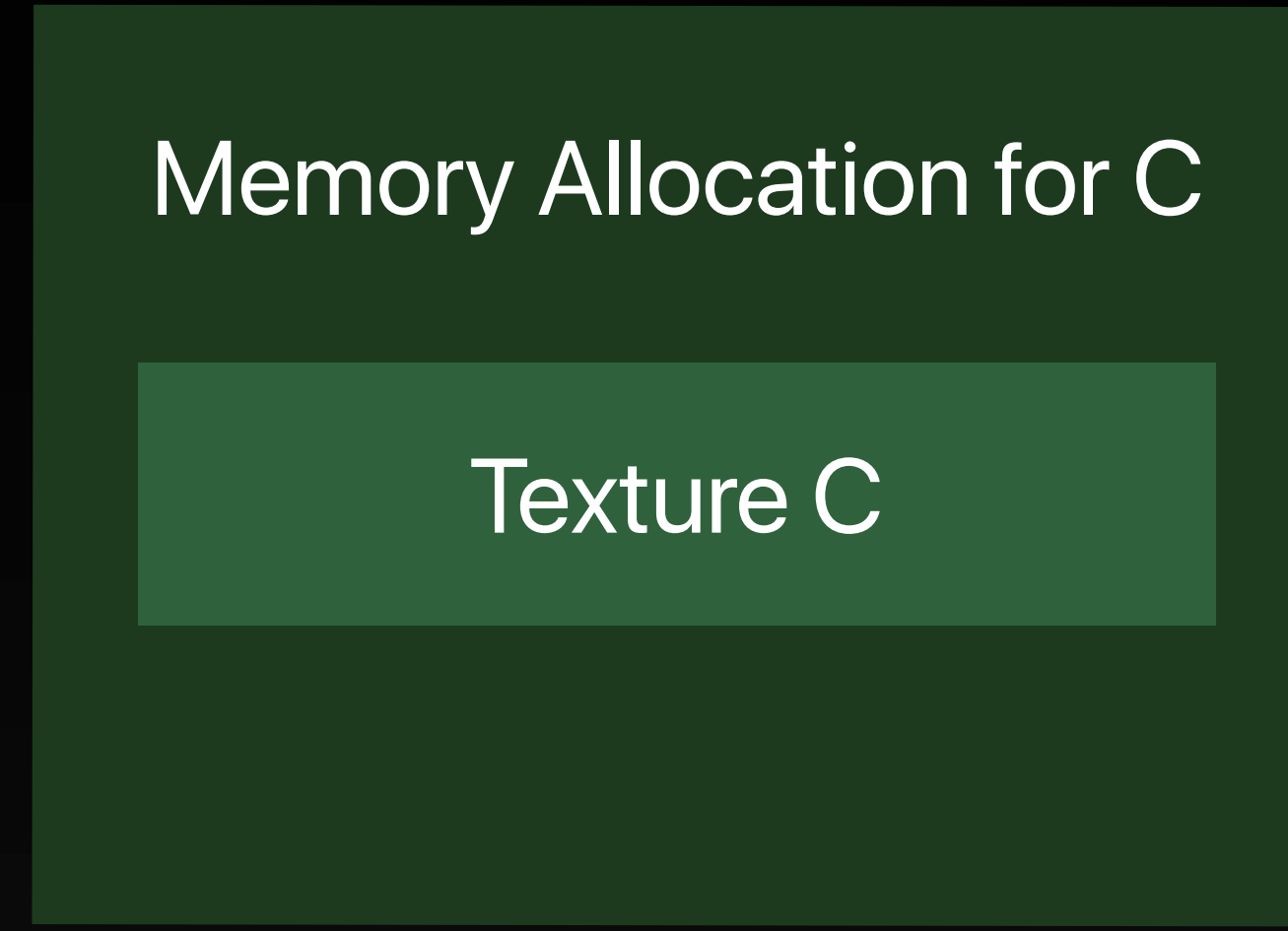
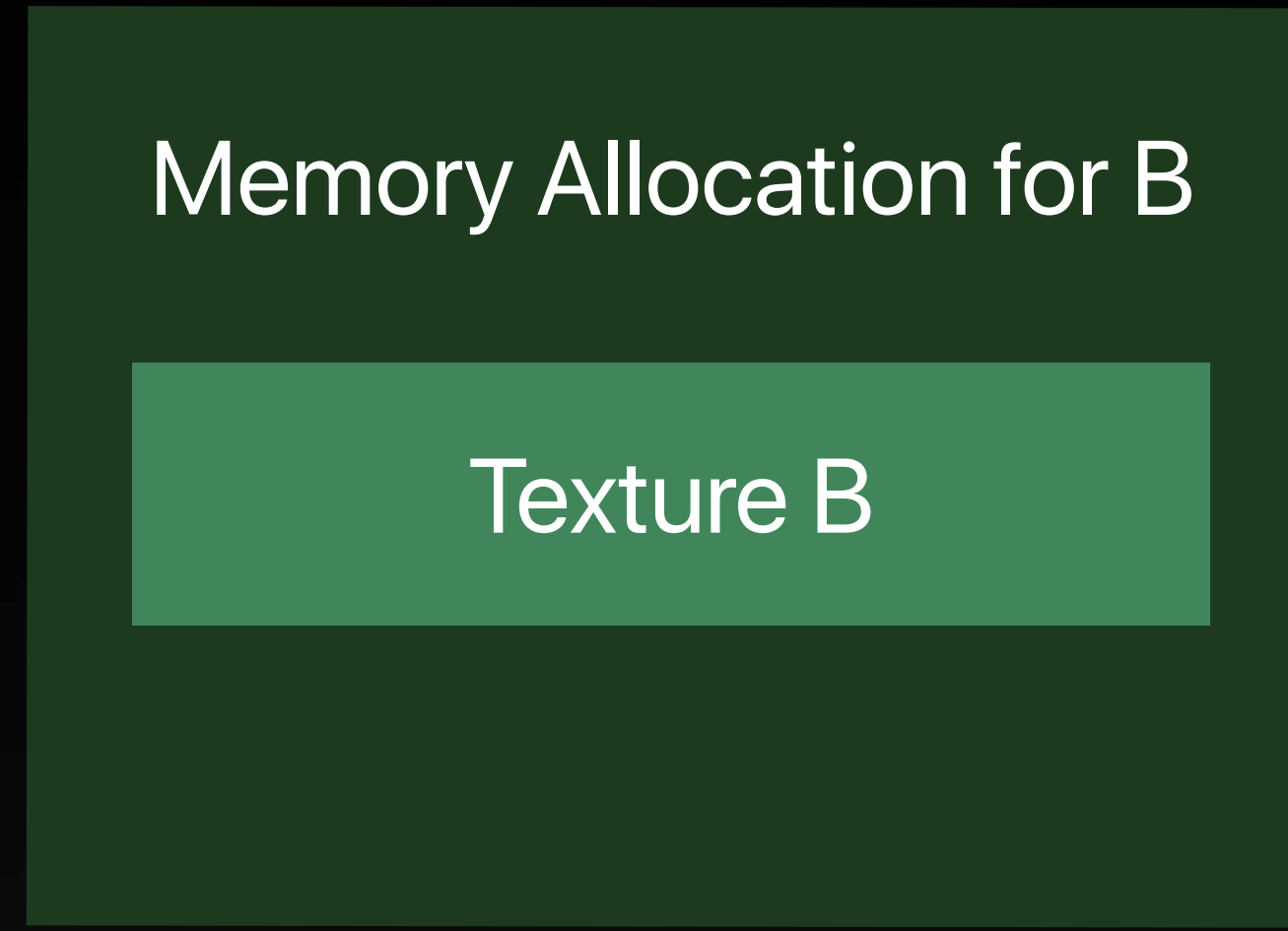
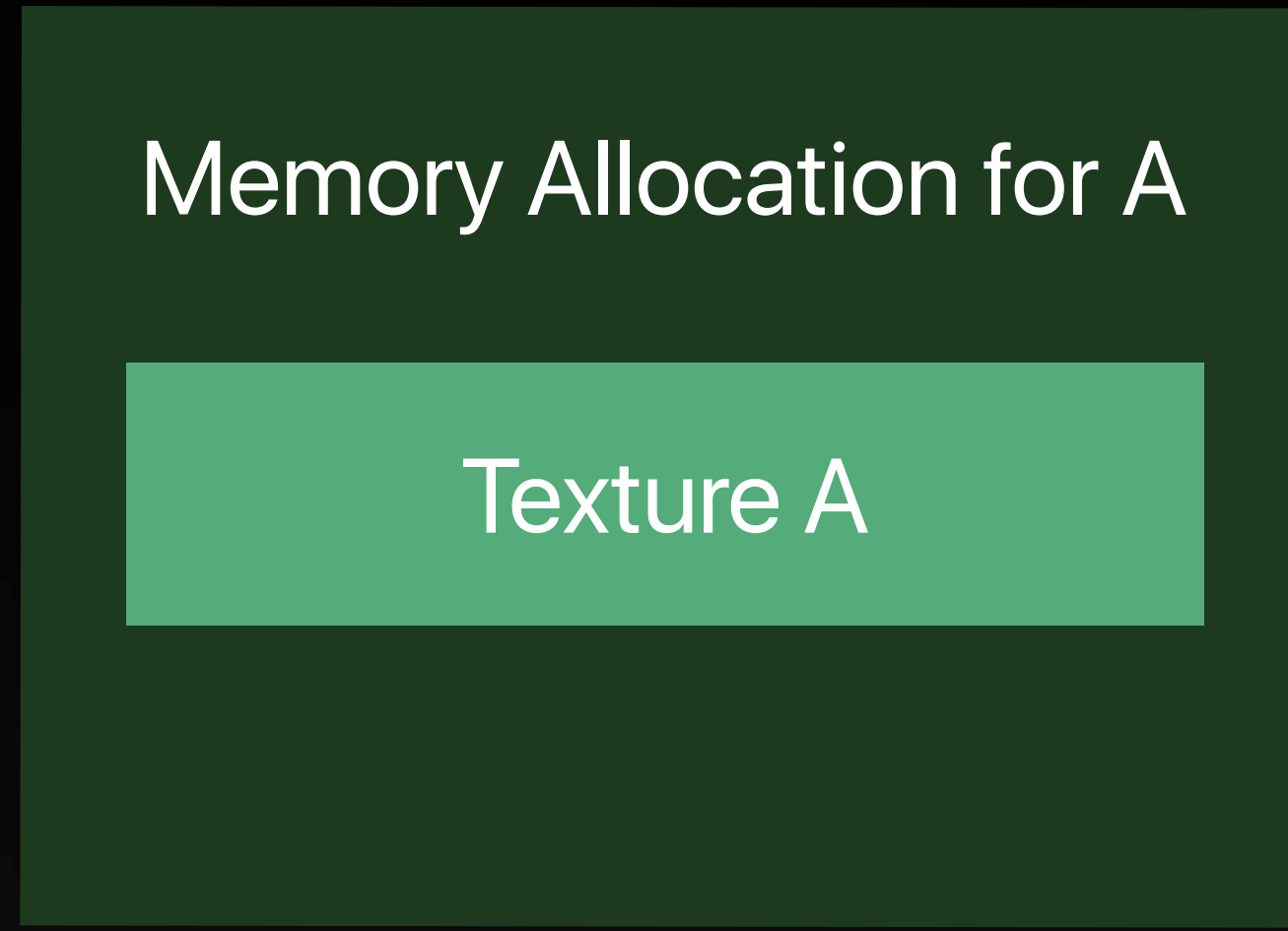
Using Heaps



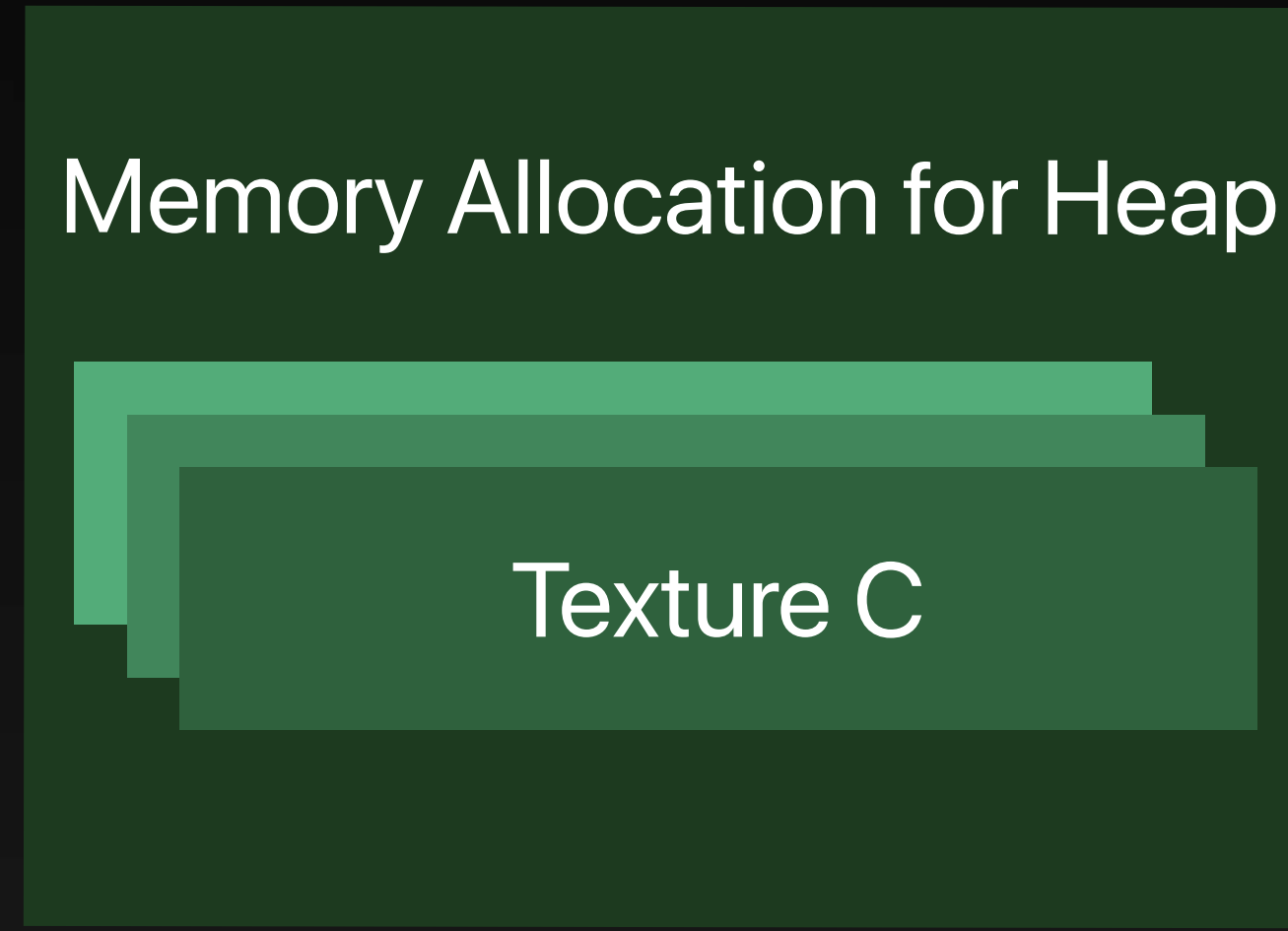
Super fast re-allocation within heap

Re-interpret existing memory

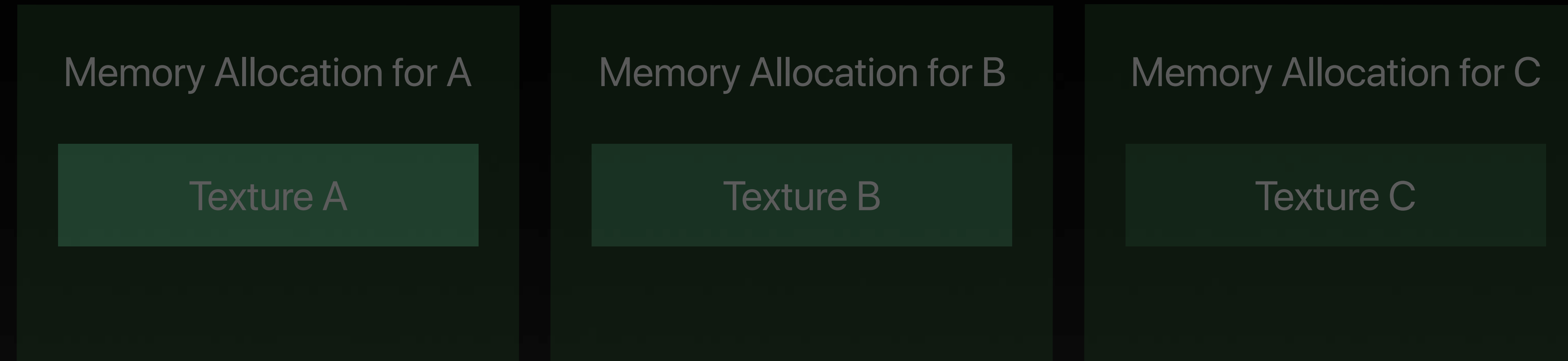
Separate Allocations



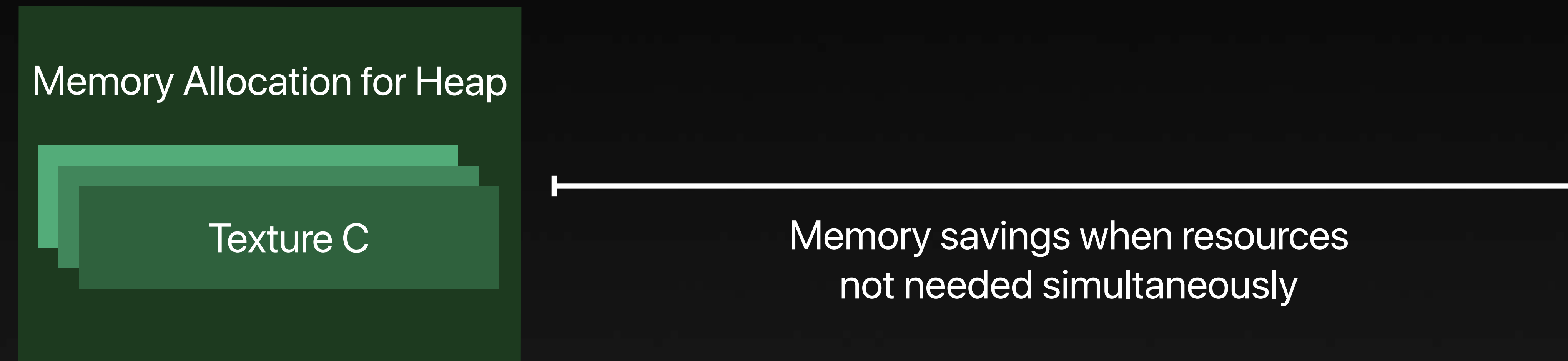
Using Heaps



Separate Allocations



Using Heaps



=	>	÷
x	+	√
%	<	-

Unified graphics and compute
MPS-optimized kernels and API

- Image processing
- Linear algebra
- Math operations



Machine Learning Acceleration



GPU-accelerated primitives

New MPS graph API for CNN

Efficient integration for existing Metal code

Convolution Transpose

Bilinear Rescale

Find Keypoints

Dilated and Sub-pixel Convolution

LSTM

Min-Max, Mean-Variance

Matrix Solvers

Batch Normalization

Binary and XNOR Convolution

Upsampling

Neuron Layers

Matrix - Matrix Multiply

Single Gate

GRU

Dilated and L2Norm Pooling

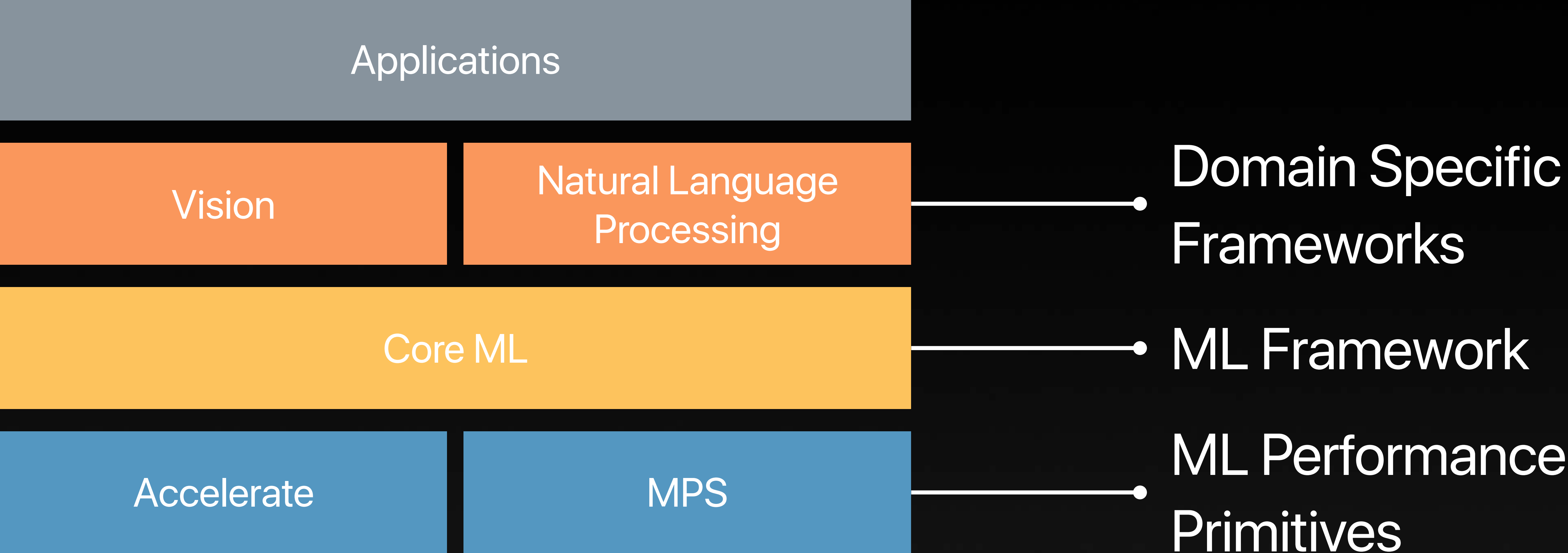
Log Softmax

Matrix - Vector Multiply

Arithmetic Ops

Convolution - 8-bit Weights





Applications

Vision

Natural Language
Processing

Domain Specific
Frameworks

Core ML

ML Framework

Accelerate

MPS

ML Performance
Primitives

Metal



VR



Enabling VR development

360° video editing

3D content creation



Optimized low-latency support for HMD

Metal viewport arrays

System trace VR timelines

Frame debugger per-eye visualization



STEAM[®]VR

Available Today

Beta for macOS



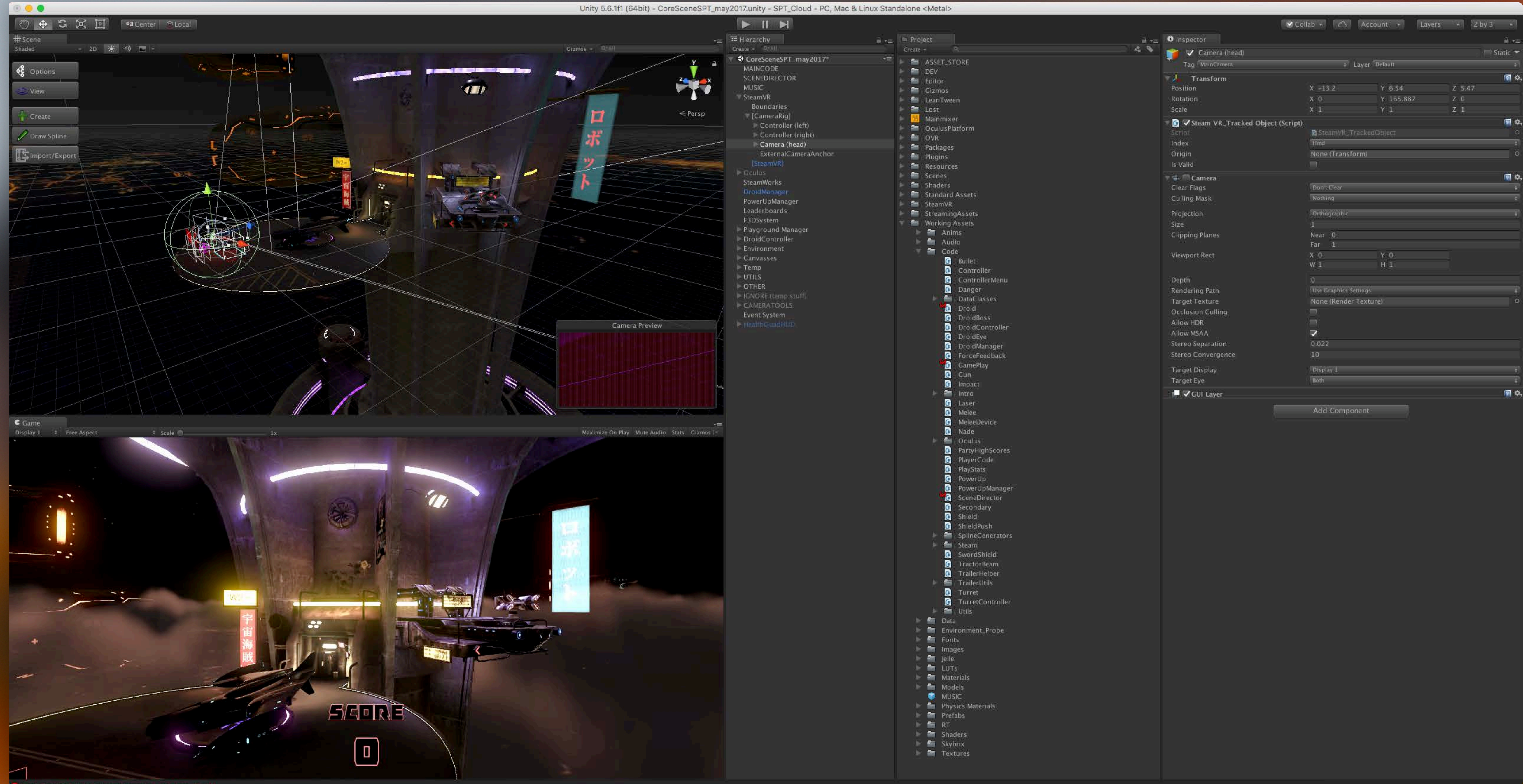
HTC Vive headset and controllers



UNREAL
ENGINE

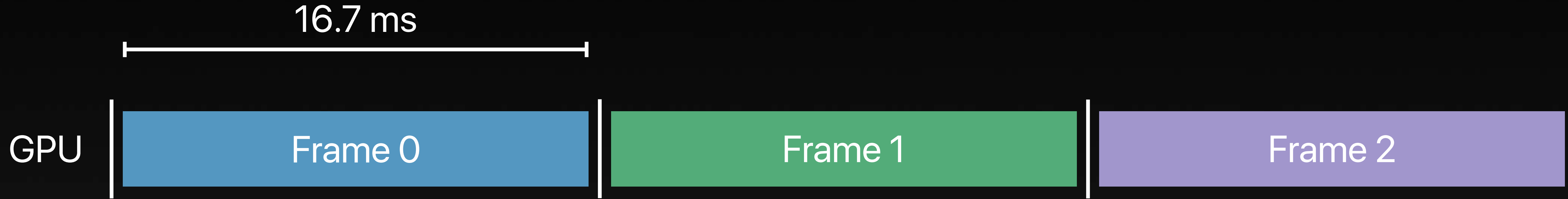






Traditional Rendering

60 frames / second

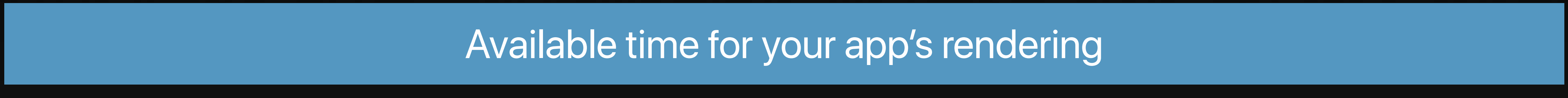


Traditional Rendering

60 frames / second

16.7 ms

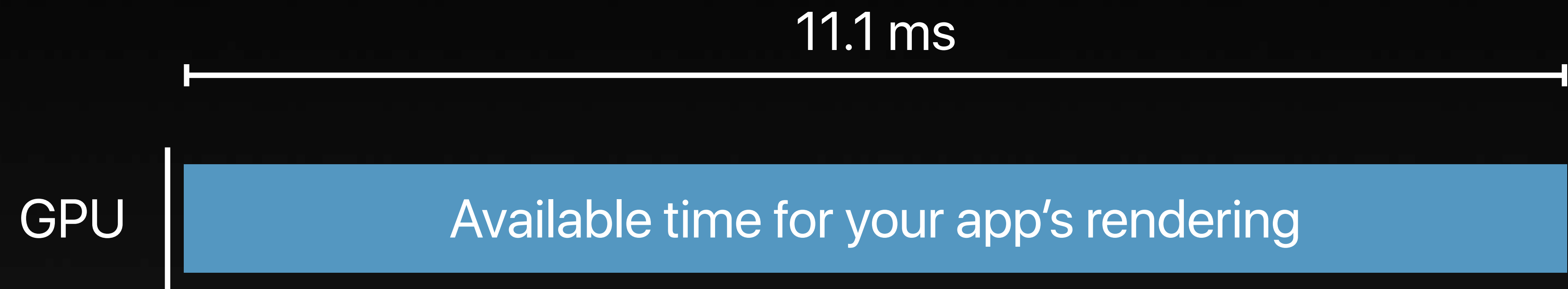
GPU



Available time for your app's rendering

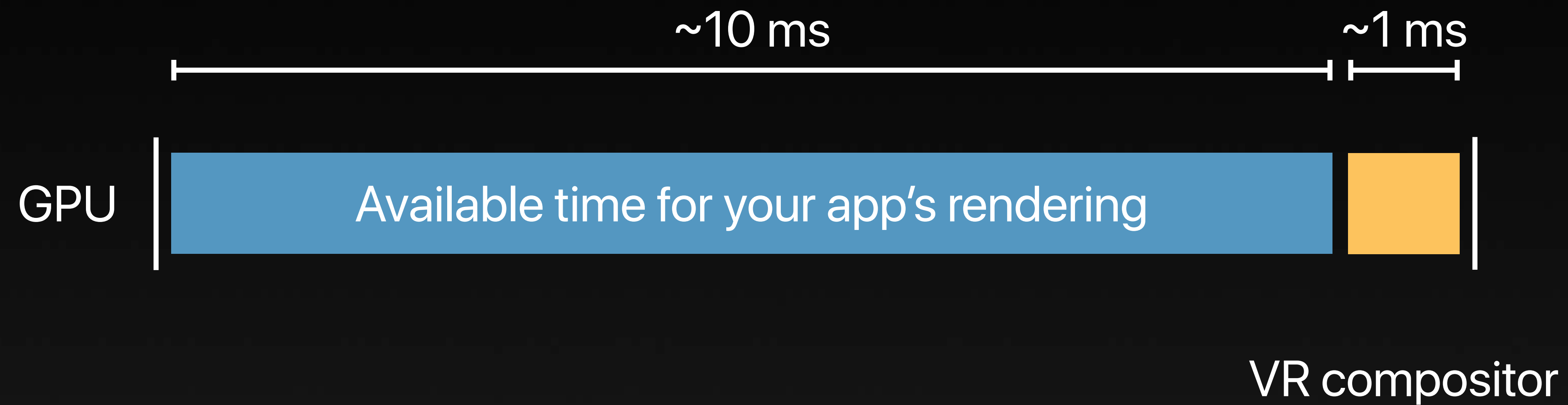
VR Rendering

90 frames / second



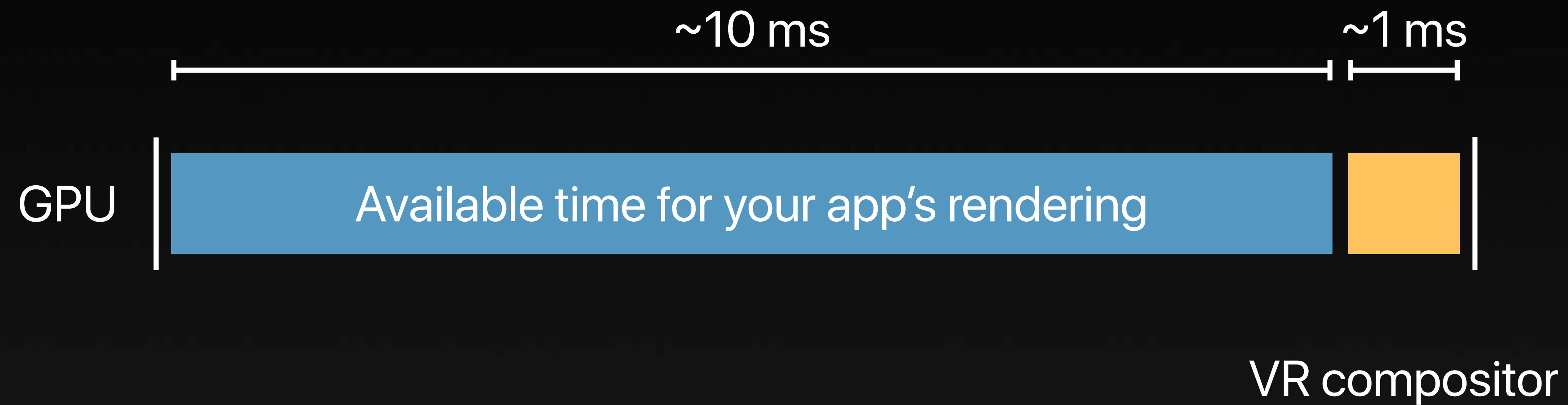
VR Rendering

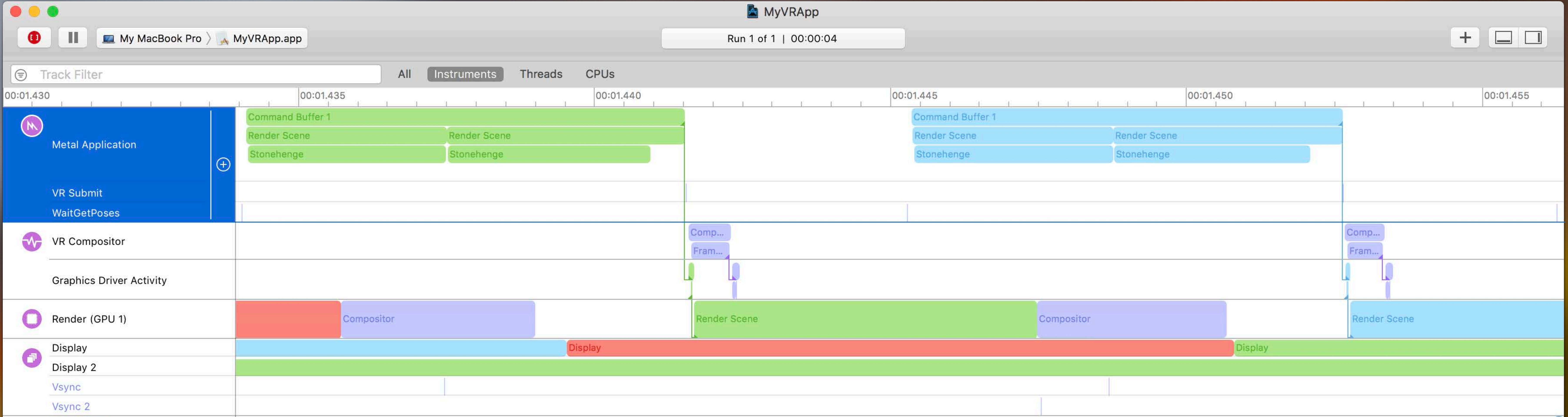
90 frames / second



VR Rendering

~100 frames / second











External GPU Support



Metal device selection API

GPU connection notification API

Improved Metal driver robustness









macOS Developer Beta

Developer Kit



Available Today

Thunderbolt 3 enclosure

AMD Radeon RX 580

USB-C hub

HTC Vive discount

Developer Program members

\$599



Advanced Optimization Tools



GPU counter profiling

Pipeline metrics and remarks

Metal resource Quick Look

Pixel and vertex data inspection

Advanced filtering

New Metal capture APIs

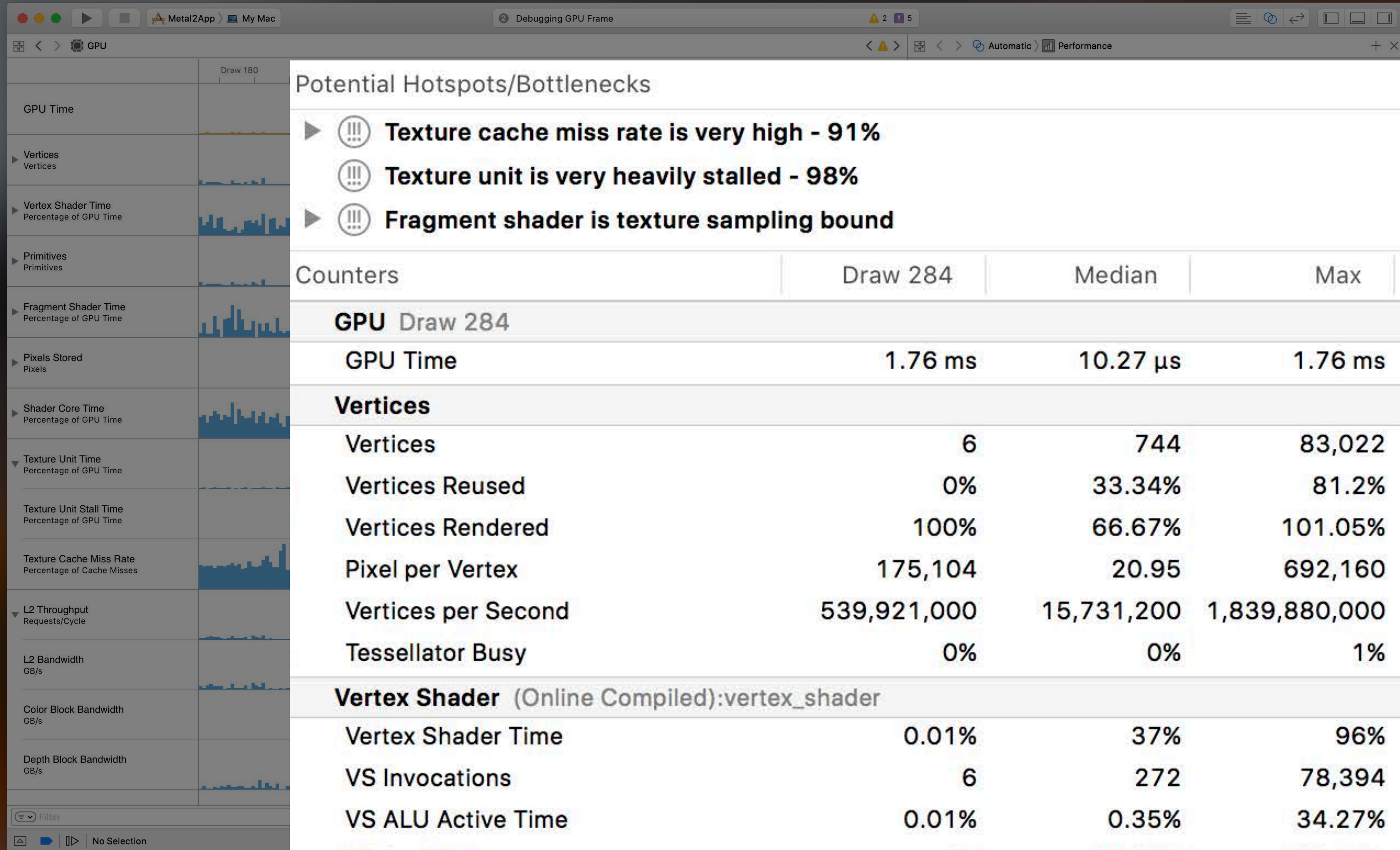
Debugging GPU Frame

GPU Performance

Potential Hotspots/Bottlenecks

- Texture cache miss rate is very high - 91%
- Texture unit is very heavily stalled - 98%
- Fragment shader is texture sampling bound

Counters	Draw 284	Median	Max	Total
GPU Draw 284				
GPU Time	1.76 ms	10.27 μs	1.76 ms	52.53 ms
Vertices				
Vertices	6	744	83,022	4,822,386
Vertices Reused	0%	33.34%	81.2%	-
Vertices Rendered	100%	66.67%	101.05%	-
Pixel per Vertex	175,104	20.95	692,160	-
Vertices per Second	539,921,000	15,731,200	1,839,880,000	-
Tessellator Busy	0%	0%	1%	-
Vertex Shader (Online Compiled):vertex_shader				
Vertex Shader Time	0.01%	37%	96%	-
VS Invocations	6	272	78,394	2,475,048
VS ALU Active Time	0.01%	0.35%	34.27%	-
VS Stall Time	0%	39.36%	90.57%	-
ALU Instructions/VS Invocation	6	150	272	265,857
Memory Instructions/VS Invocation	2	19	41	36,541
Control Instructions/VS Invocation	0	0	2	1,258
VS ALU to Memory Instructions Ratio	3	7.41	12.54	-
Average VS SIMD-Groups	0.29	0.61	13.49	2,640.99
Average VS SIMD-Group Latency	381	873.55	18,928.56	1,666,285.75
VS SIMD-Group Memory Stall Time	42.79%	7.88%	99%	-
VS SIMD-Group Export Stall Time	2.89%	4.83%	97.88%	-
Parameter Cache Stall Time	0%	0%	93%	-
Primitives				
Primitives	2	248	27,674	1,607,462
Primitives Culled	0%	47.5%	150%	-
Primitives Culled (Zero-Area)	0%	0%	100%	-
Primitives Culled (Clipper)	0%	0%	100%	-
Primitives Clipped	0%	0%	100%	-
Primitives Rendered	100%	100%	300%	-
Pixels per Primitive	525,312	62.85	2,076,480	-
Hi-Z Test Fails	0%	92.88%	100%	-
PreZ Test Fails	0%	0%	99.84%	-
ROP Stall Time	0.03%	0%	44.66%	-
Primitives per Second	179,973,600	11,265,590	2,398,959,000	-
Fragment Shader (Online Compiled):pixel_shader				
Fragment Shader Time	98%	0%	99%	-
FS Invocations	1,048,576	0	2,127,253	41,050,720
FS ALU Active Time	0.39%	0%	91.4%	-
FS Stall Time	97.43%	23.52%	97.43%	-
ALU Instructions/FS Invocation	8	0	1,011.24	229,645.49
Memory Instructions/FS Invocation	1	0	92	17,752.65



Demo















Metal 2



ARKit



iPhone 6s and later

iPad Pro and later

Hundreds of Millions

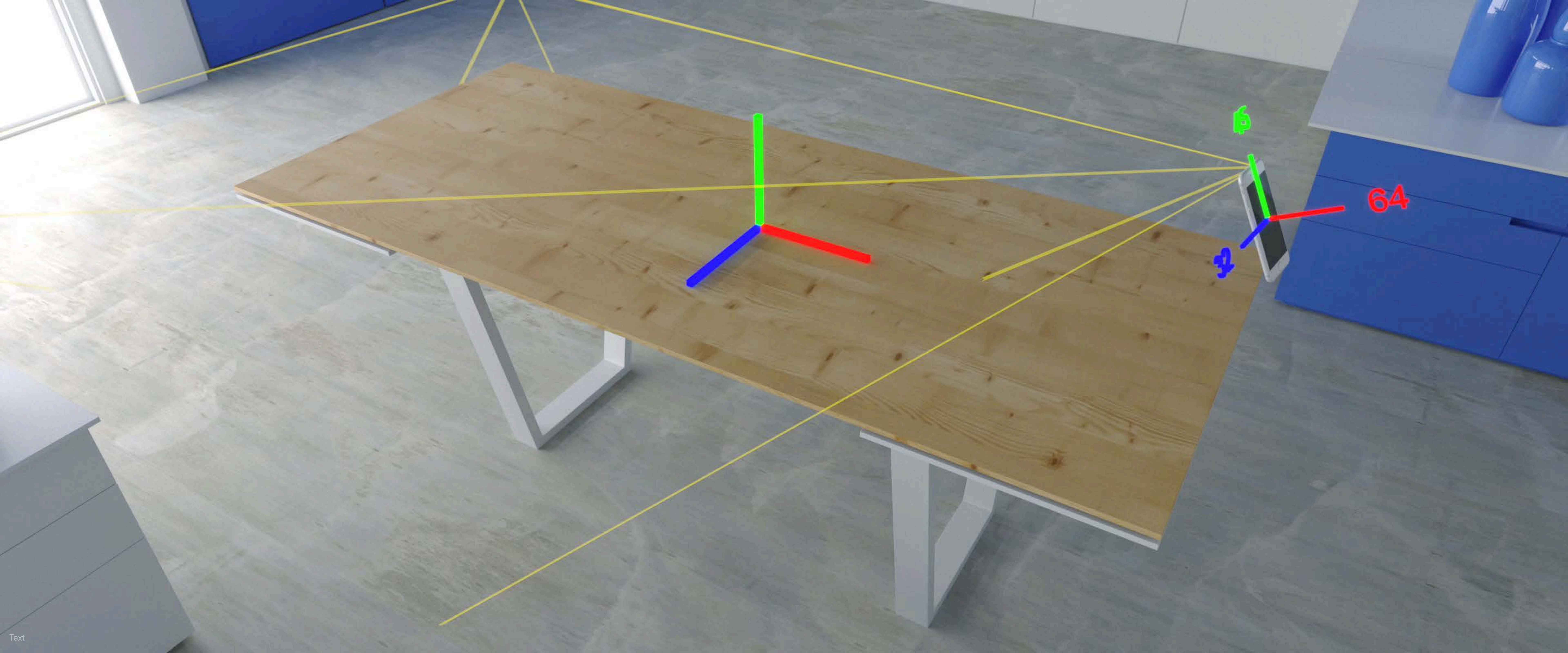
ARKit Capable Devices

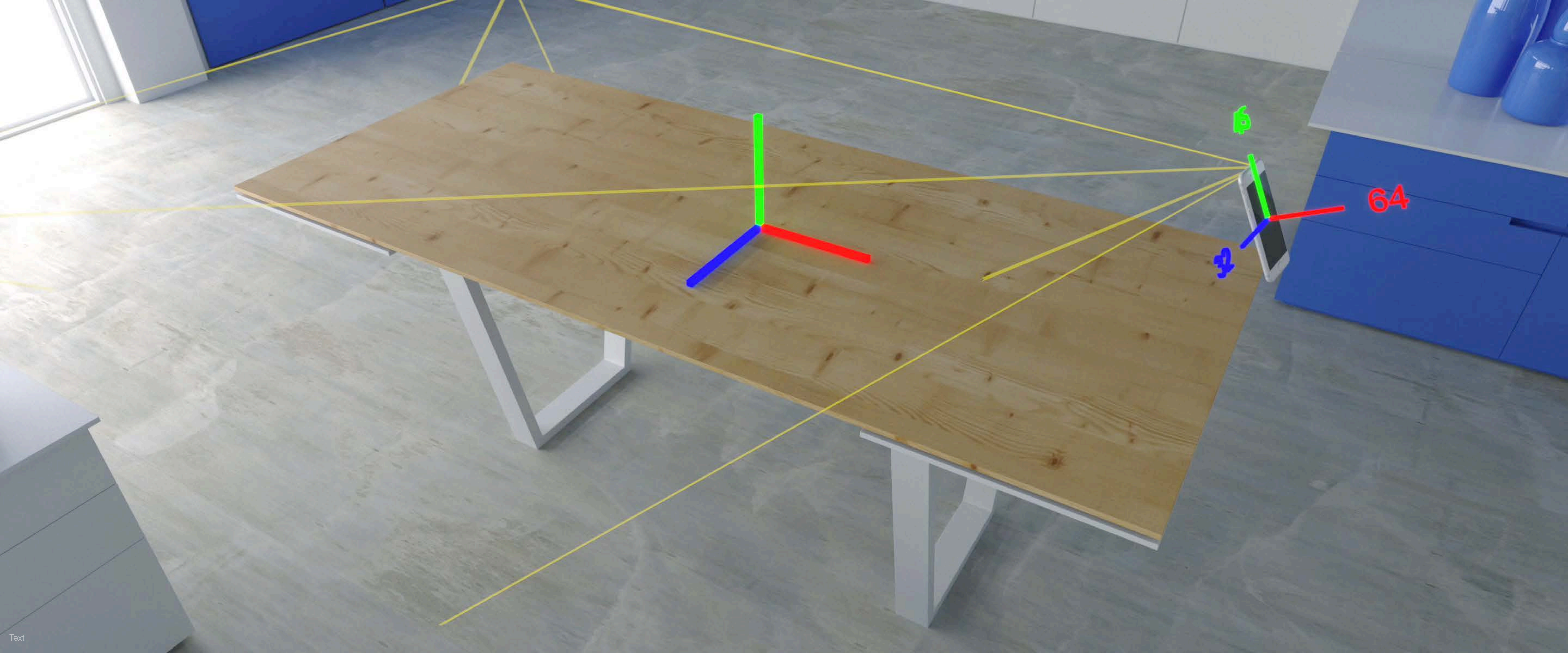


Visual-Inertial Odometry



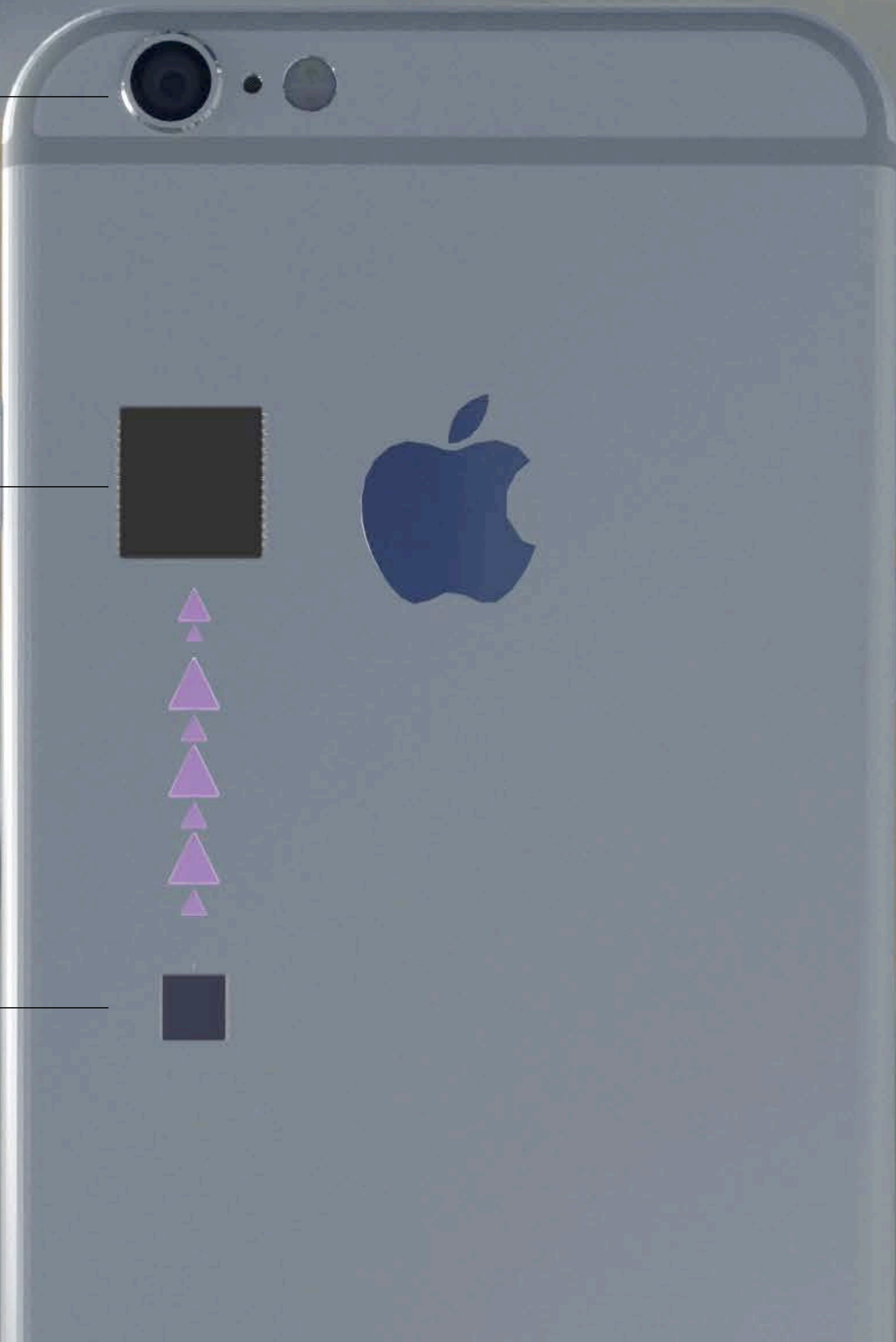
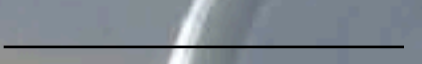








Camera



CPU



Gyro and Accelerometer



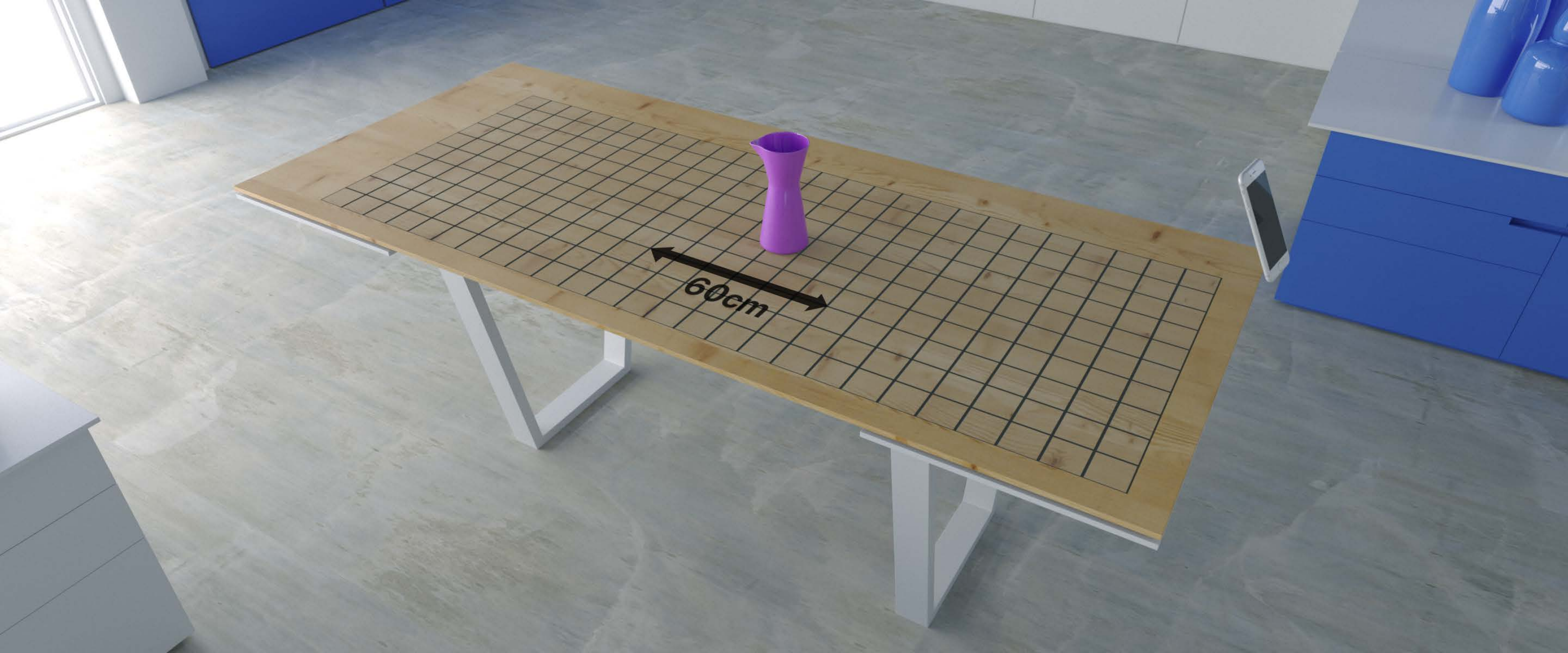


Scene Understanding







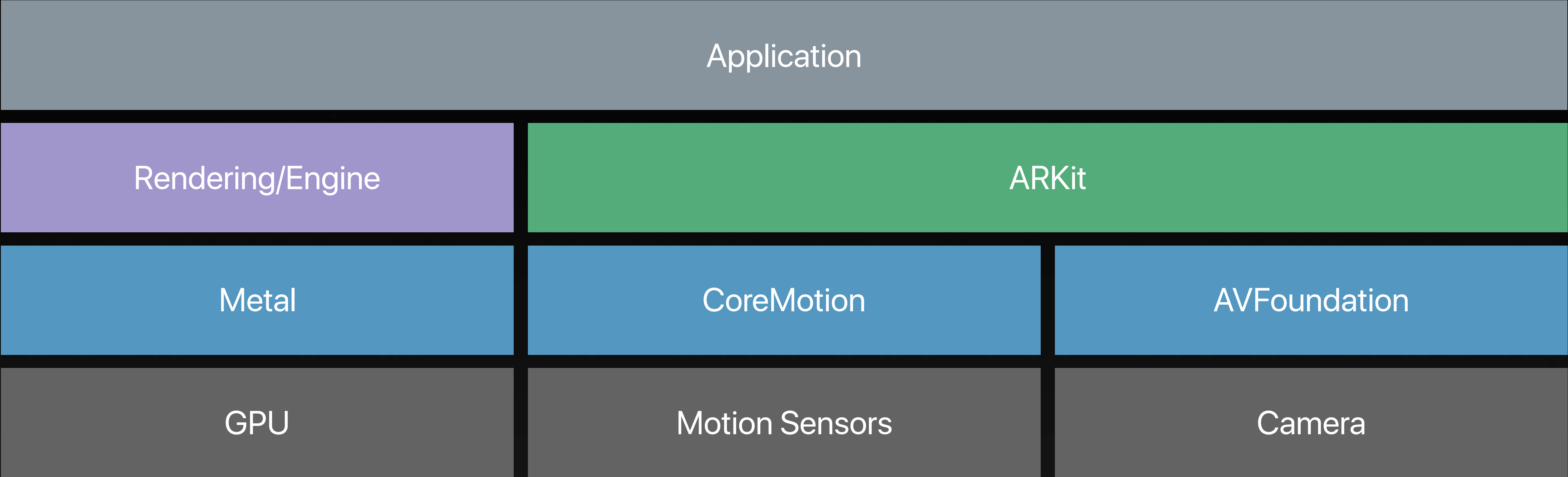


60cm

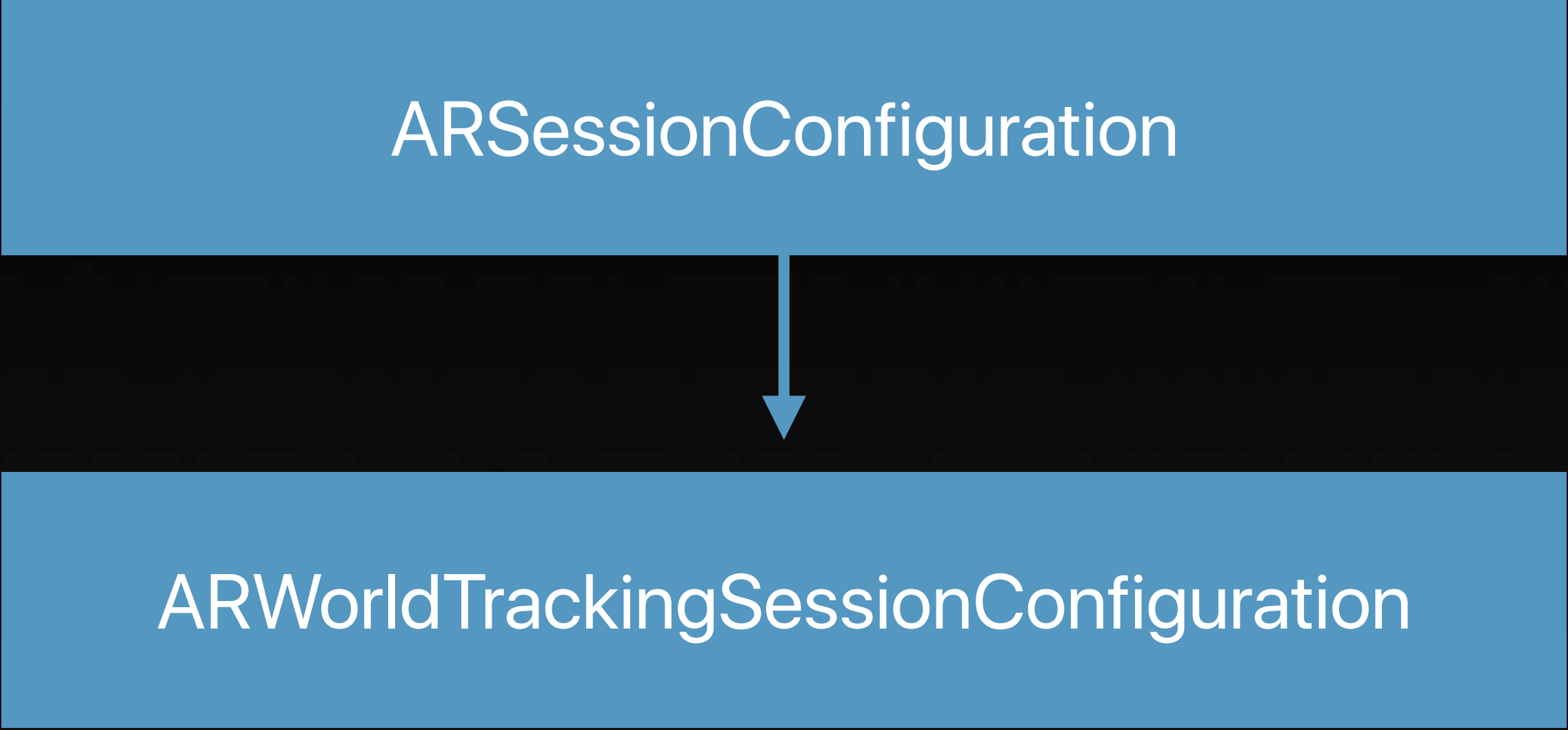






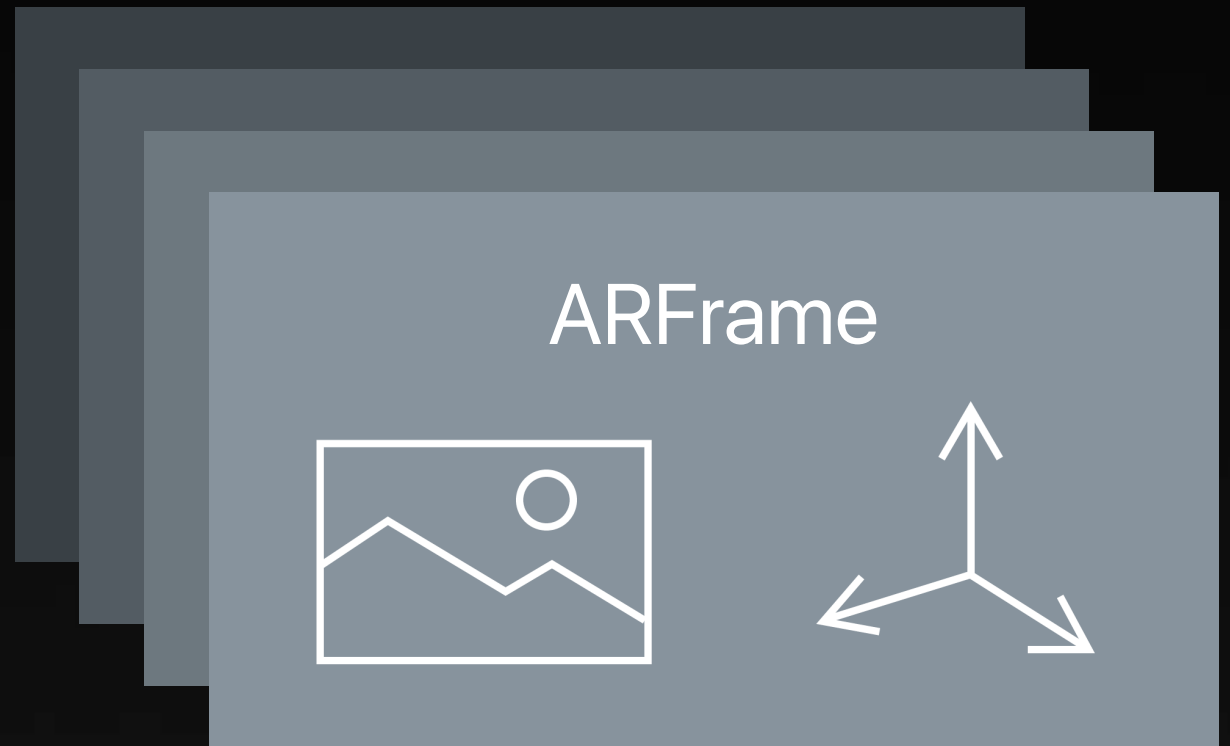


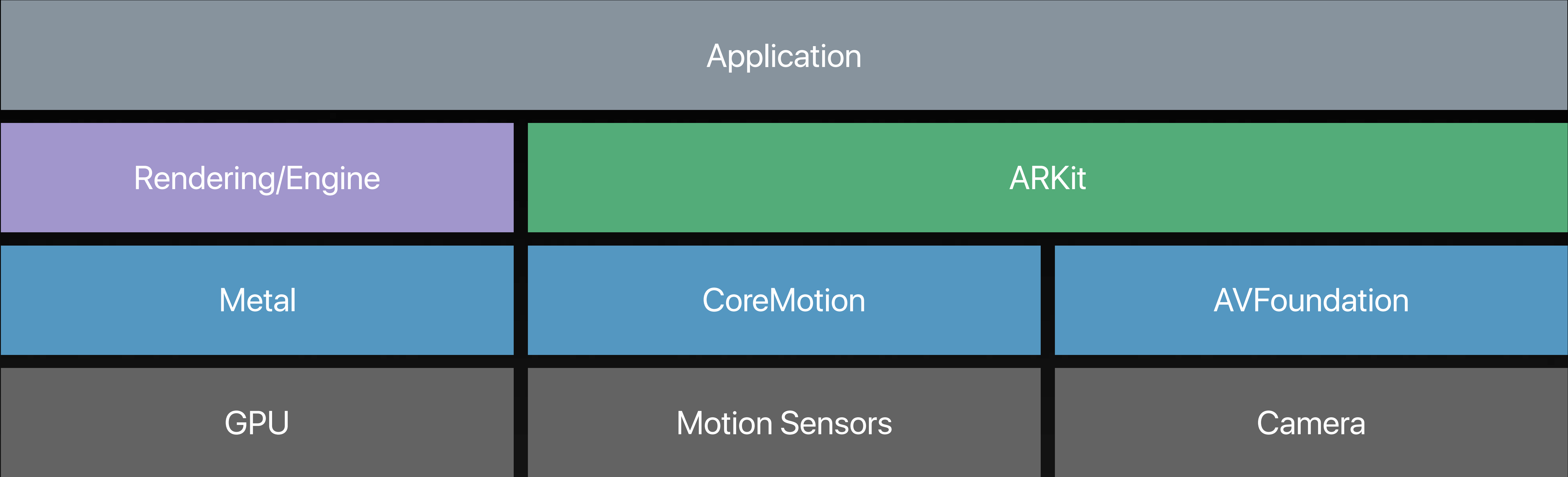
ARKit





ARSessionDelegate
or
currentFrame





Rendering/Engine



Metal 2







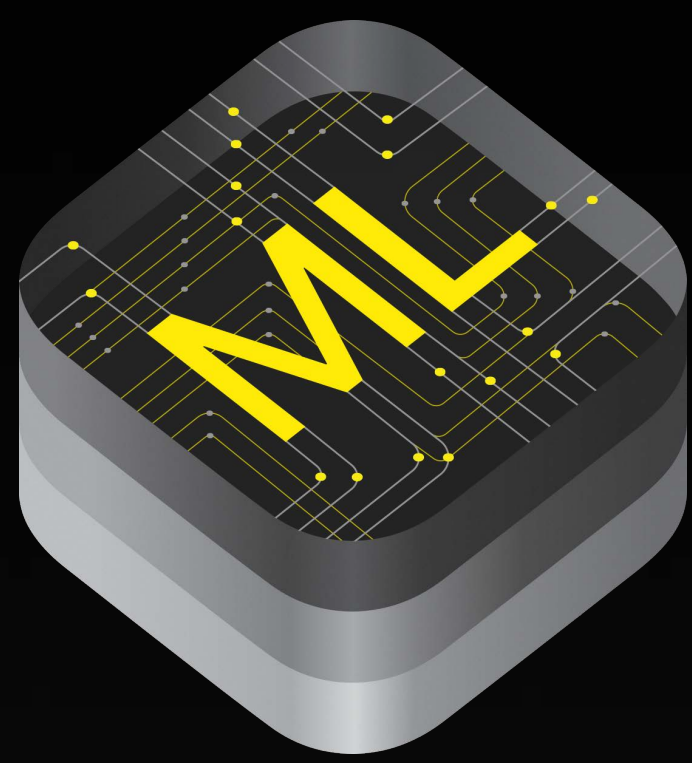
UNREAL
ENGINE



Martin Sanders

Hands On







iOS 11

macOS High Sierra

watchOS 4

tvOS 11

Xcode 9

Swift Playgrounds 2

developer.apple.com/wwdc/resources/



Over 100 sessions

Over 200 labs

Over 1000 Apple engineers

 **WWDC17**