

#WWDC19

Building AR Experiences with Reality Composer

Michelle C., Designer

Pau Sastre Miguel, Software Engineer

Abhi Ashutosh, Software Engineer

Reality Composer

Get started with AR and 3D

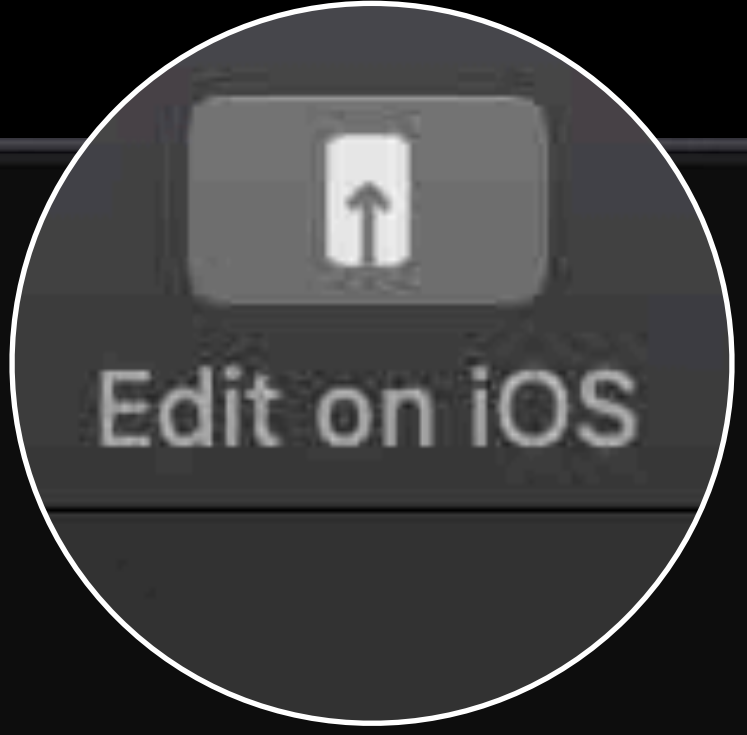
Layout and pre-visualization

Content library

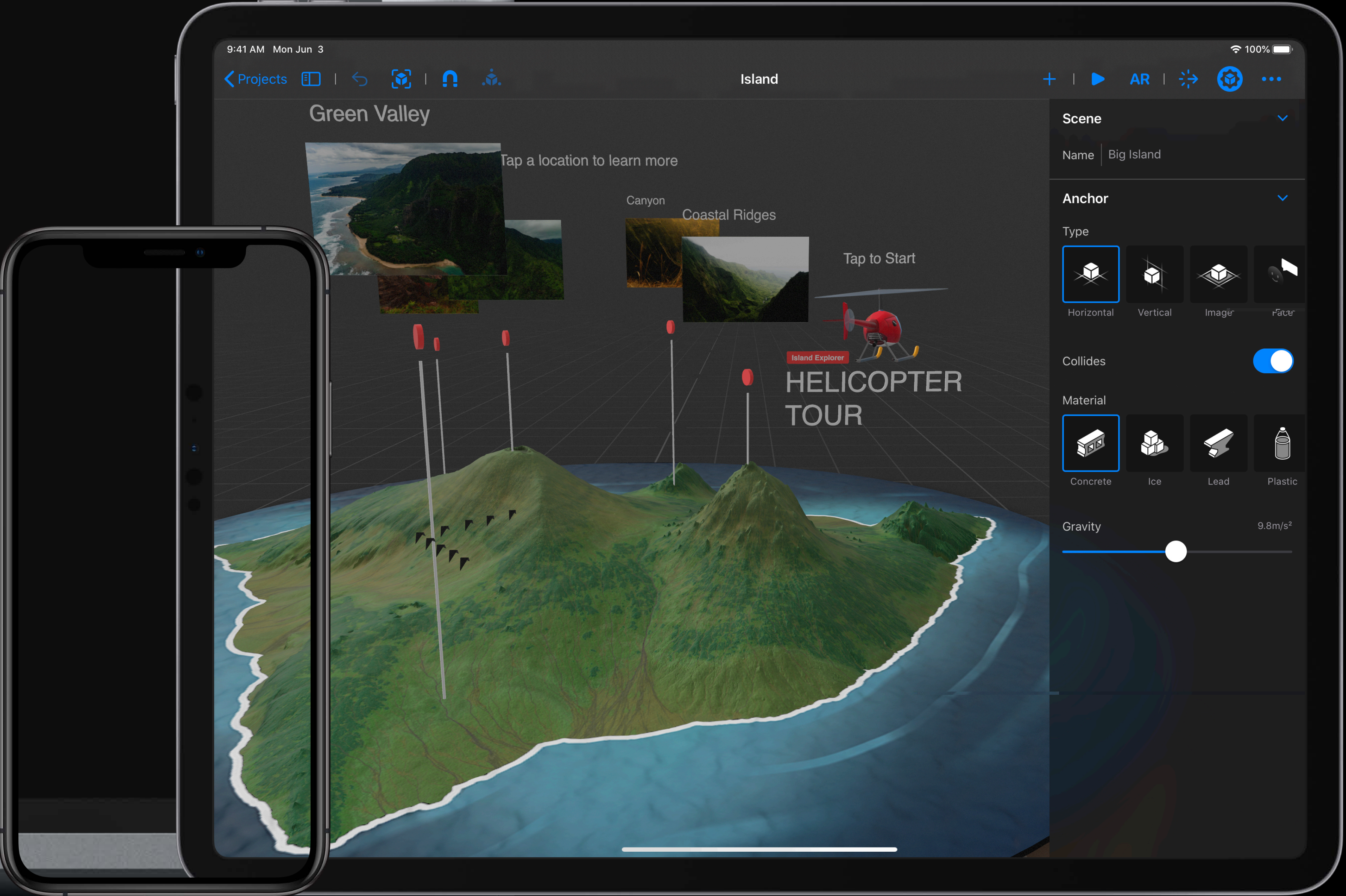
Simple interactions

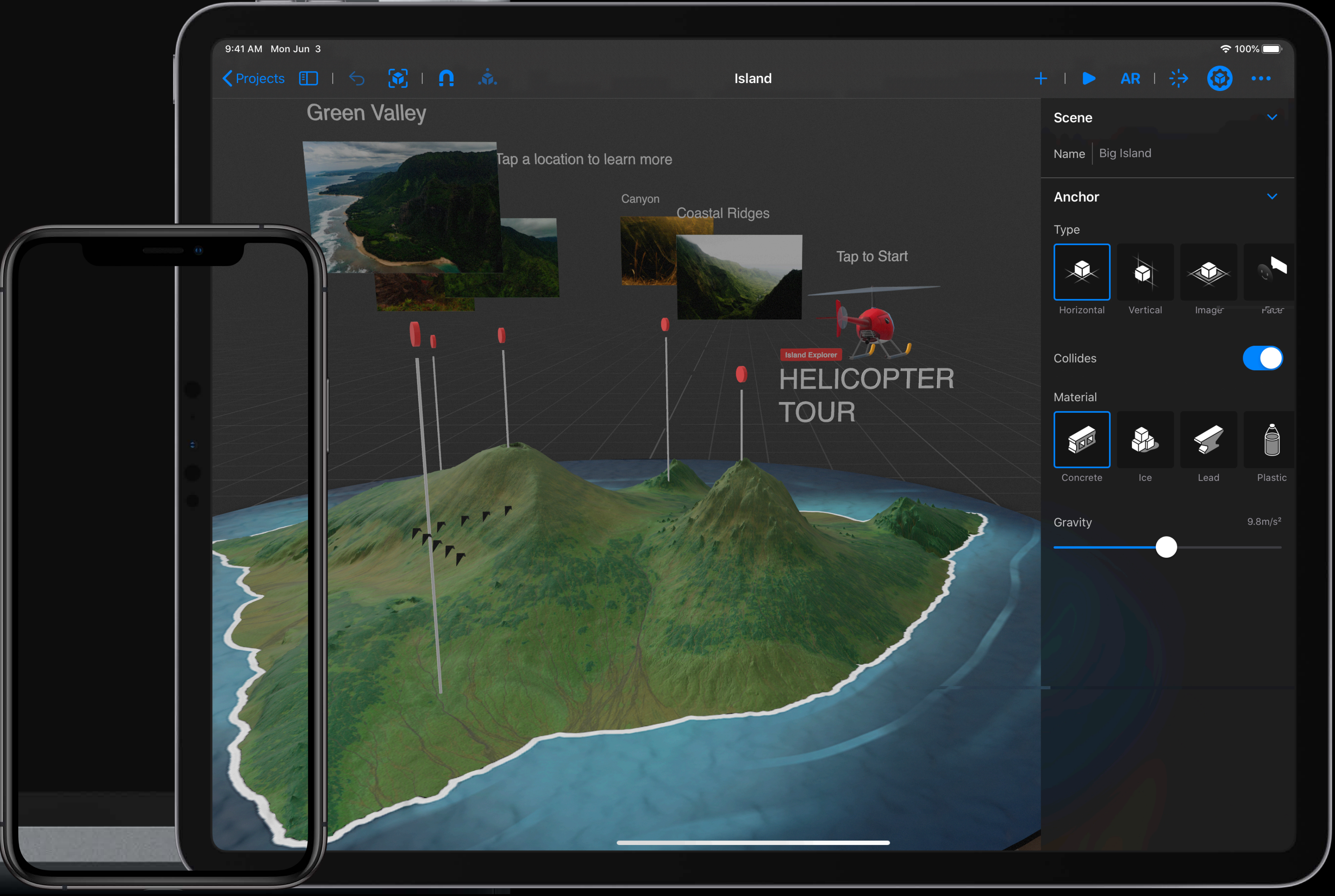
Xcode integration





MacBook Pro





Building Scenes

Adding Behaviors

Using Physics

Building Apps

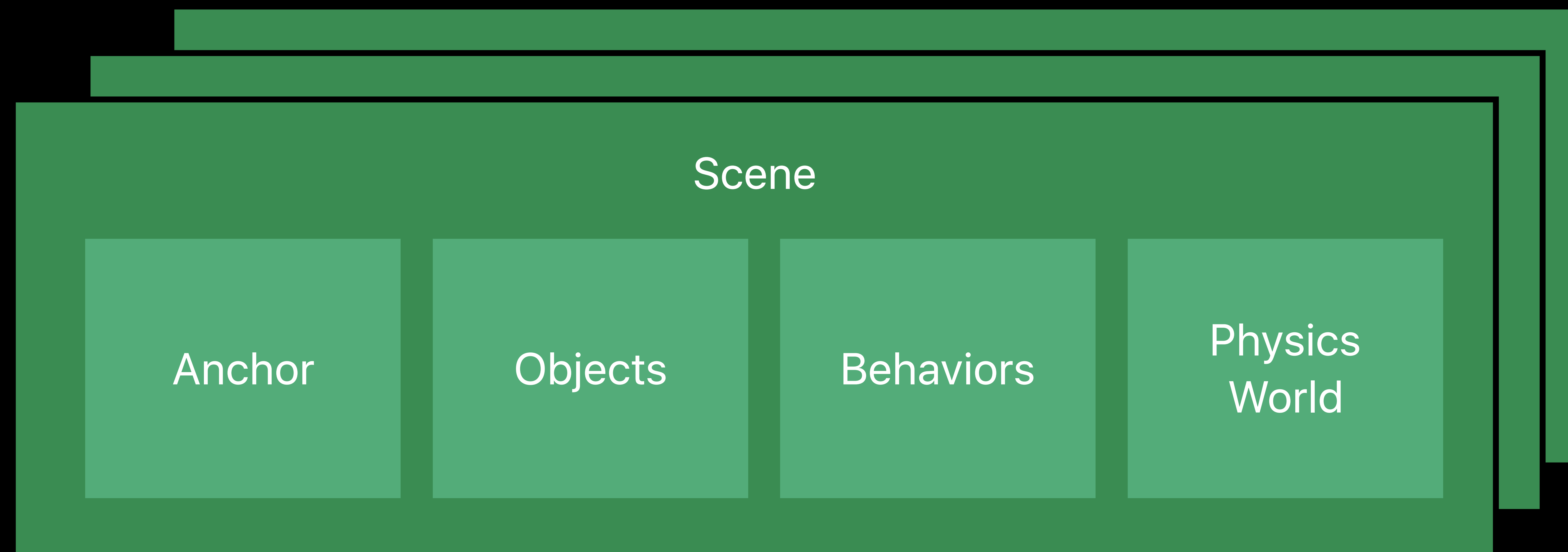
Building Scenes

Adding Behaviors

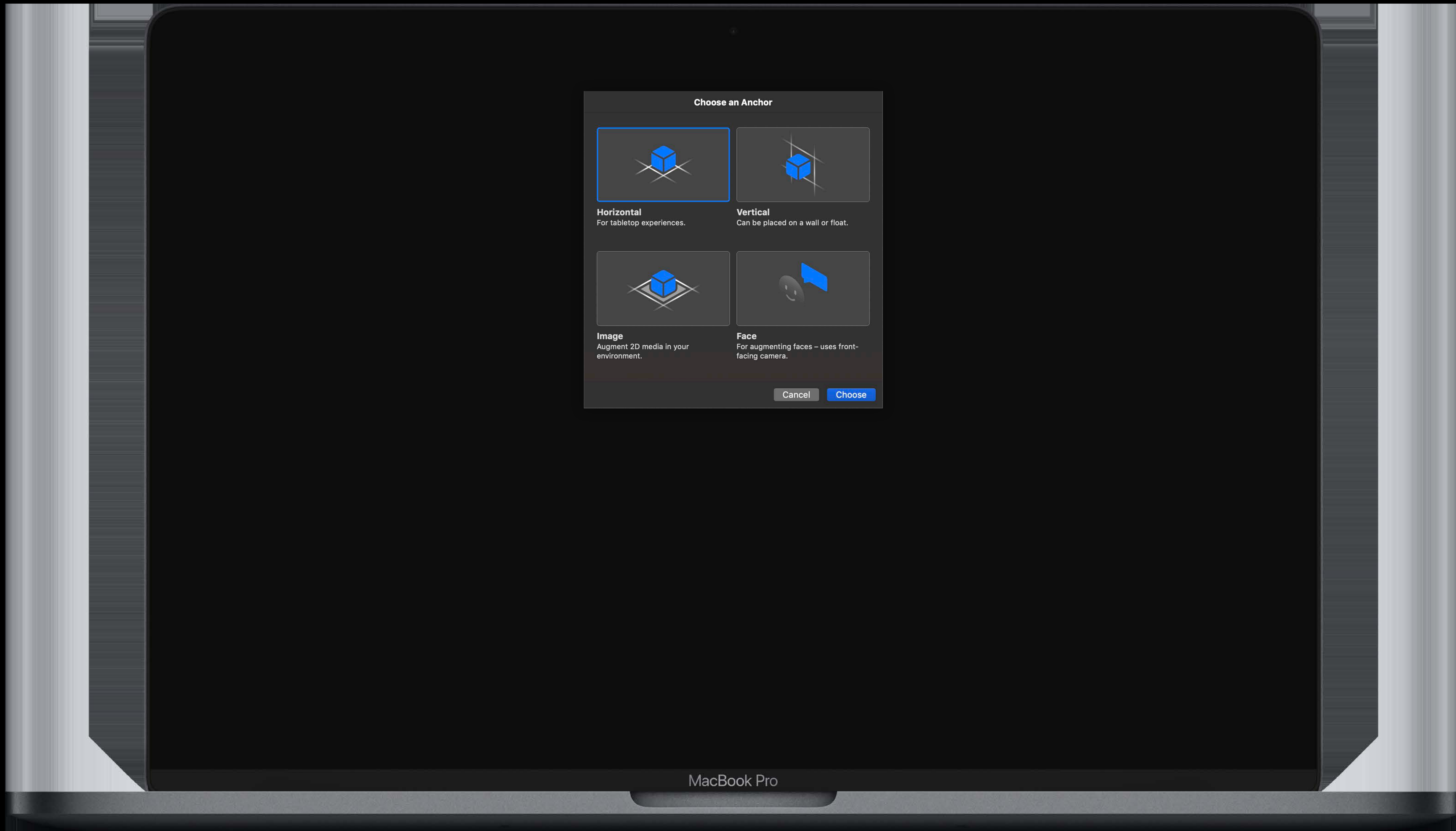
Using Physics

Building Apps

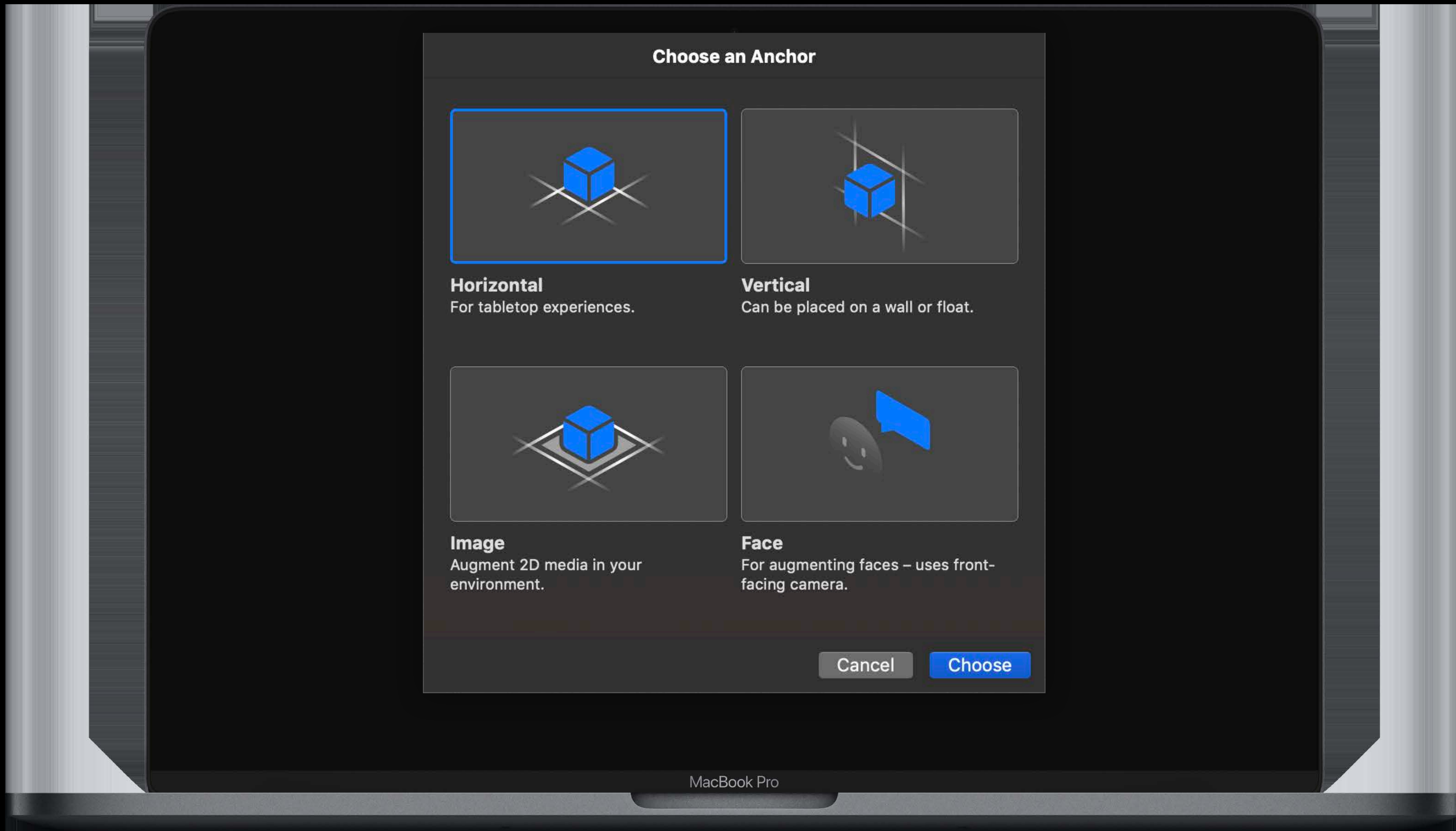
Anatomy of a Scene



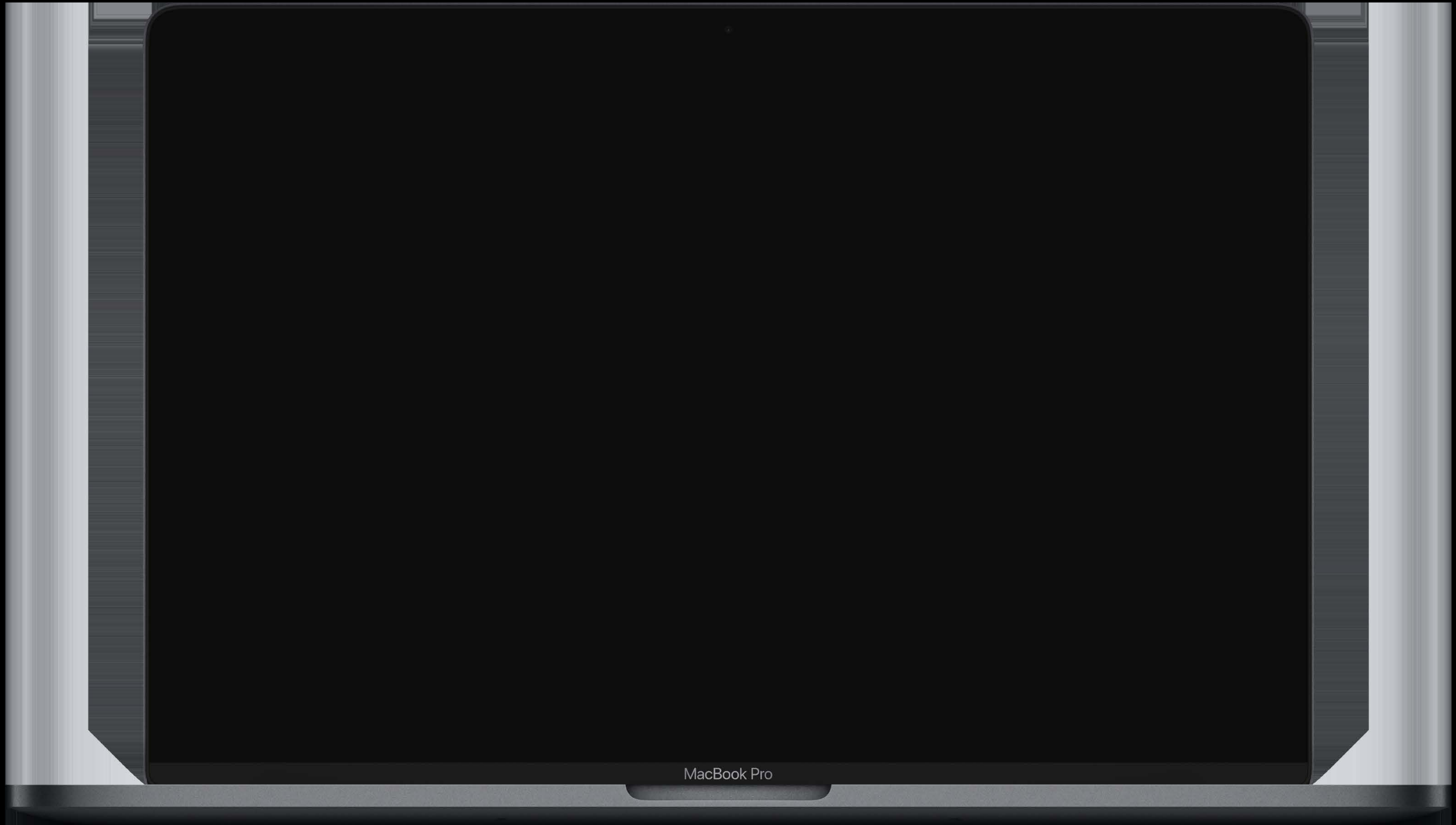
Anchoring the Scene



Anchoring the Scene



Working with Scenes



Demo

Building Scenes

Adding Behaviors

Using Physics

Building Apps

Building Scenes

Adding Behaviors

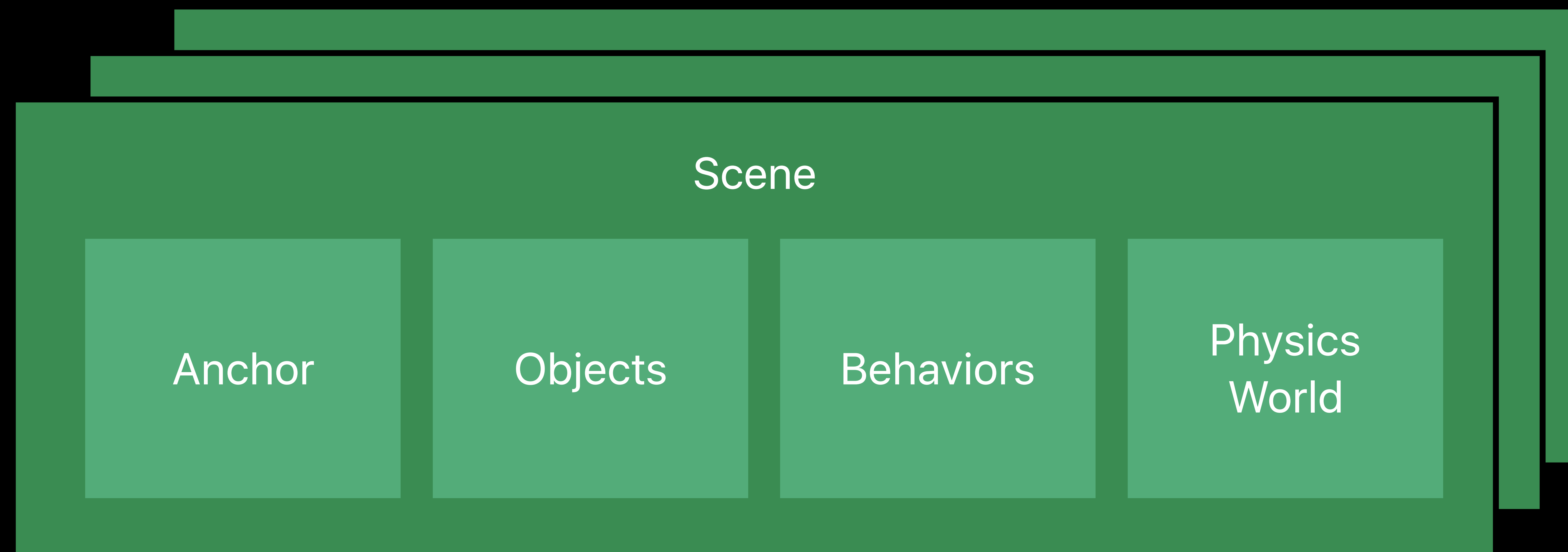
Using Physics

Building Apps

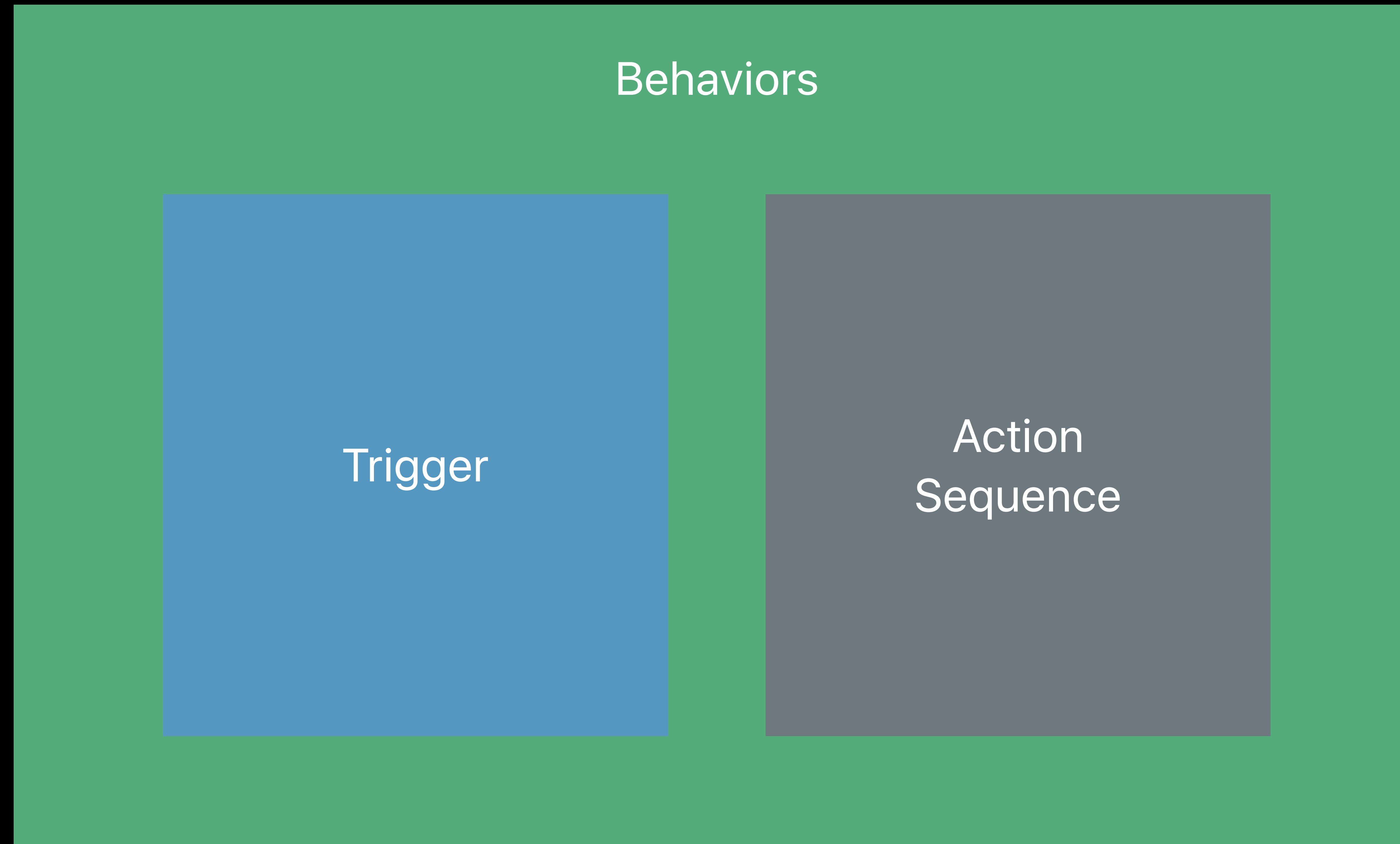
Adding Behaviors

Pau Sastre Miguel, Software Engineer

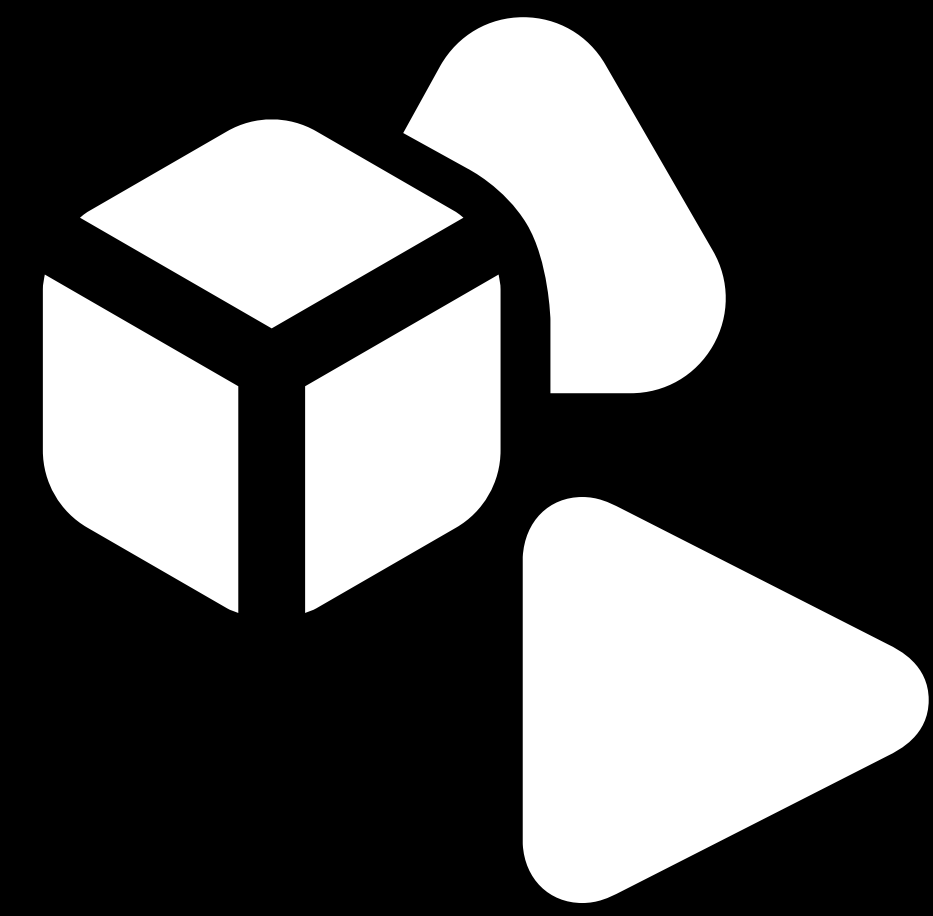
Anatomy of a Scene



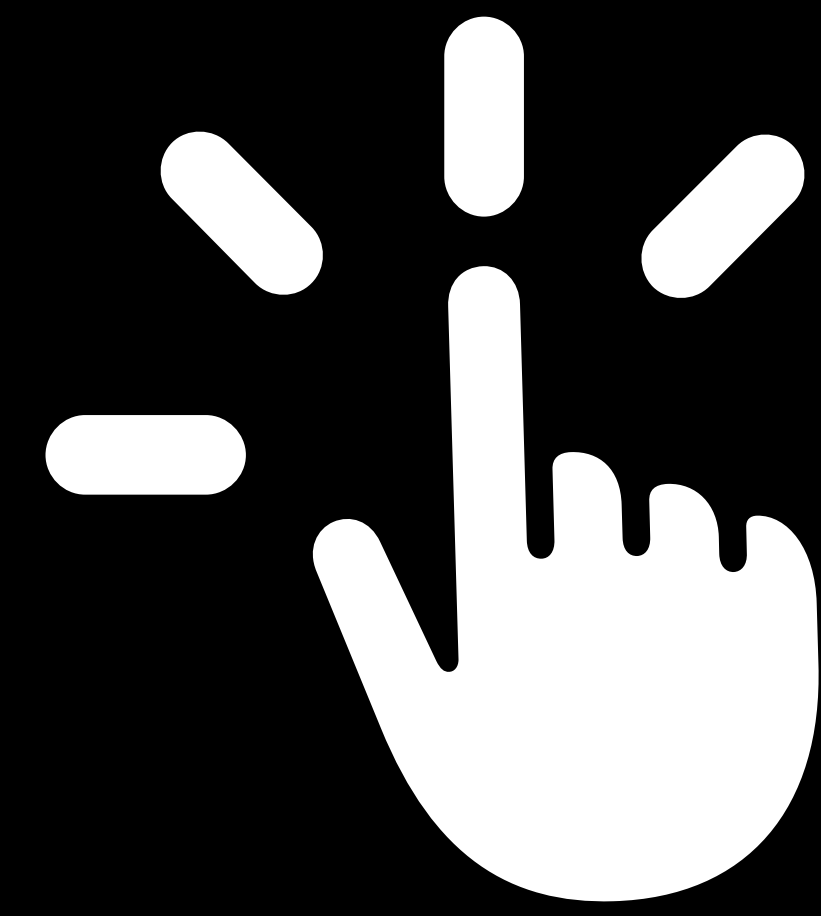
Behaviors



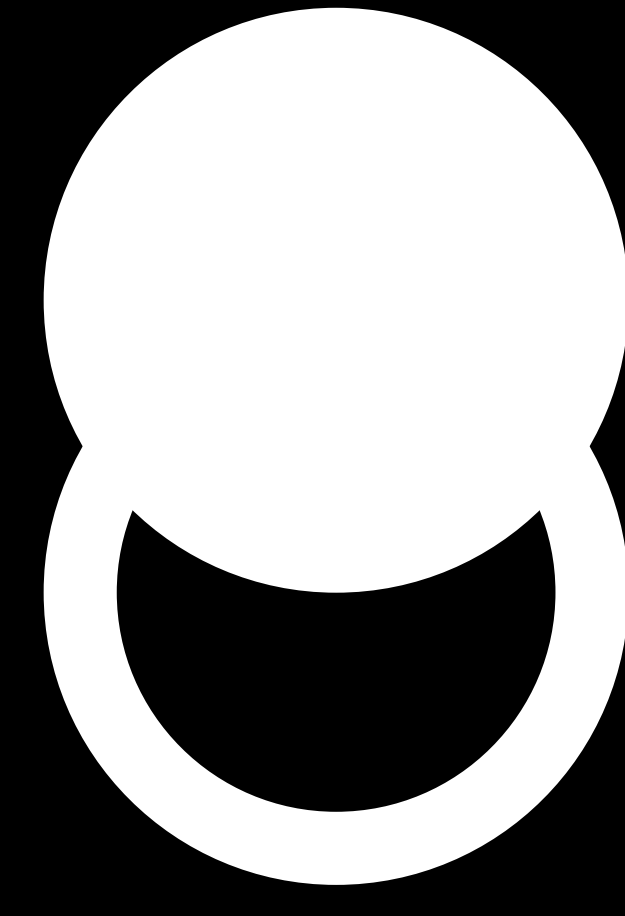
Triggers



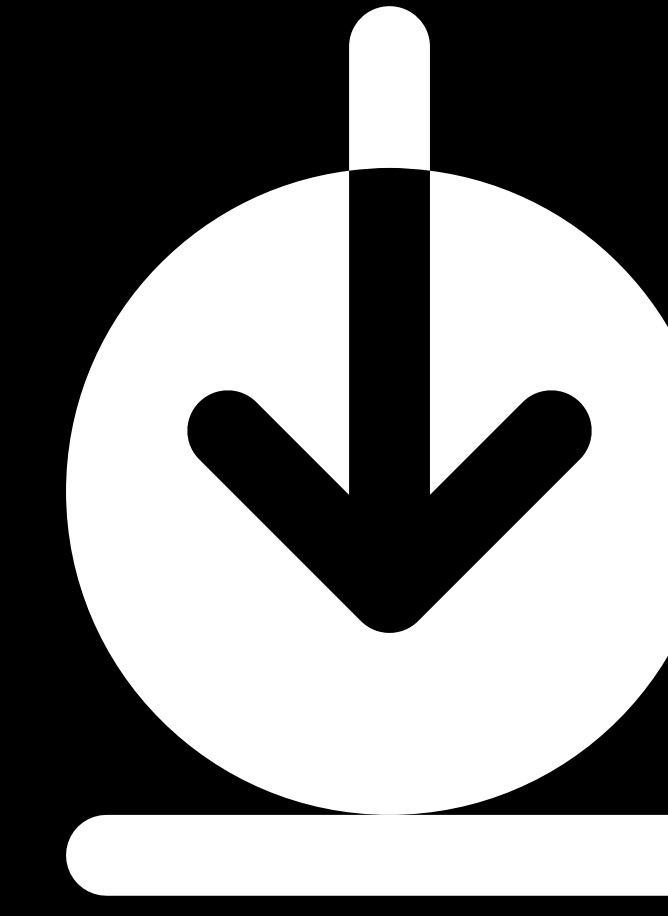
Start



Tap



Proximity

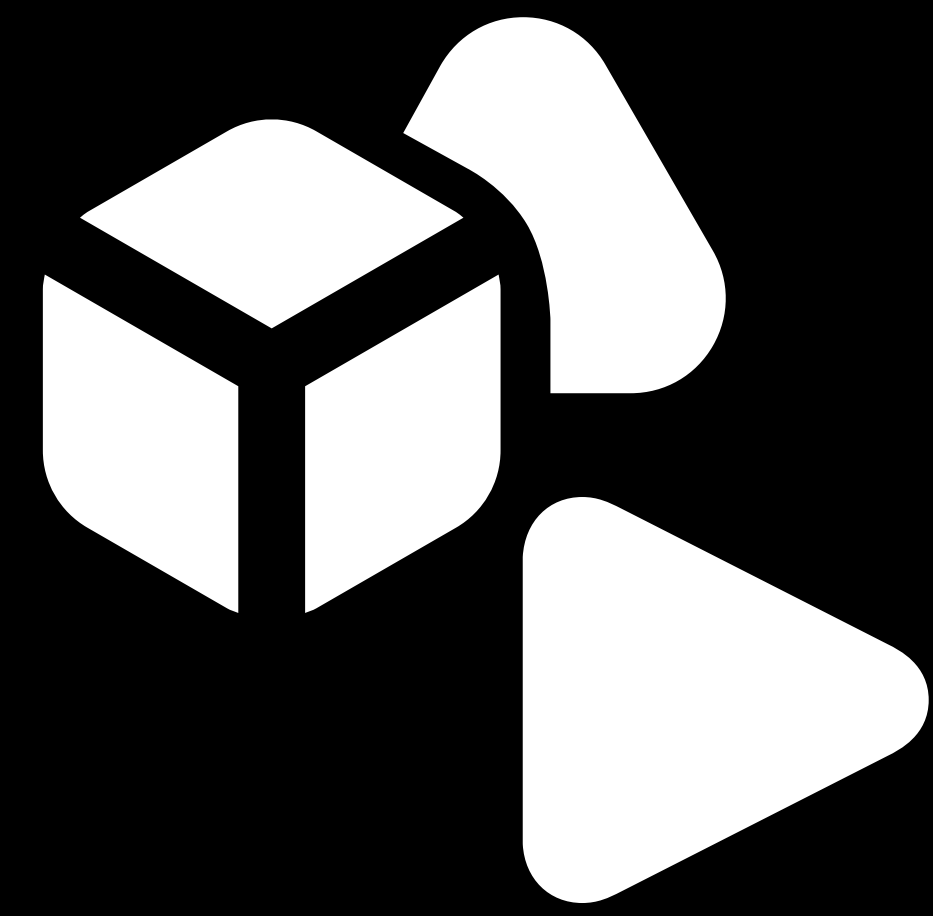


Collision

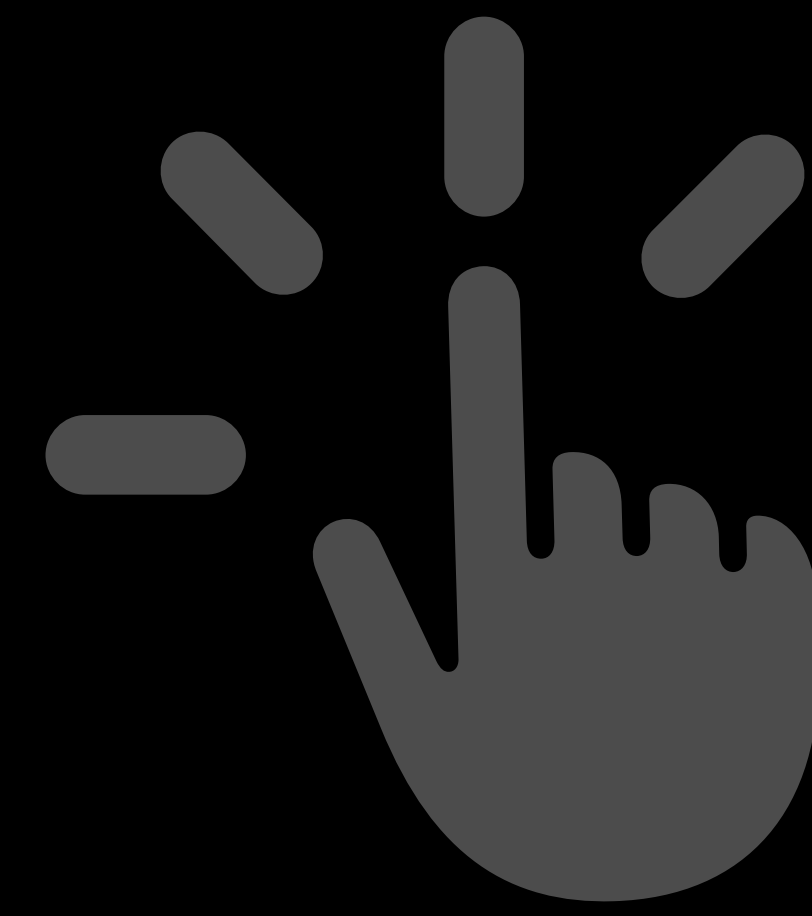


Notification

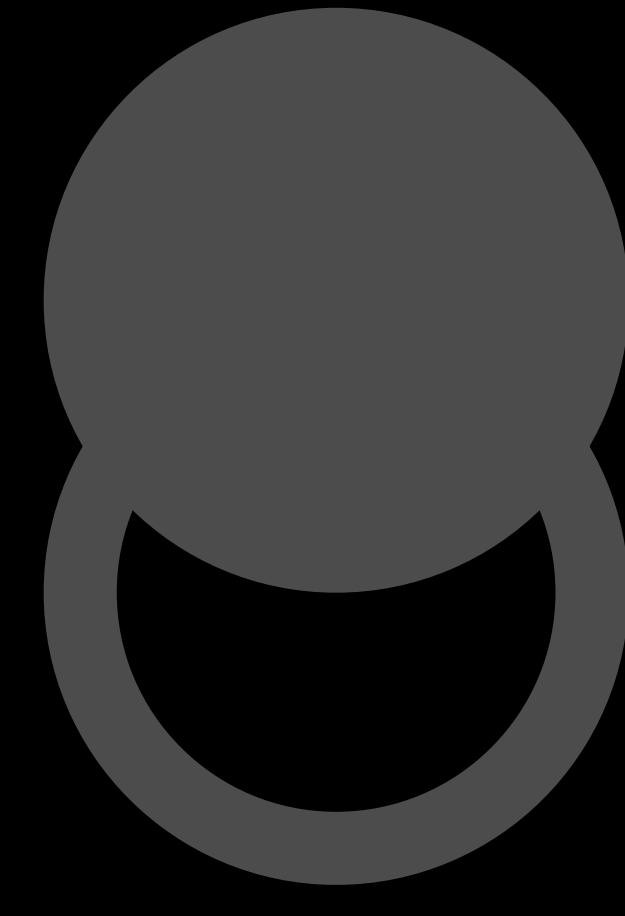
Triggers



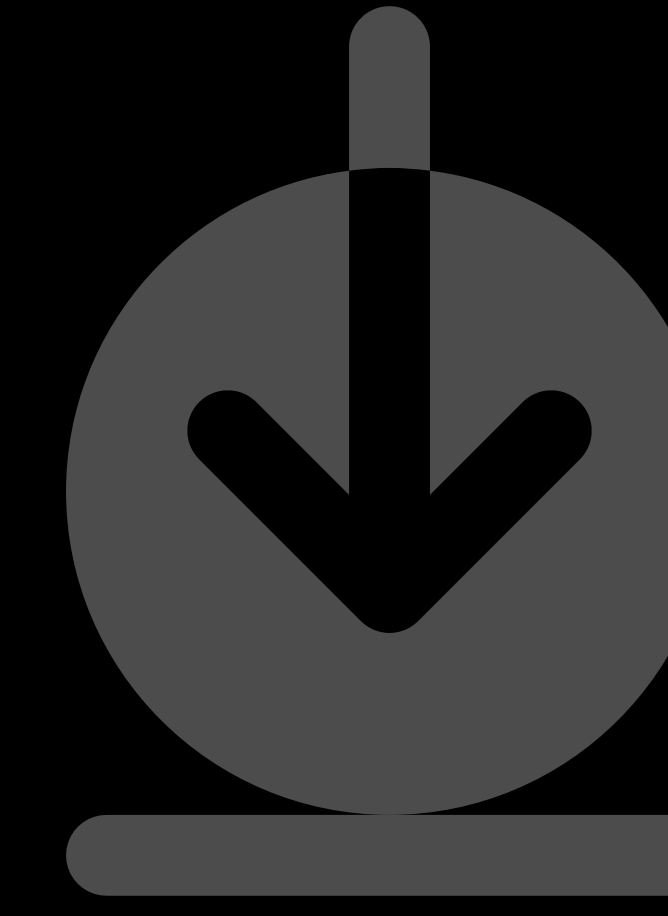
Start



Tap



Proximity

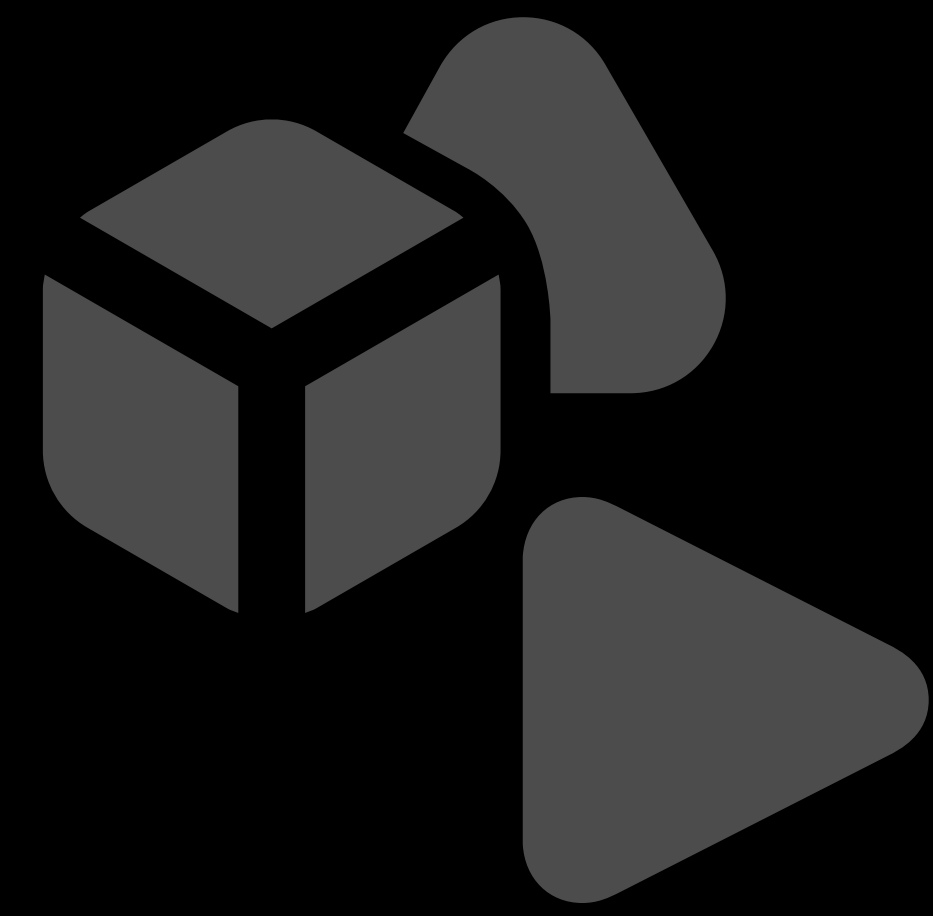


Collision

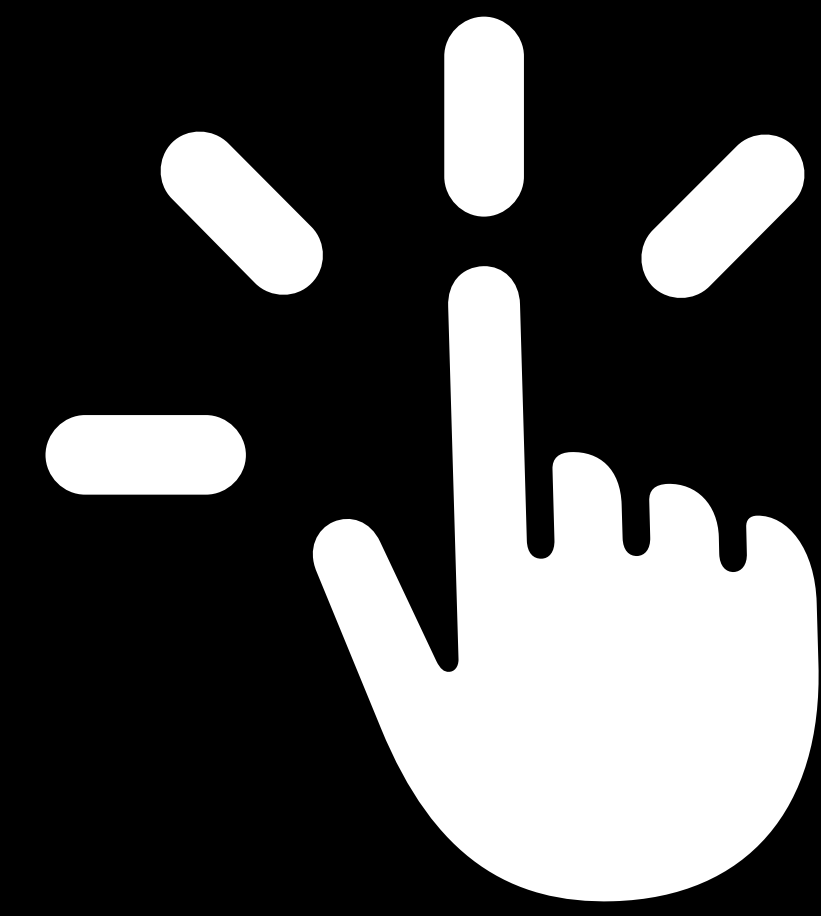


Notification

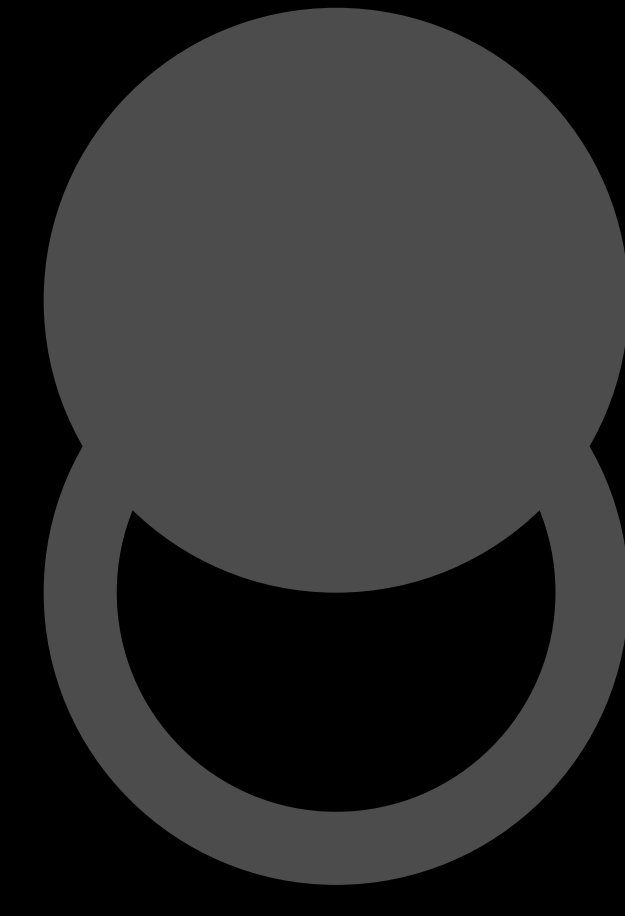
Triggers



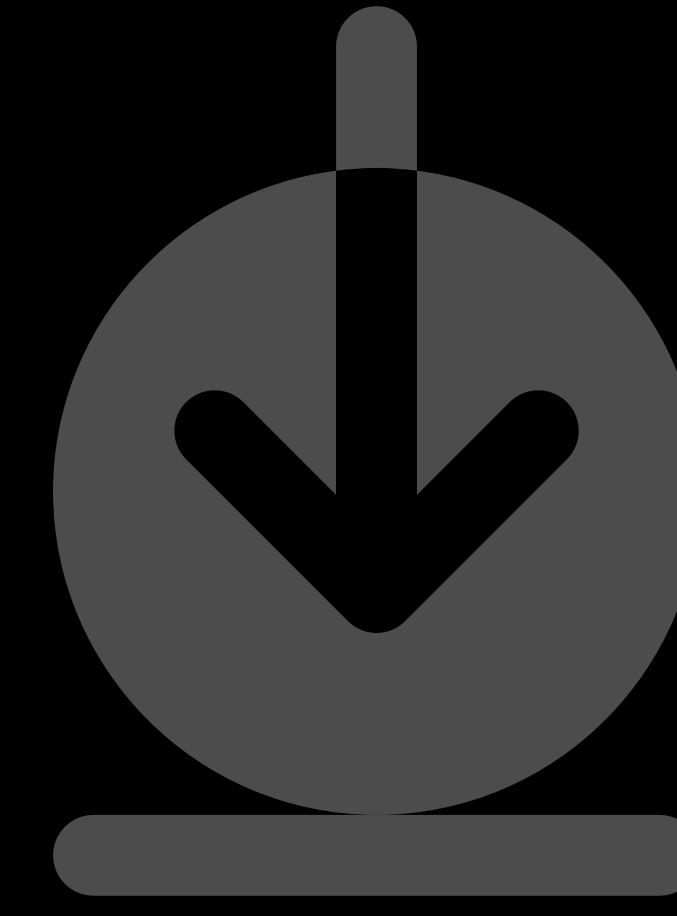
Start



Tap



Proximity

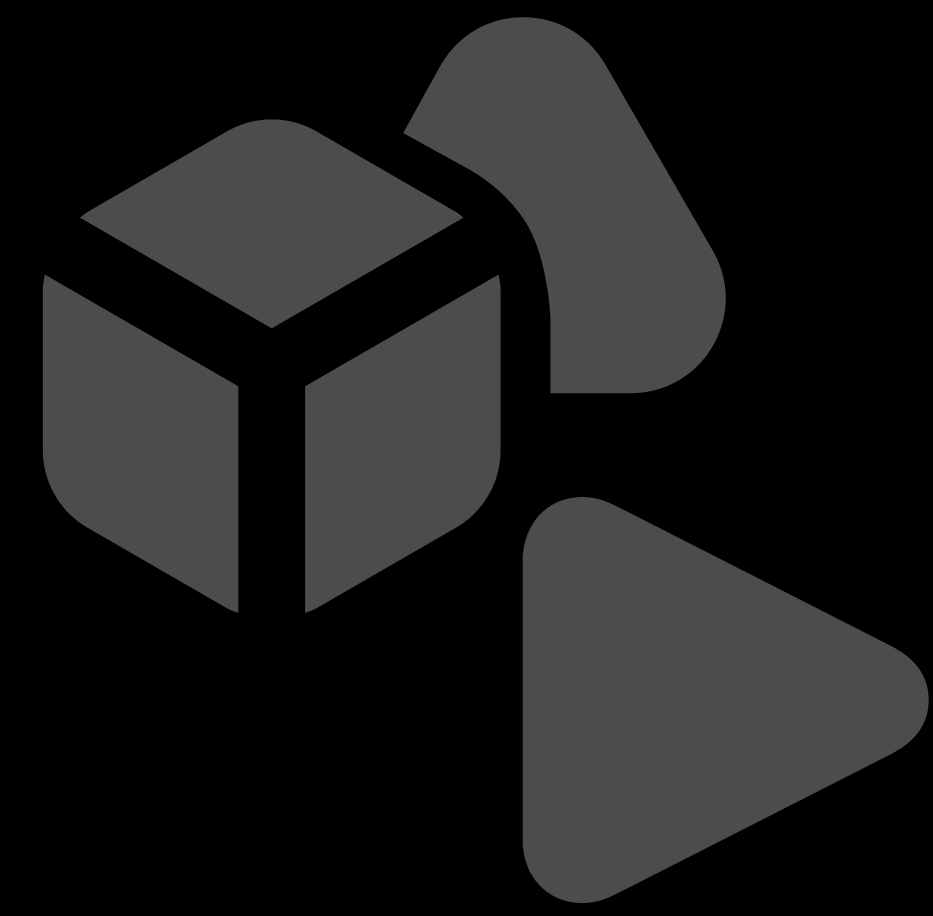


Collision

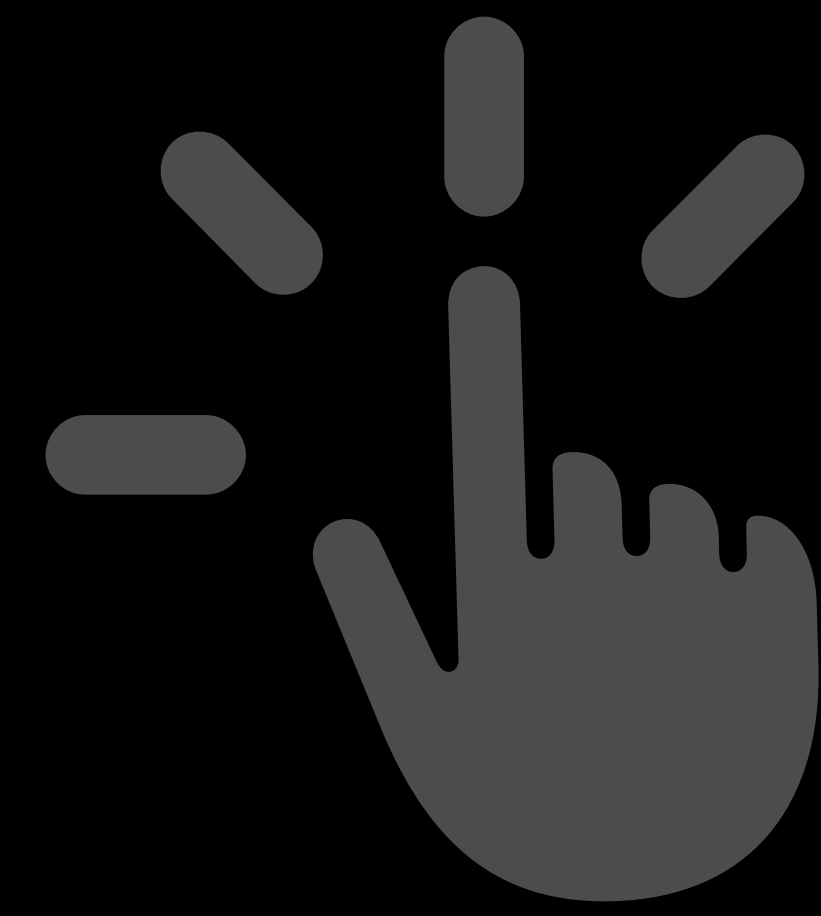


Notification

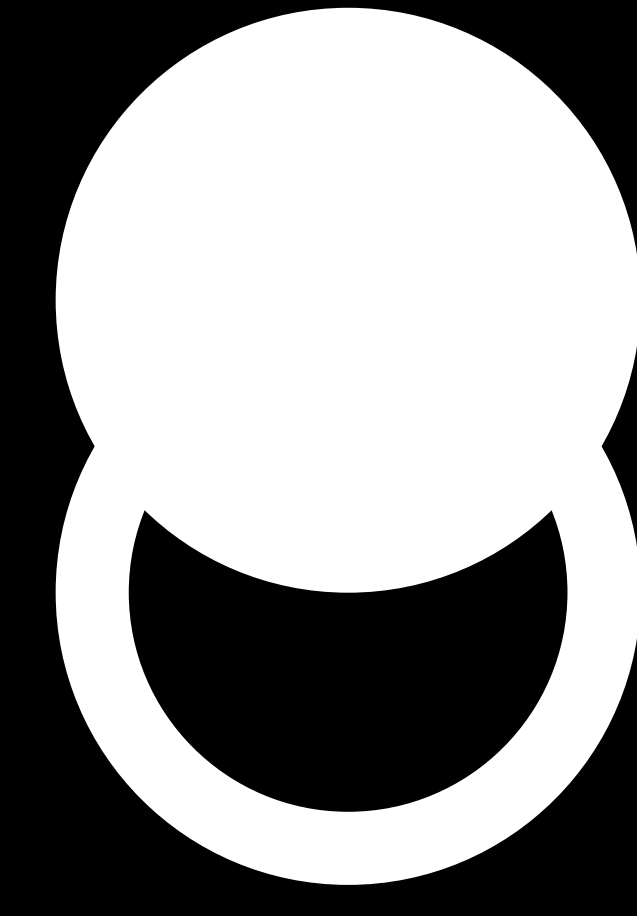
Triggers



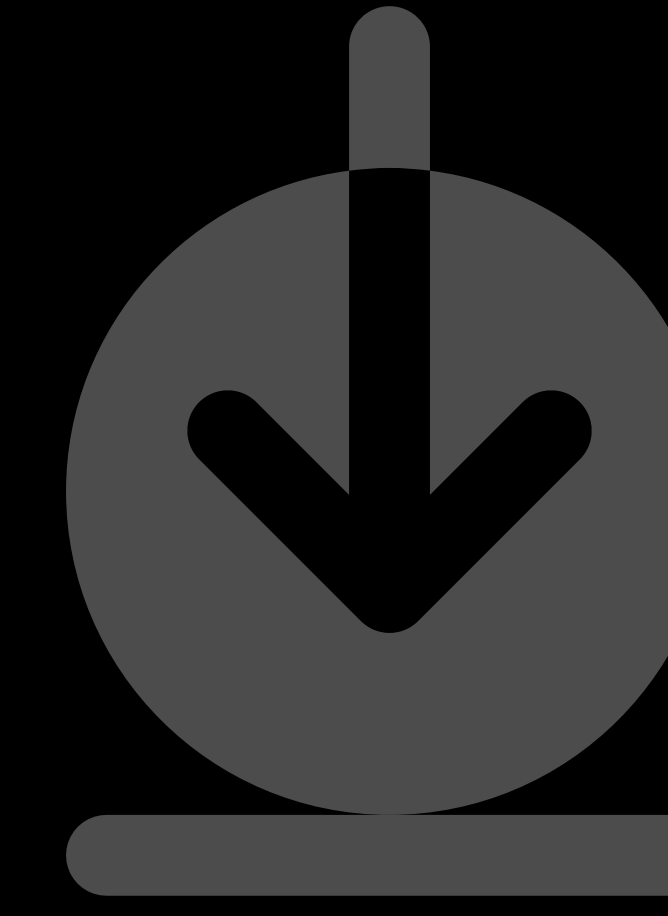
Start



Tap



Proximity

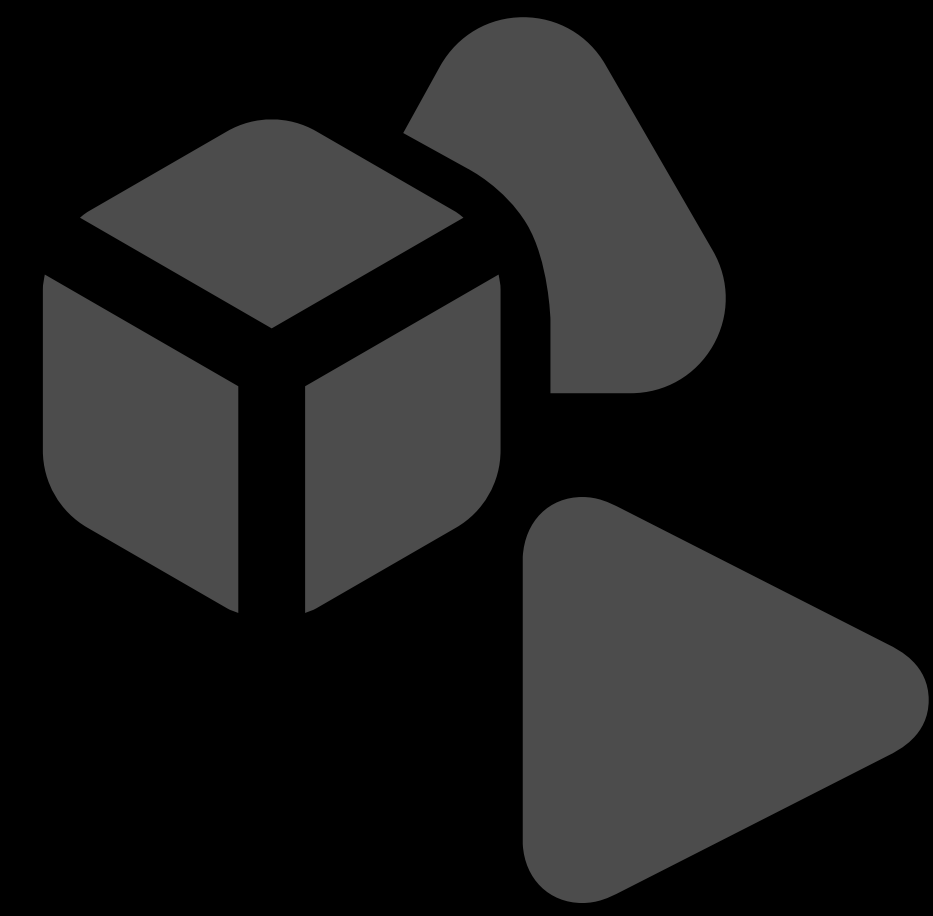


Collision

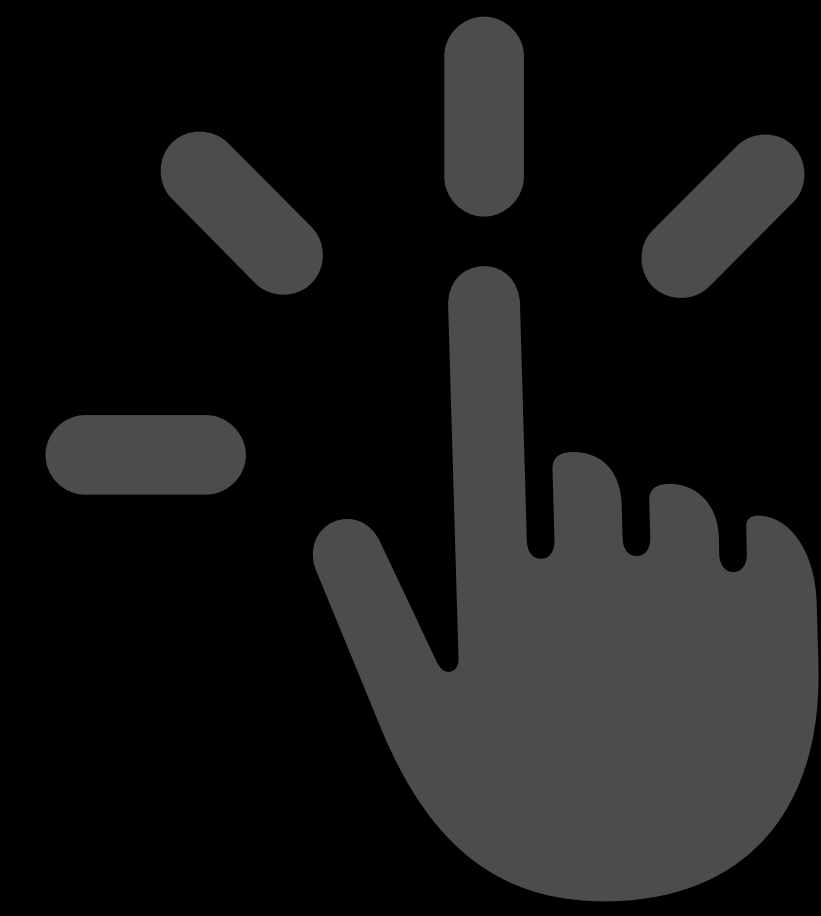


Notification

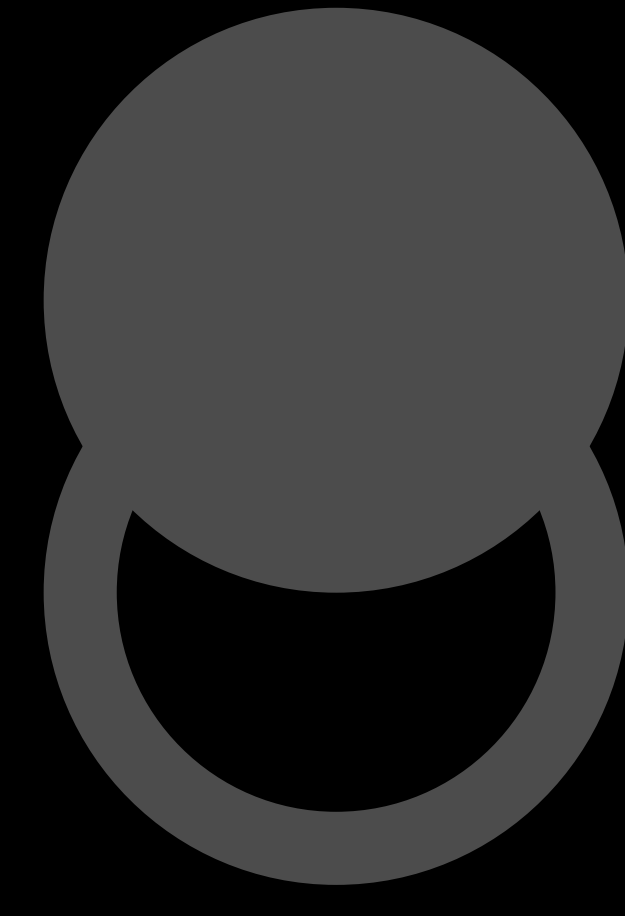
Triggers



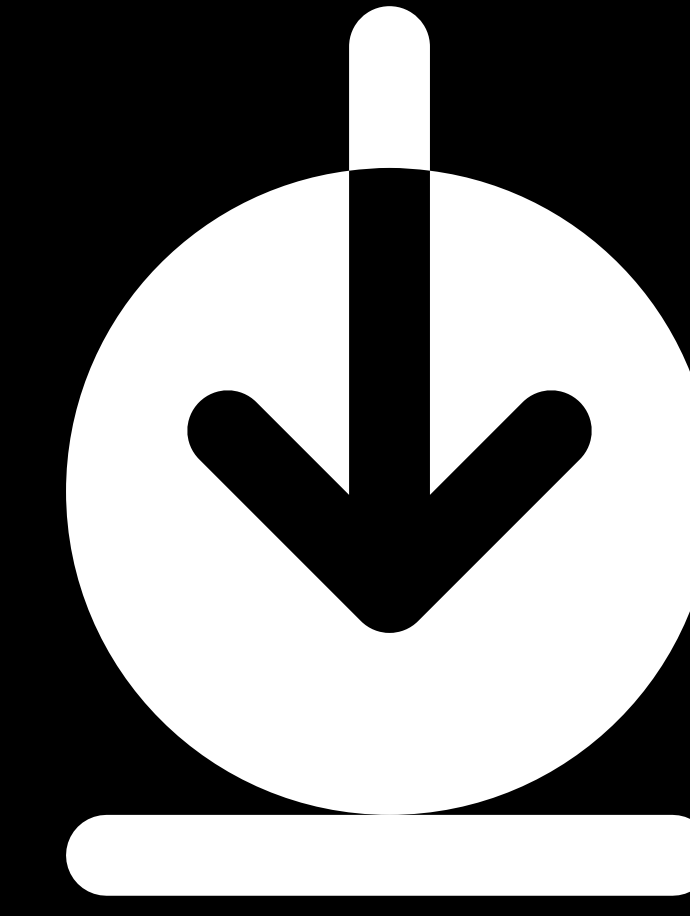
Start



Tap



Proximity

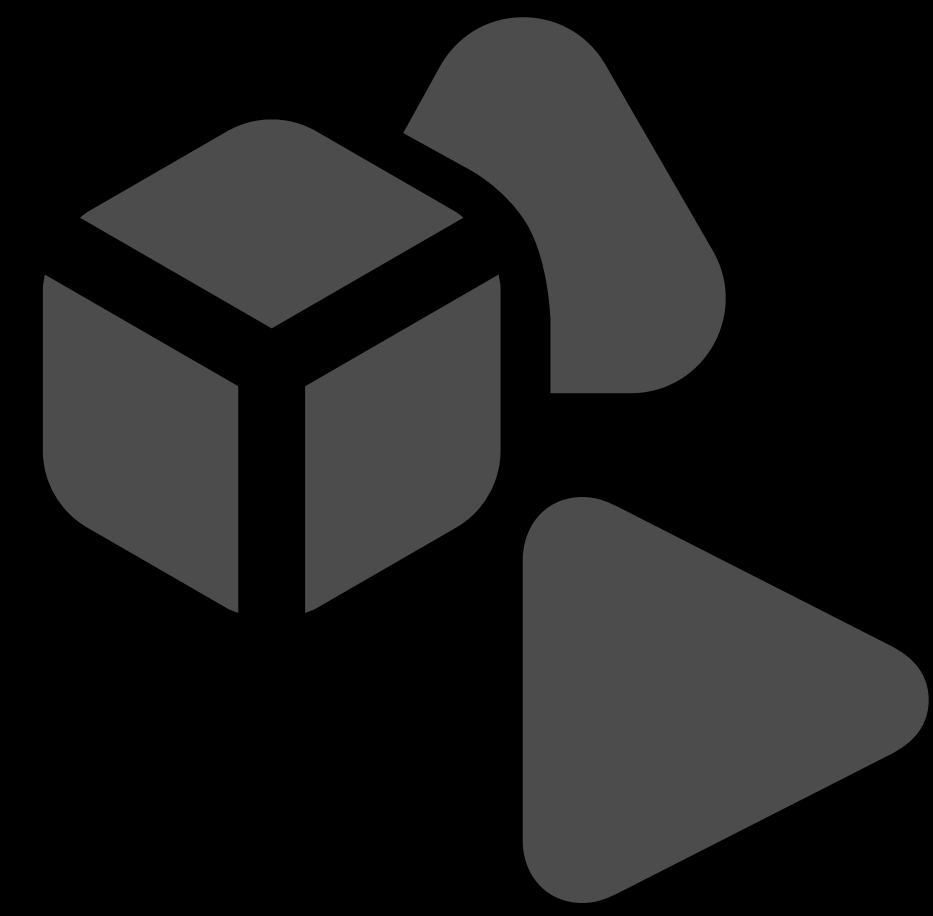


Collision

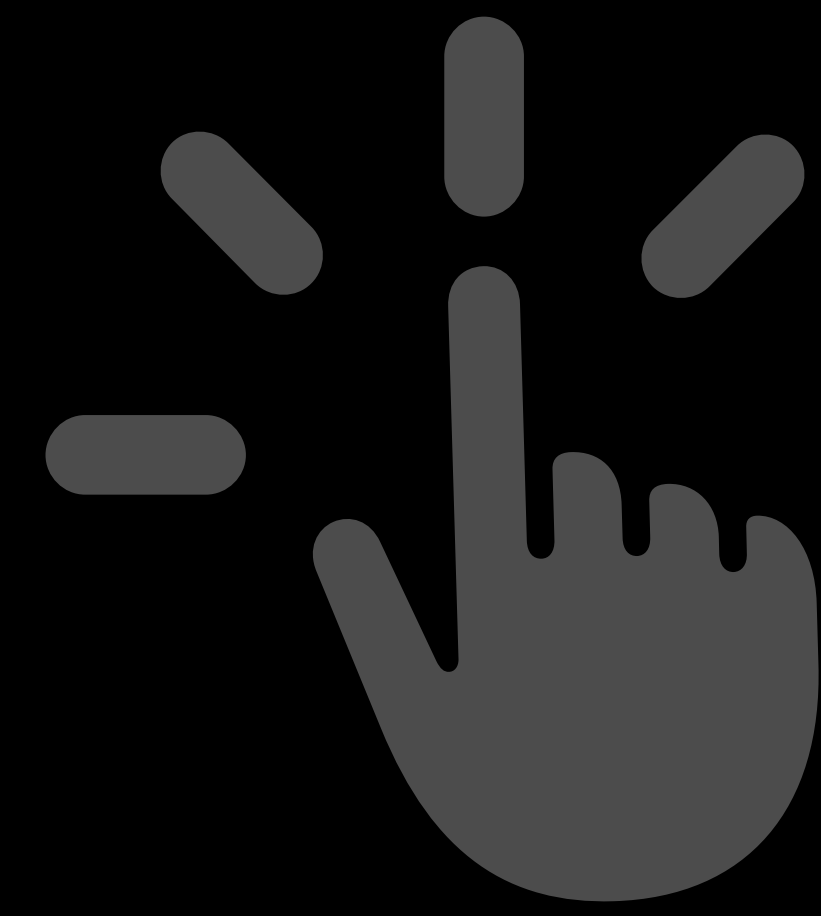


Notification

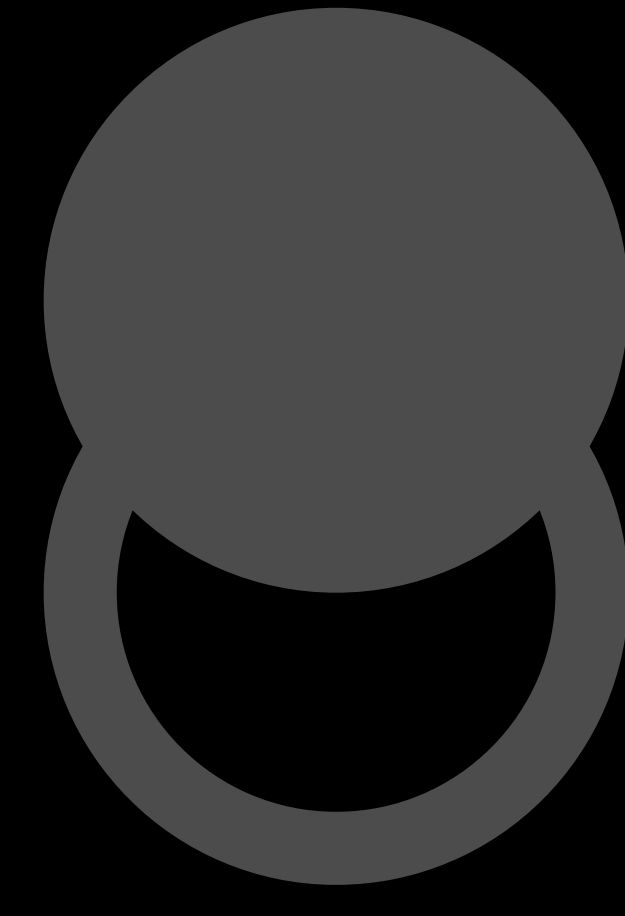
Triggers



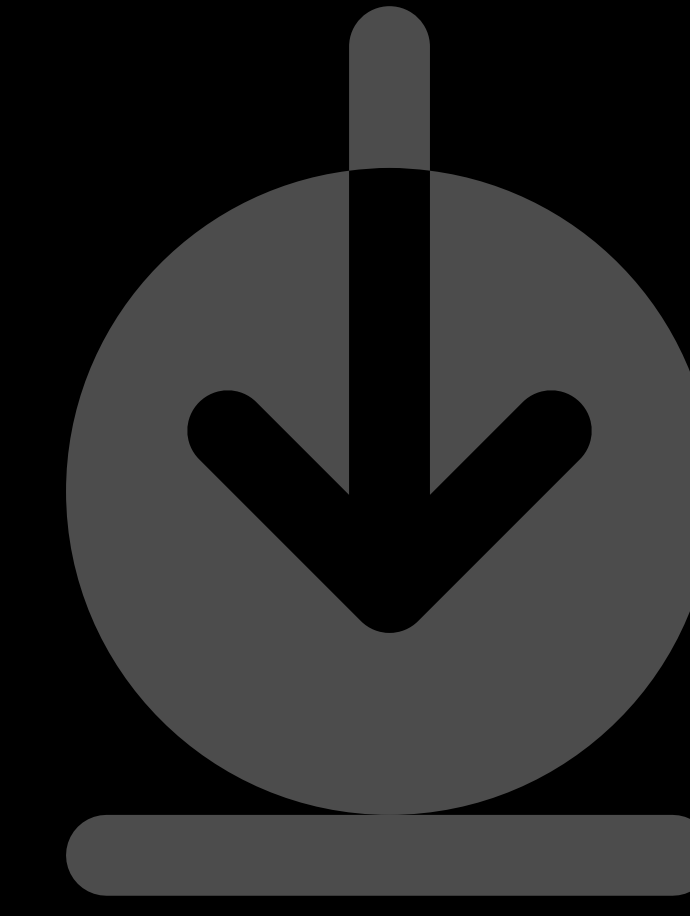
Start



Tap



Proximity



Collision



Notification

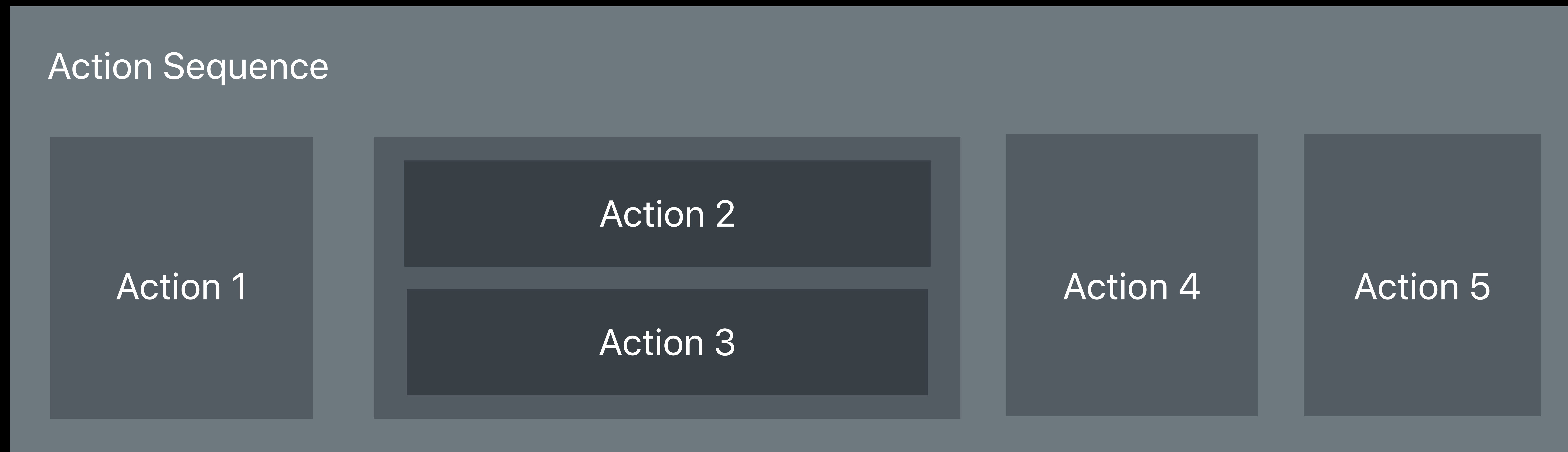
Action Sequence

Groups

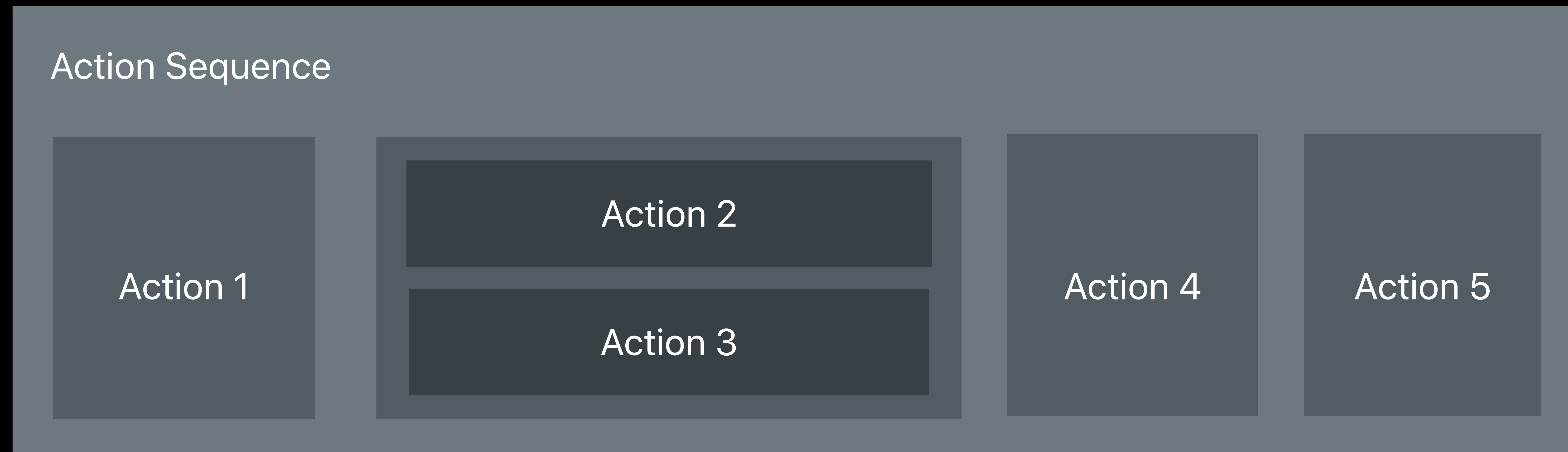
Looping

Exclusive action sequences

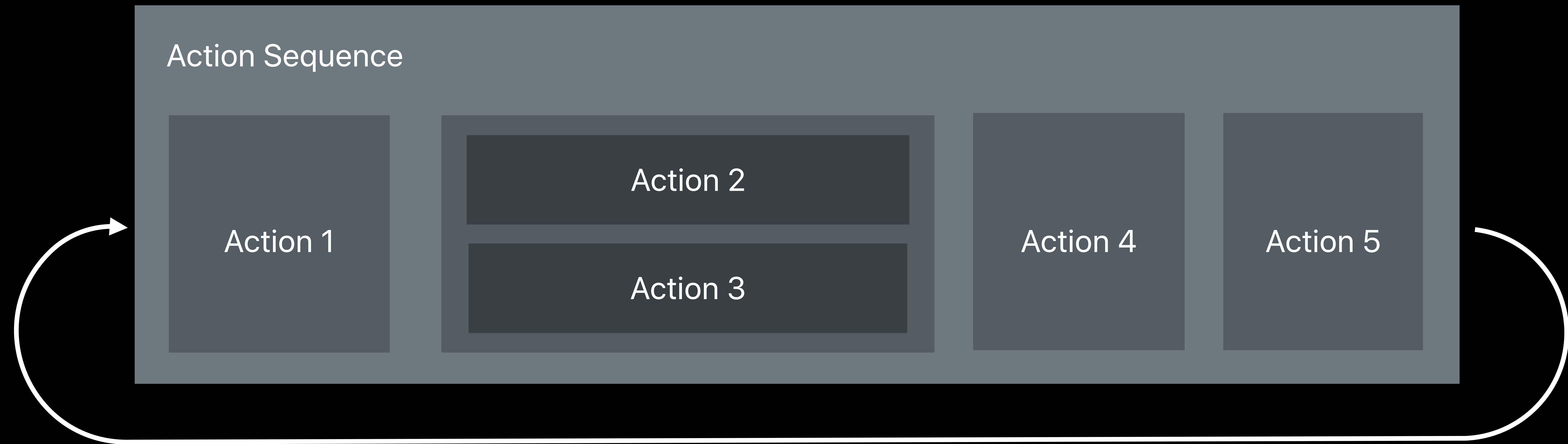
Groups



Looping

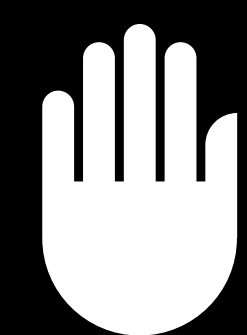


Looping



Exclusive Action Sequences

Exclusive
Action Sequence 1



Exclusive
Action Sequence 2



Action Sequence 3



Time



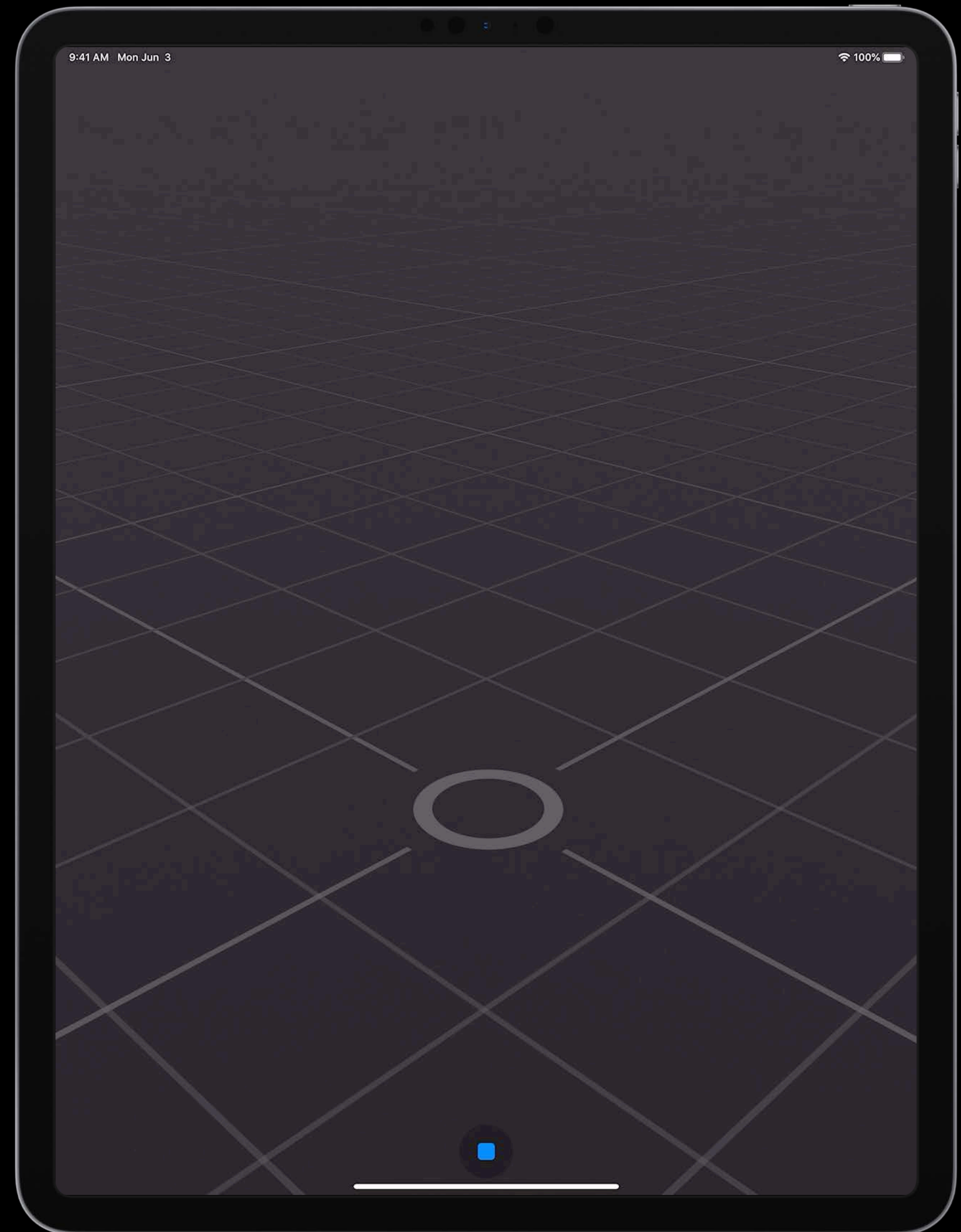
Visibility Actions



Show



Hide



Visibility Actions



Show



Hide



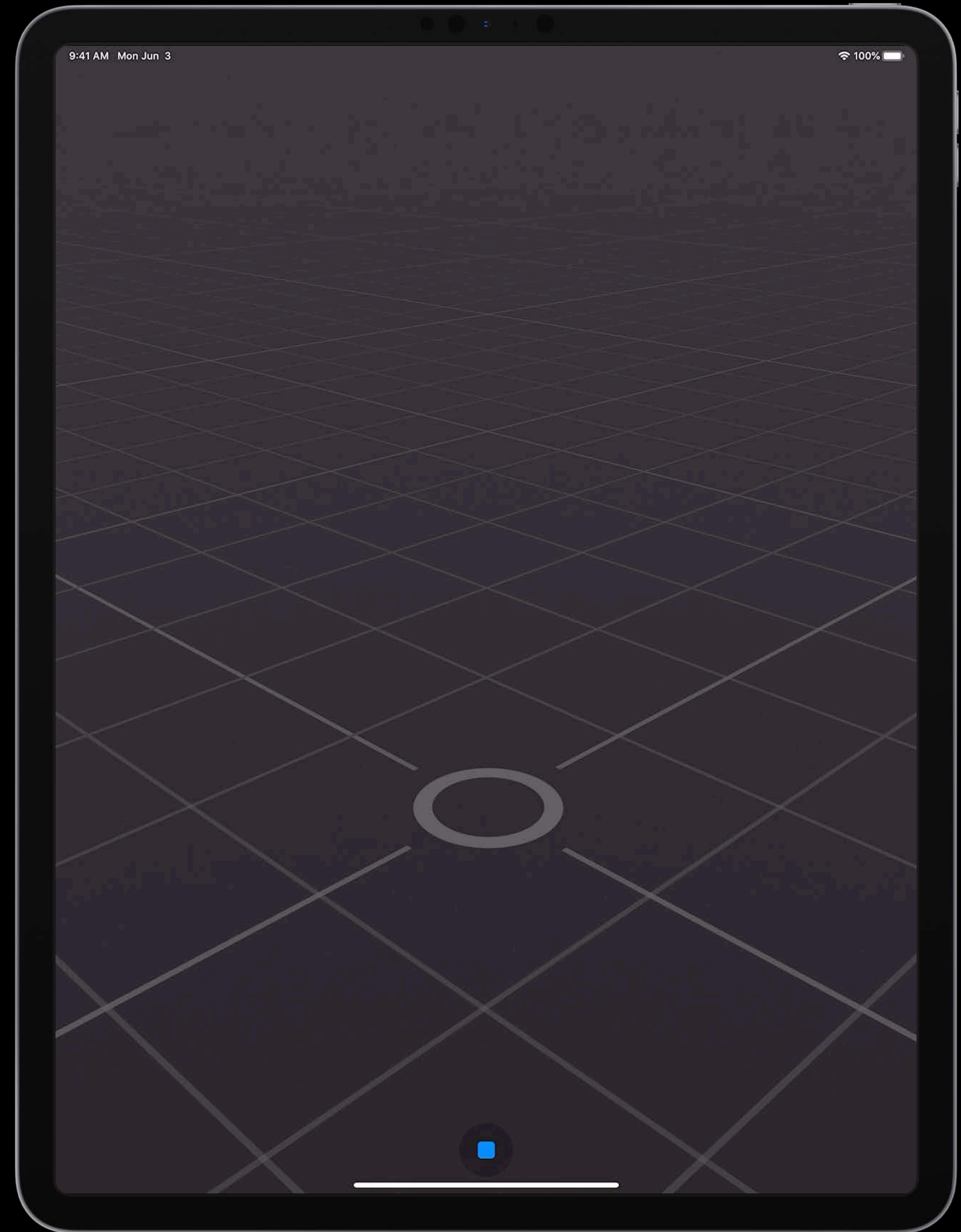
Visibility Actions



Show



Hide



Visibility Actions



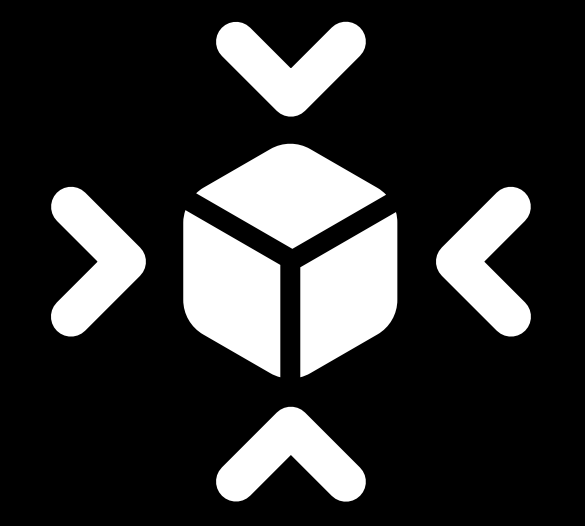
Show



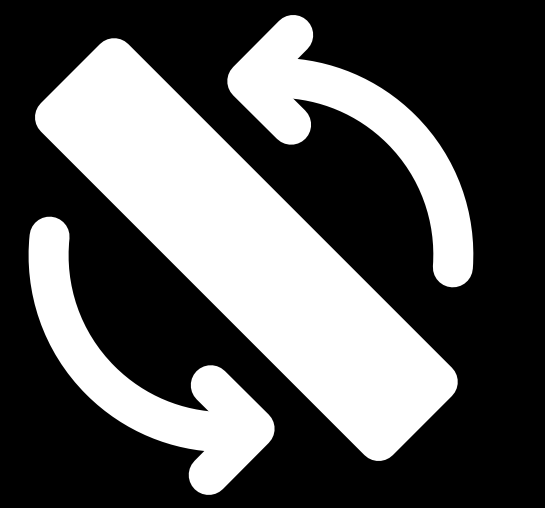
Hide



Animate Actions



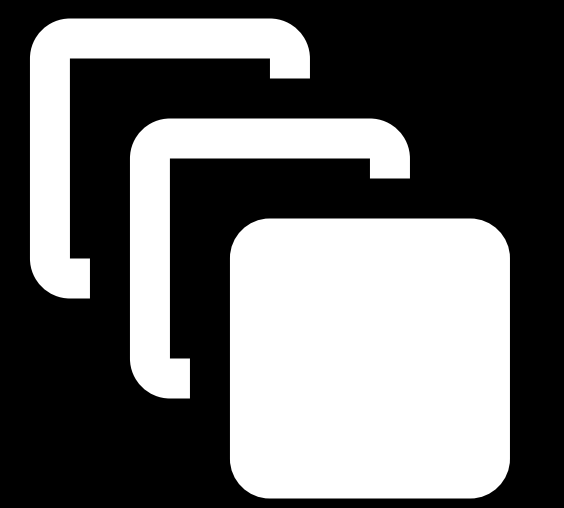
Emphasis



Spin



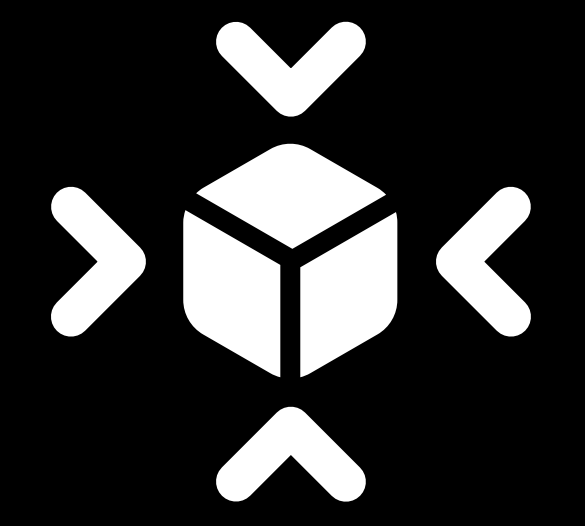
Orbit



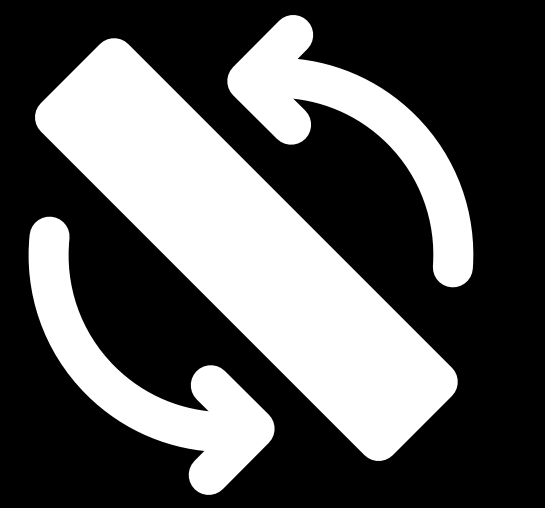
Usdz Animation



Animate Actions



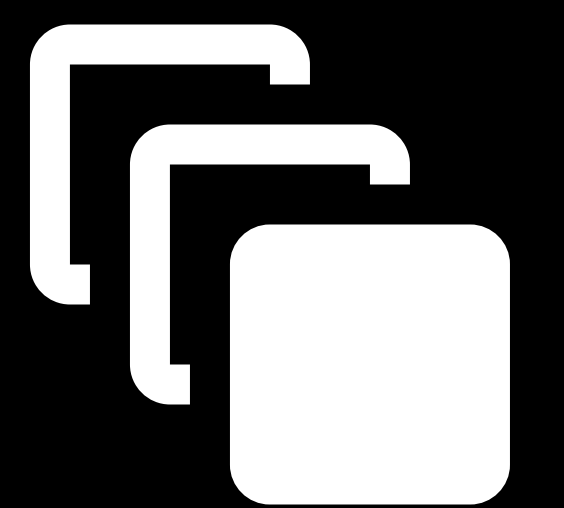
Emphasis



Spin



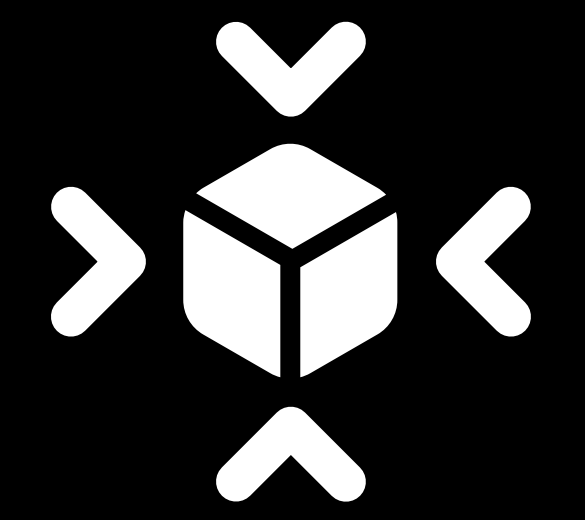
Orbit



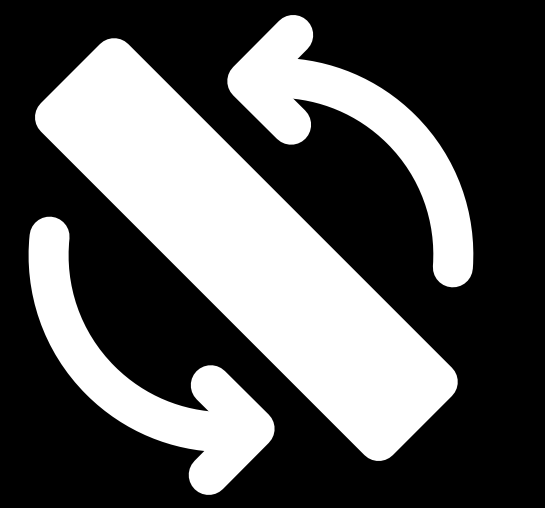
Usdz Animation



Animate Actions



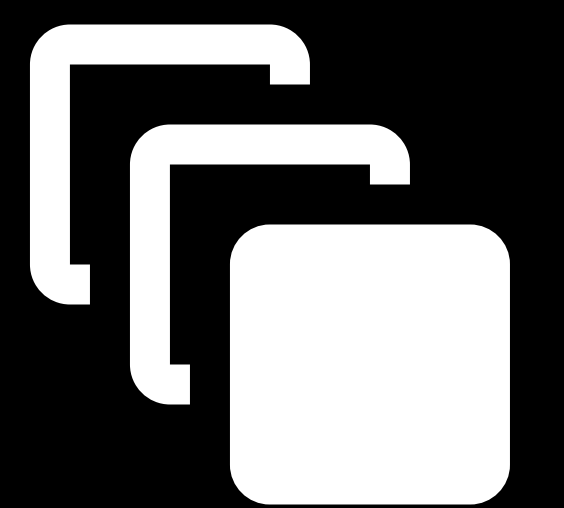
Emphasis



Spin



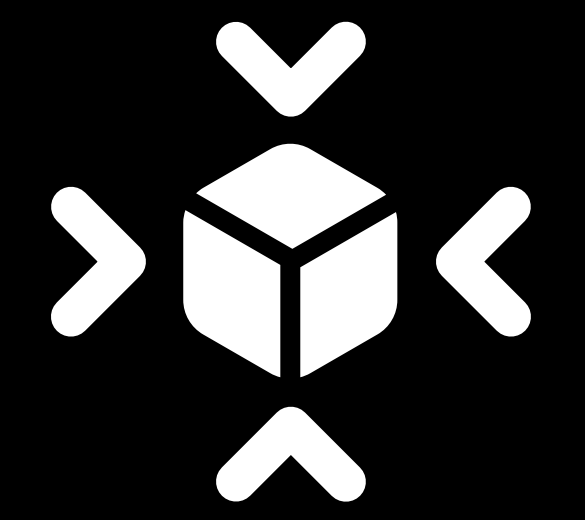
Orbit



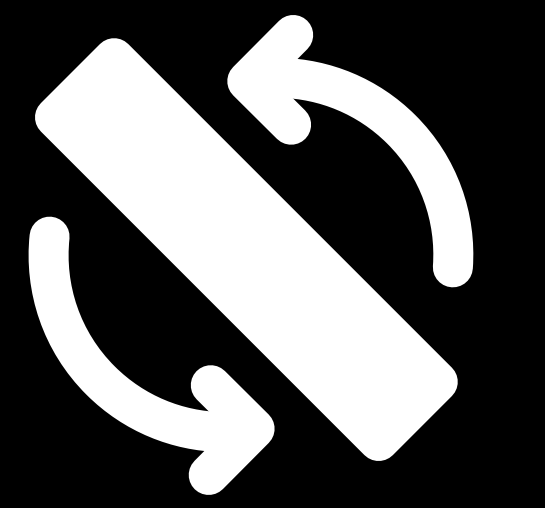
Usdz Animation



Animate Actions



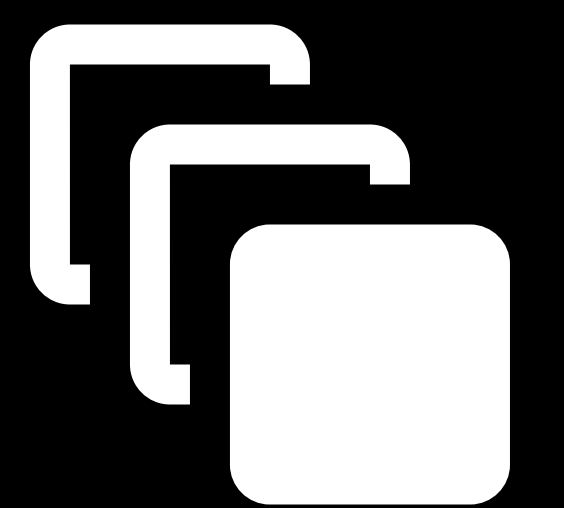
Emphasis



Spin



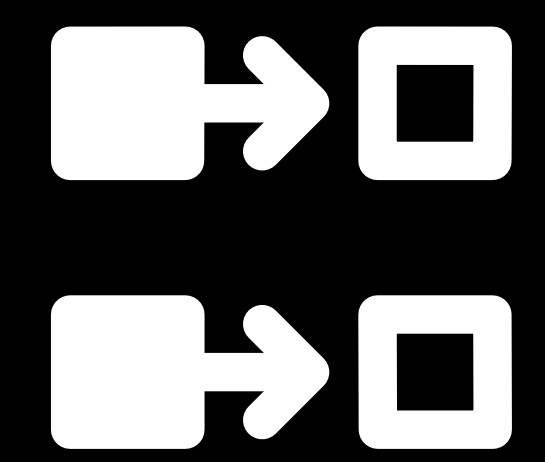
Orbit



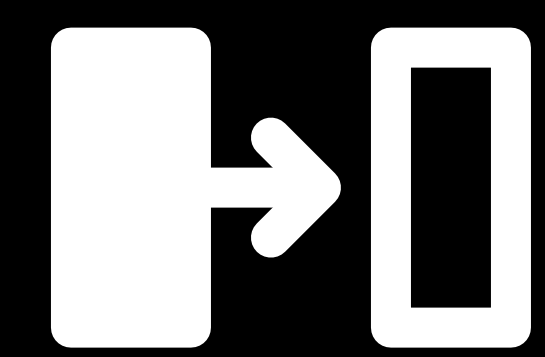
Usdz Animation



Move Actions



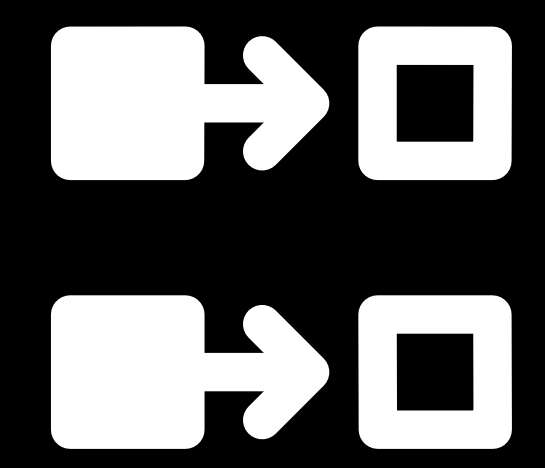
Move By



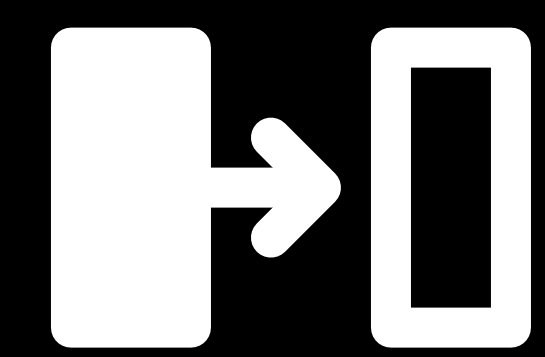
Move To



Move Actions



Move By



Move To



Move Actions

  Move By

 Move To



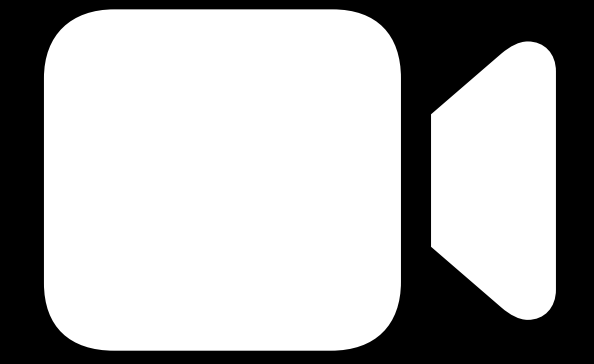
Move Actions

  Move By

 Move To



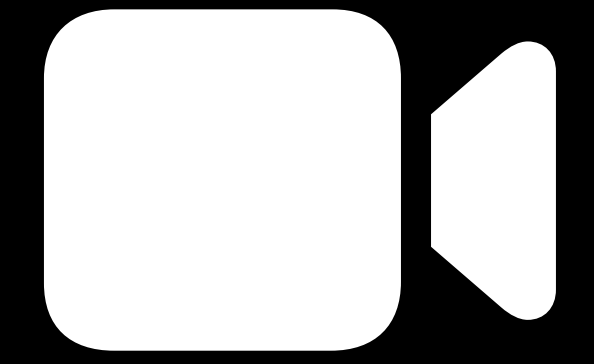
Look At Action



Look At



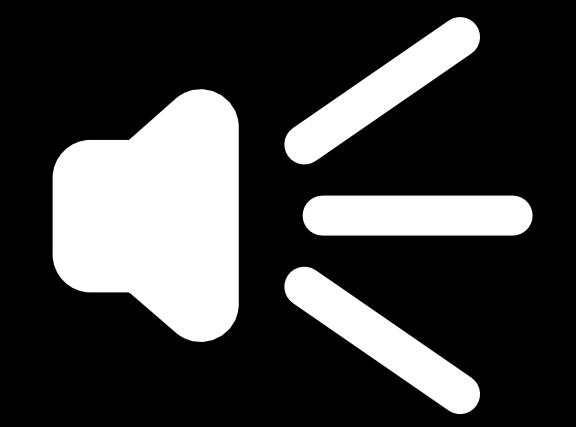
Look At Action



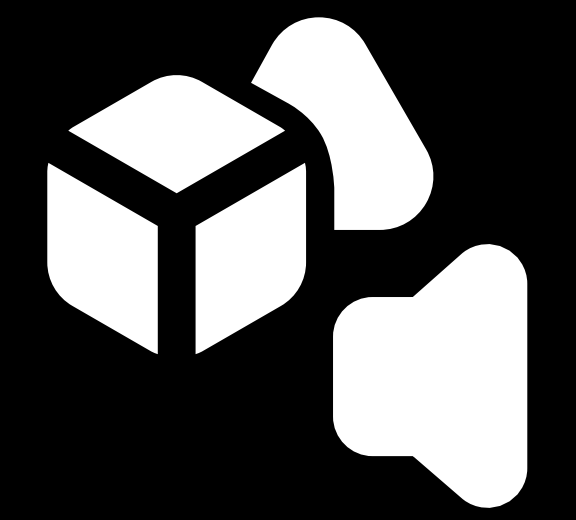
Look At



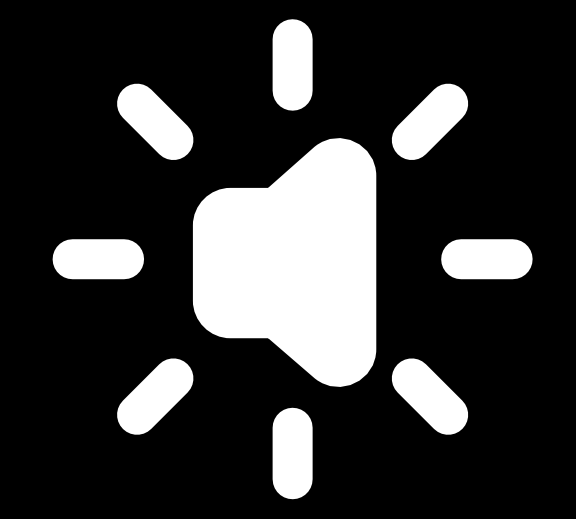
Audio Actions



Play Sound



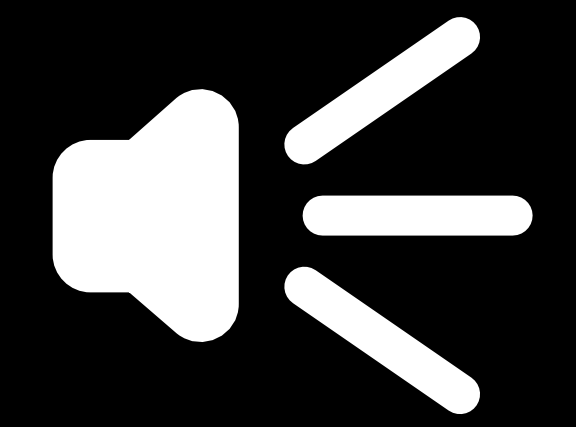
Play Ambient



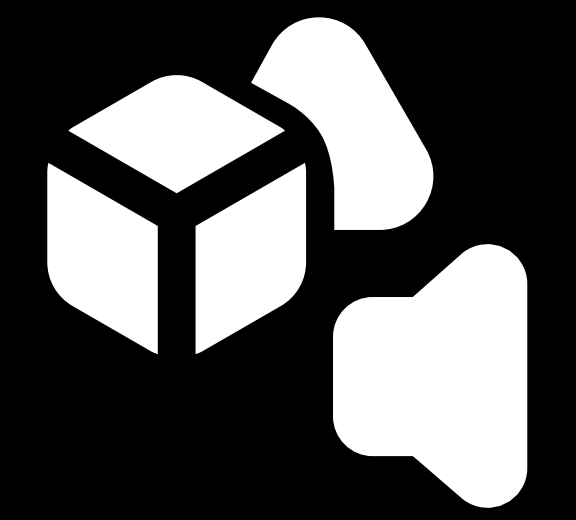
Play Music



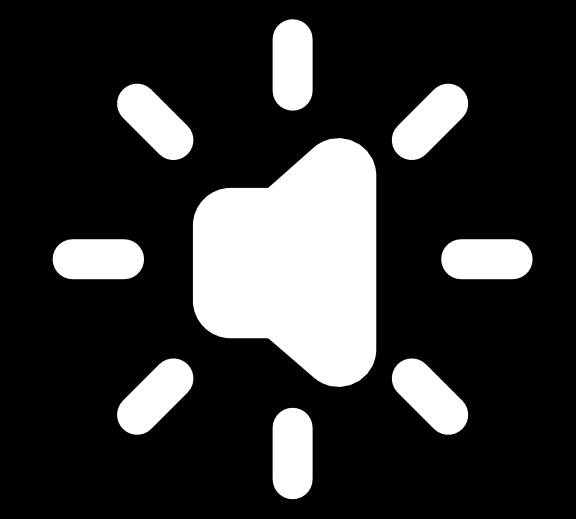
Audio Actions



Play Sound



Play Ambient



Play Music



Demo

Building Scenes

Adding Behaviors

Using Physics

Building Apps

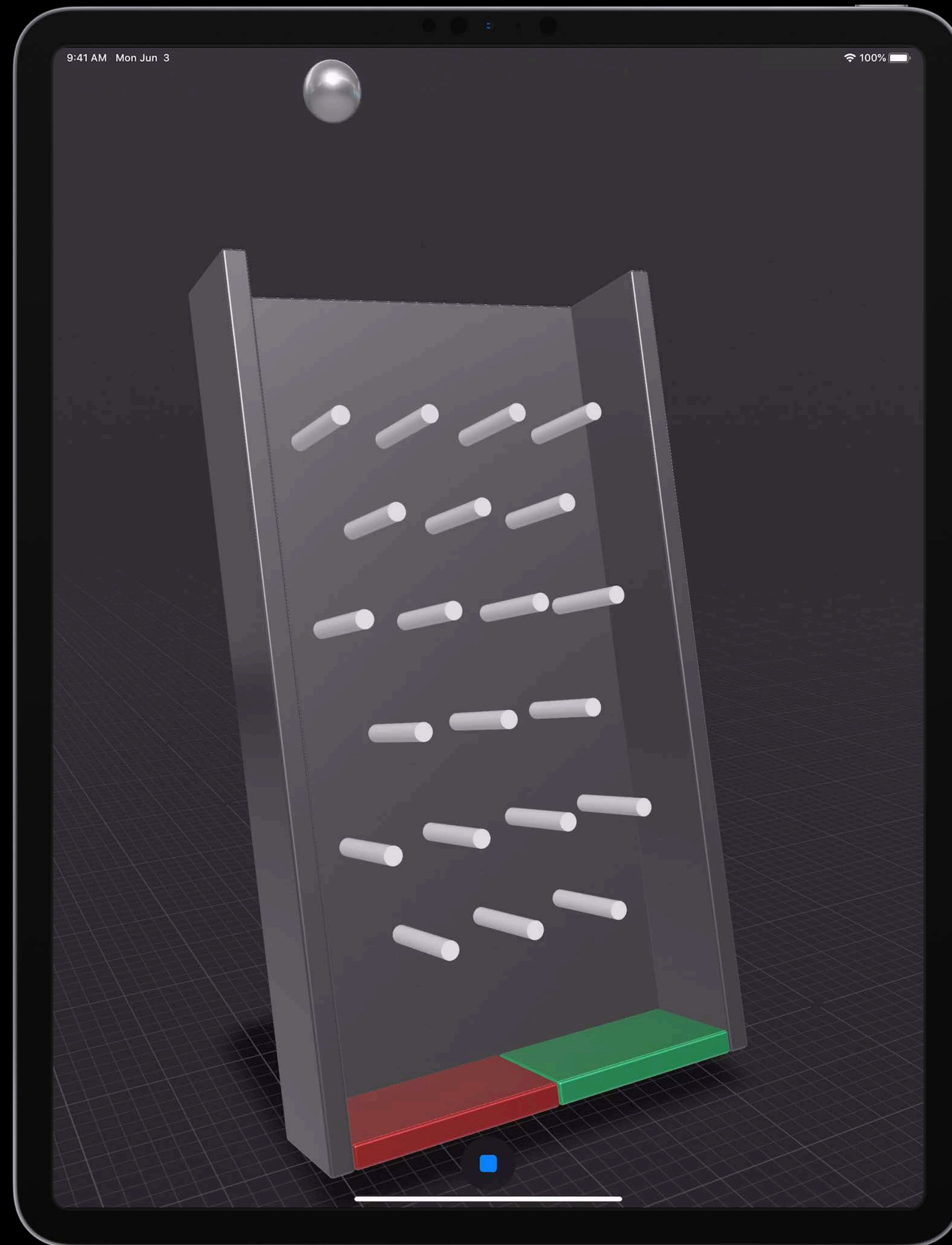
Building Scenes

Adding Behaviors

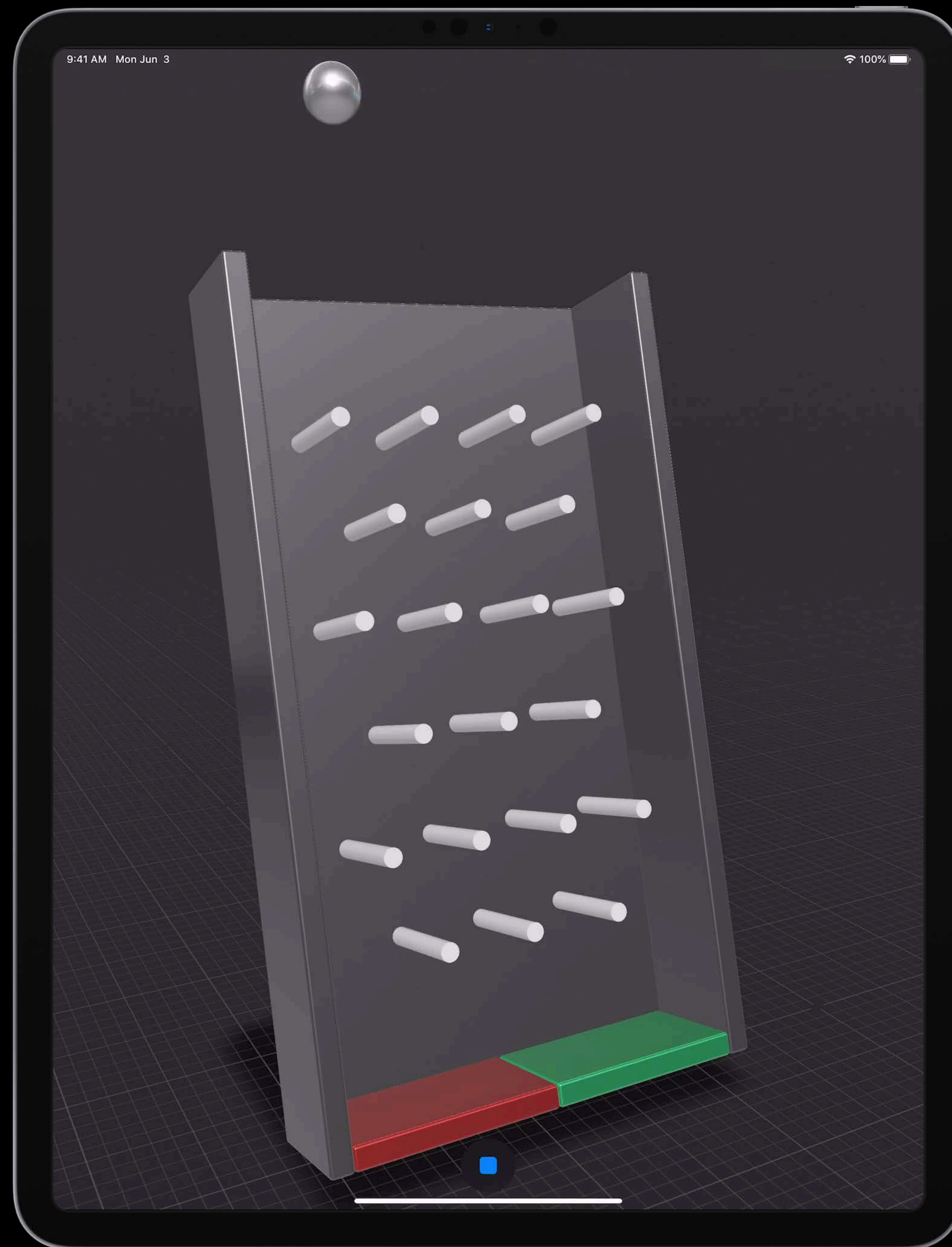
Using Physics

Building Apps

Physics



Physics



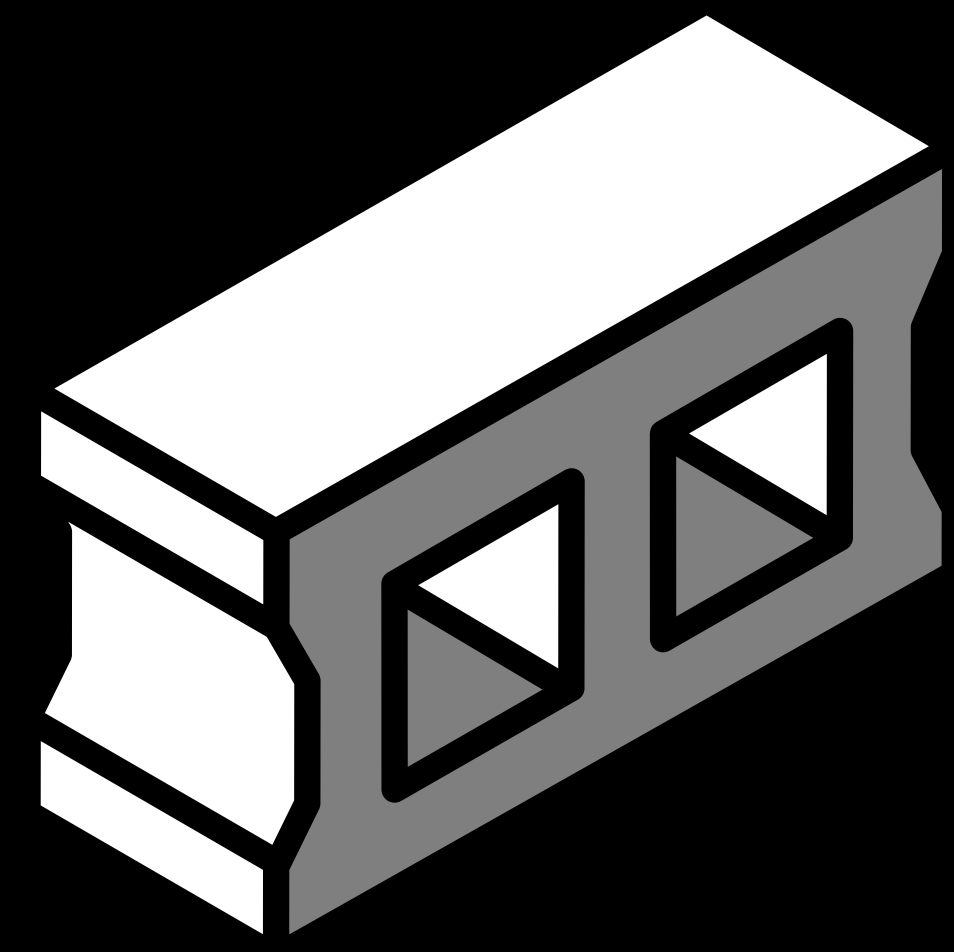
Physics

Physical materials

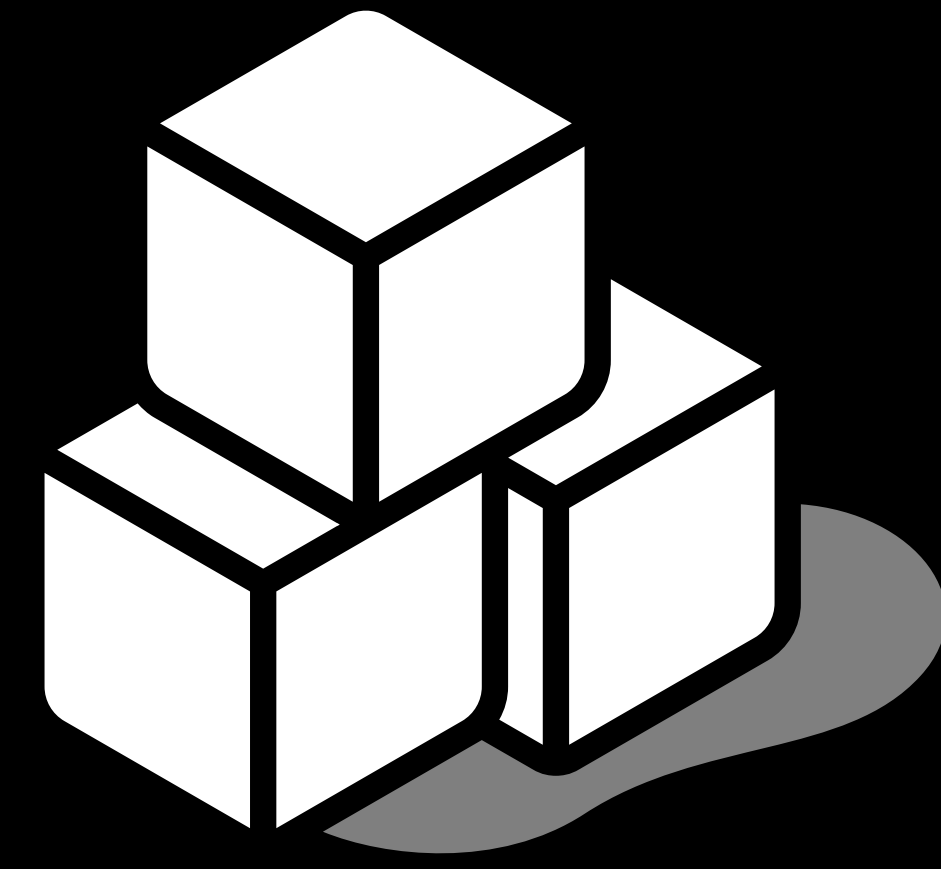
Forces

Collisions

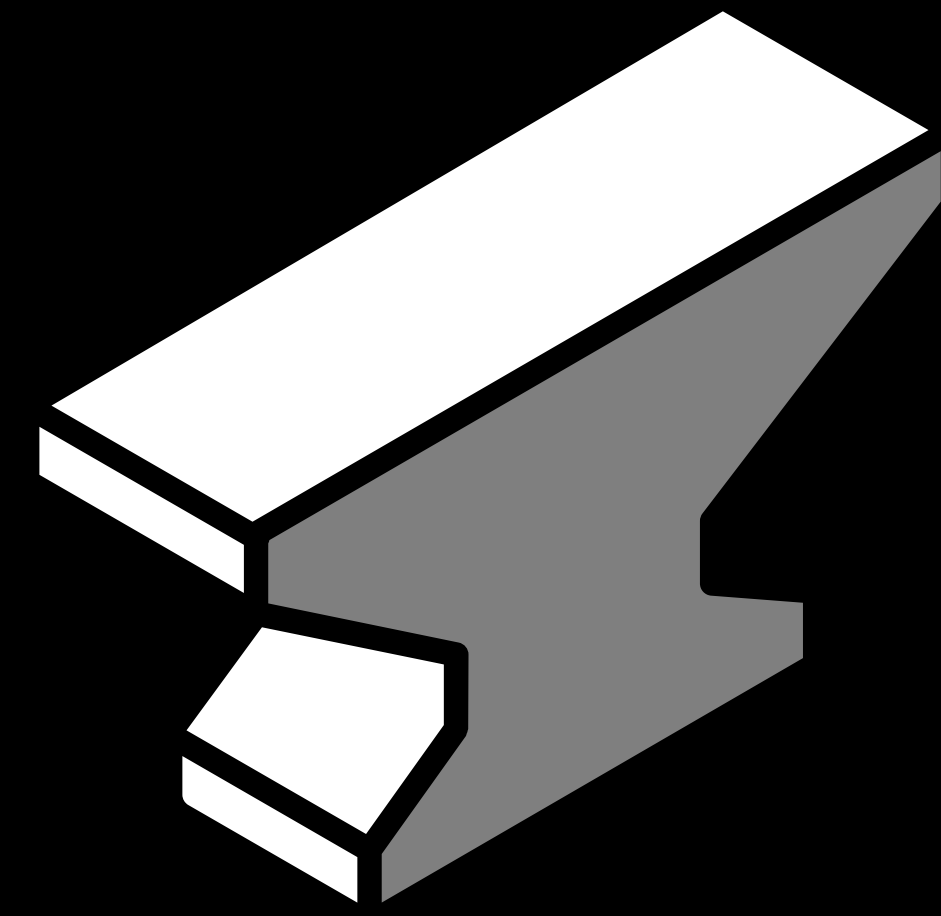
Materials



Concrete



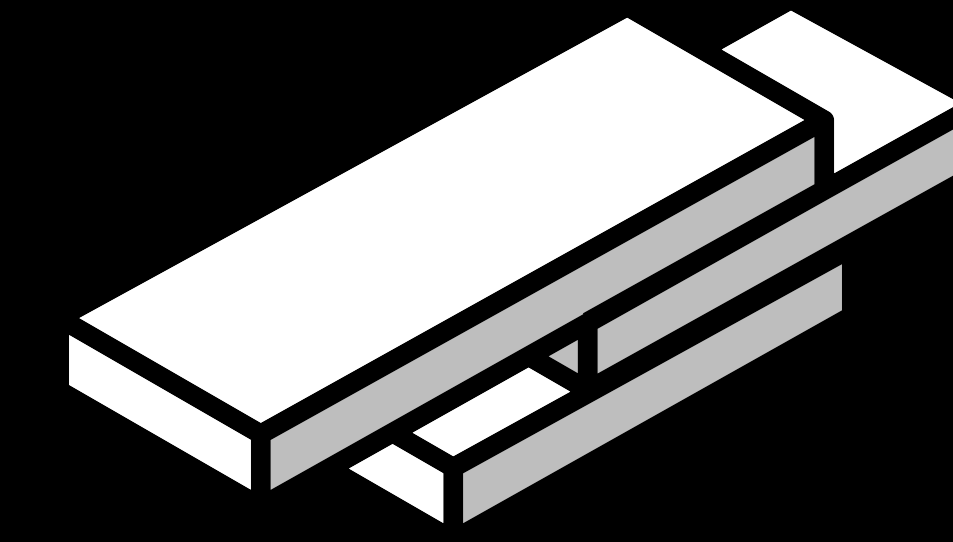
Ice



Lead



Plastic



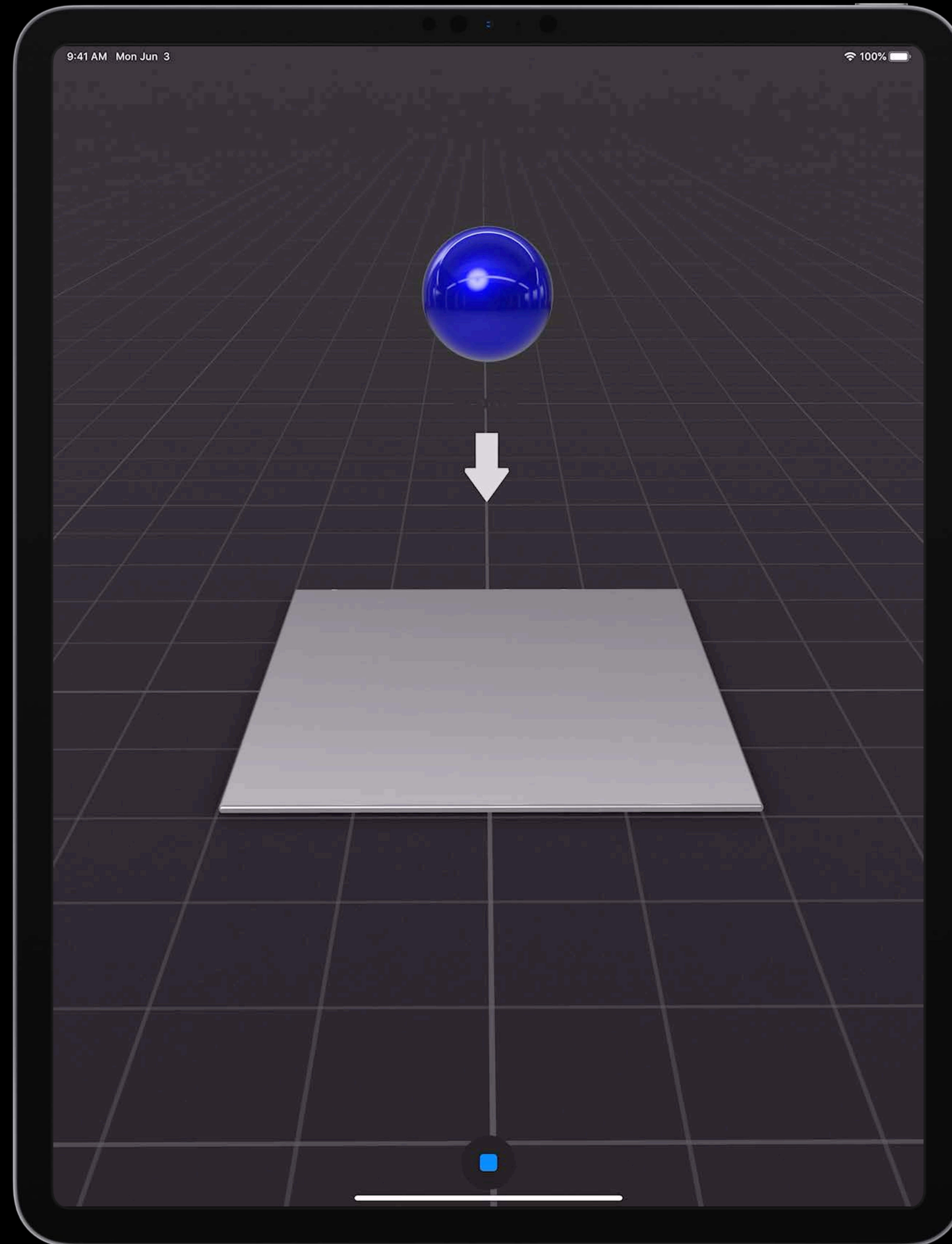
Wood



Rubber

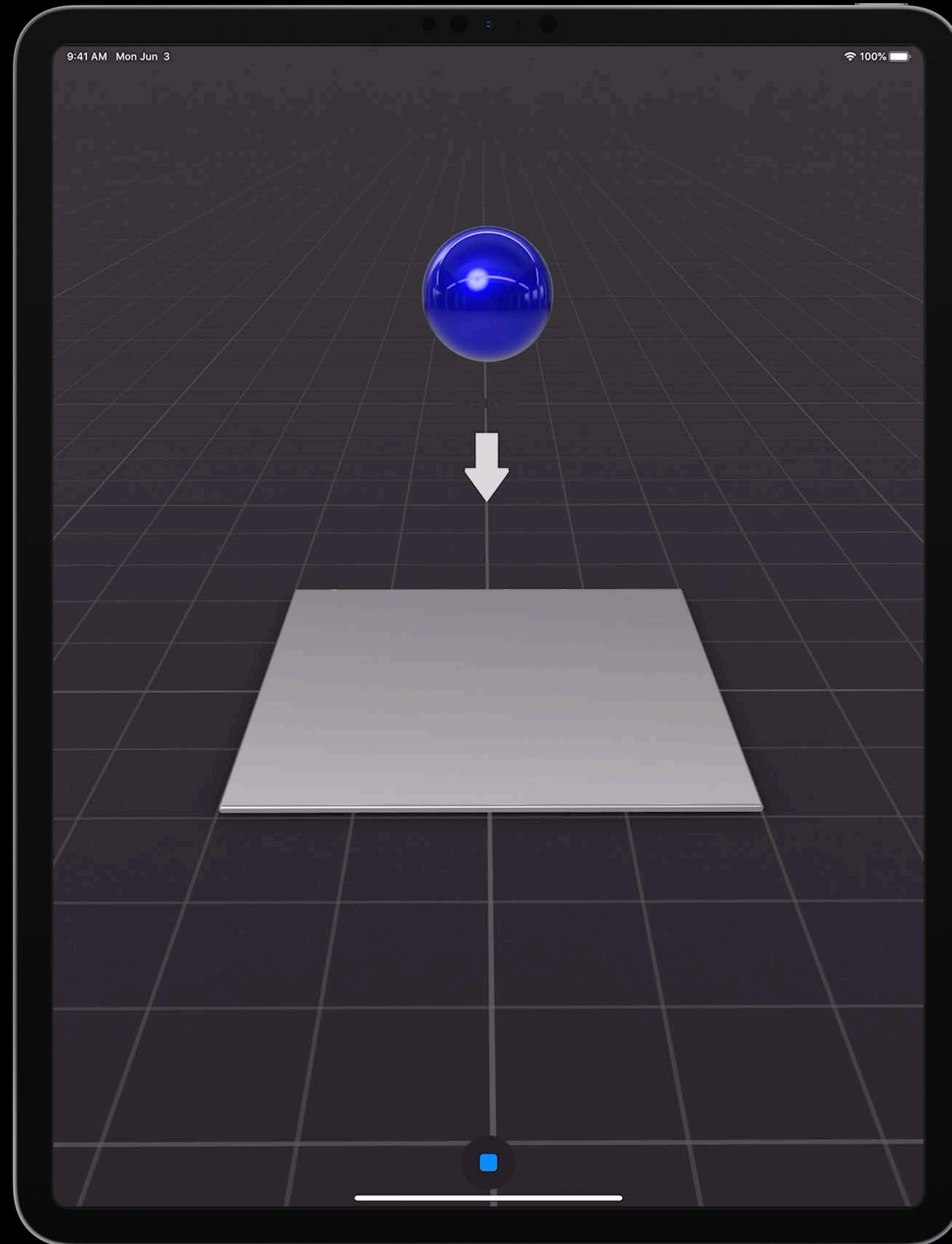
Forces

Gravity



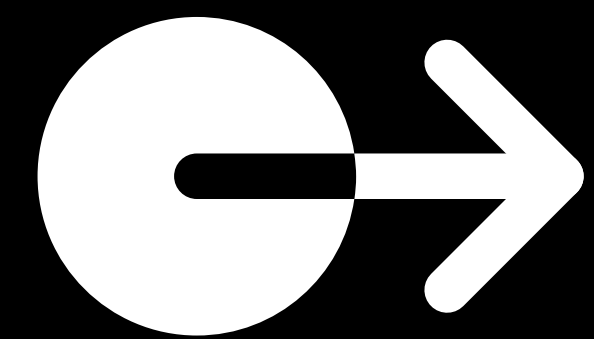
Forces

Gravity

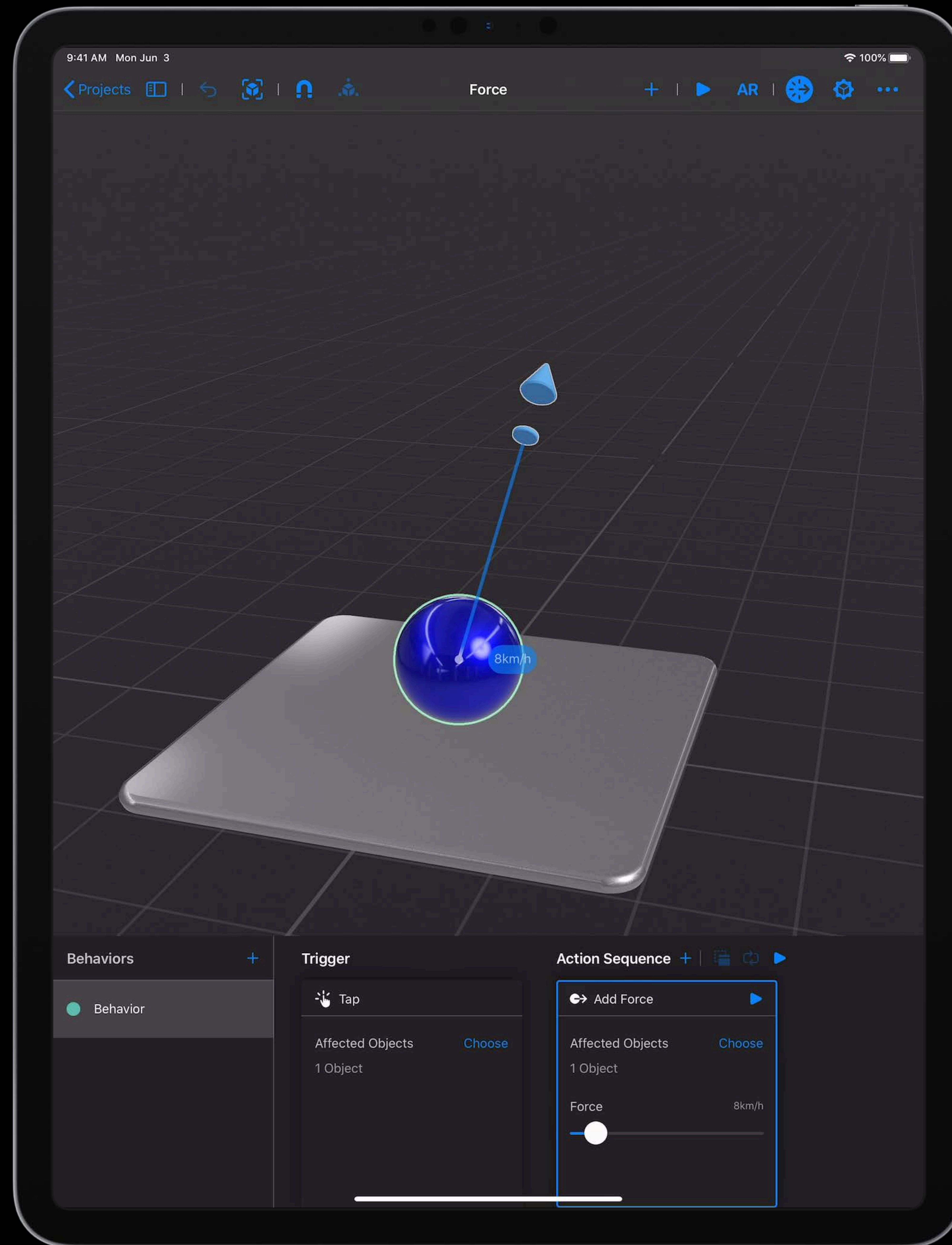


Forces

Force action

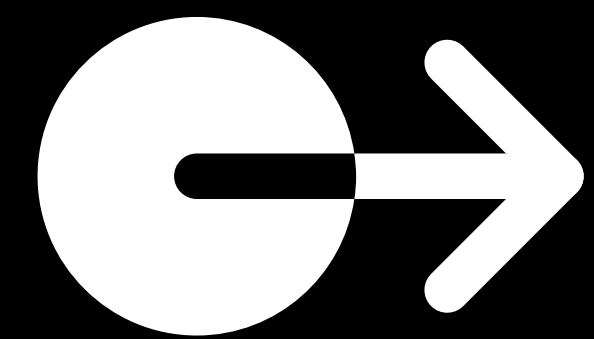


Add Force

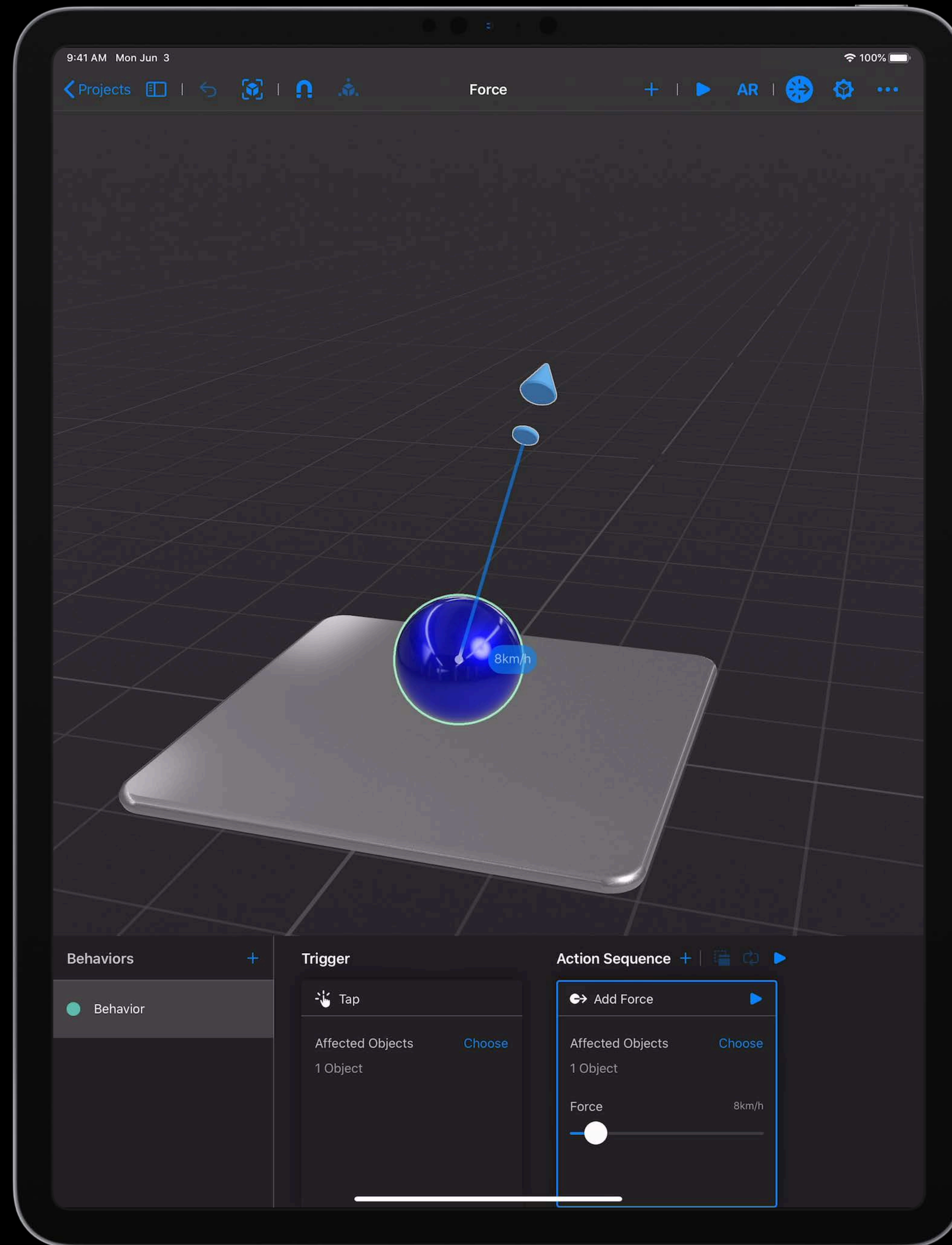


Forces

Force action

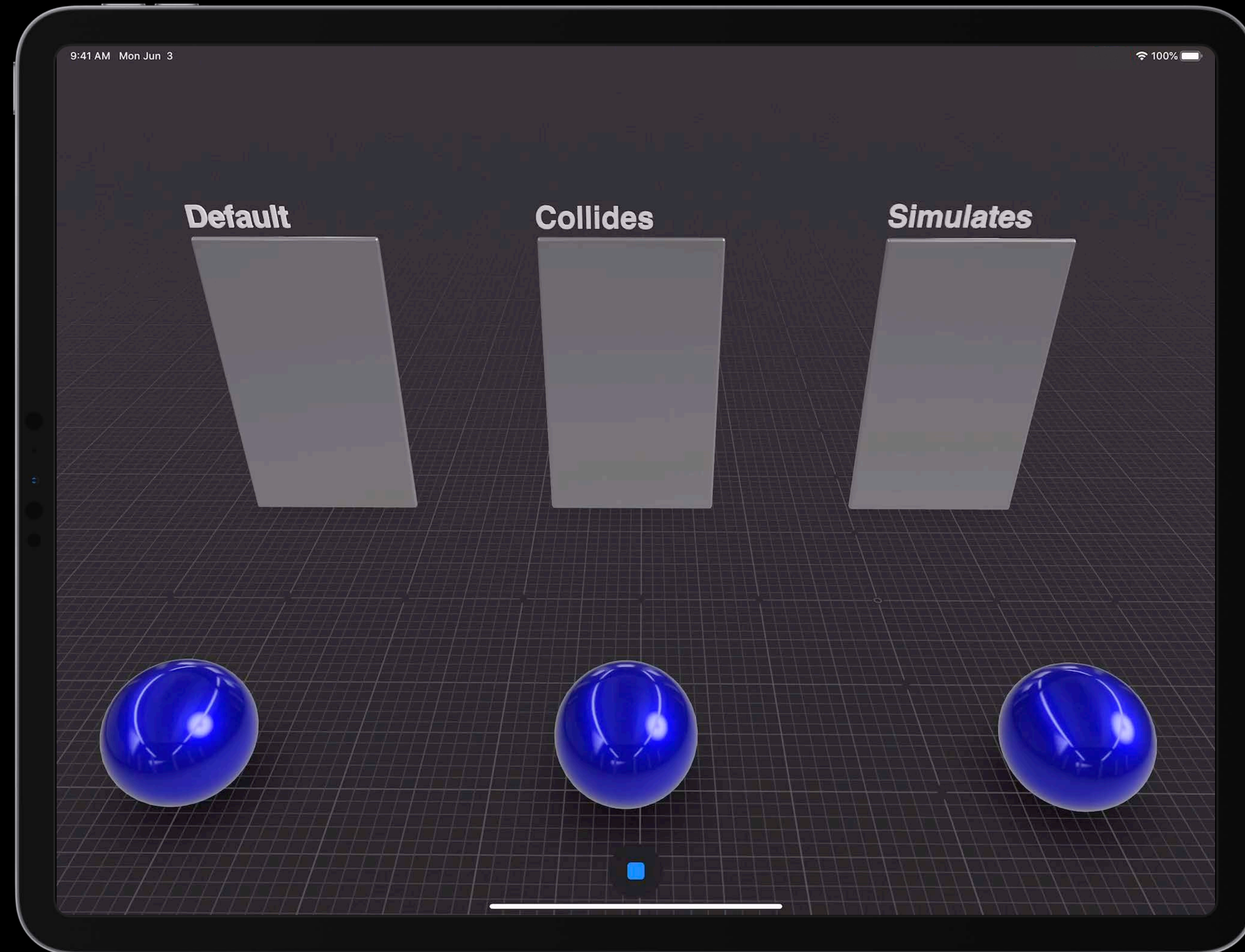


Add Force



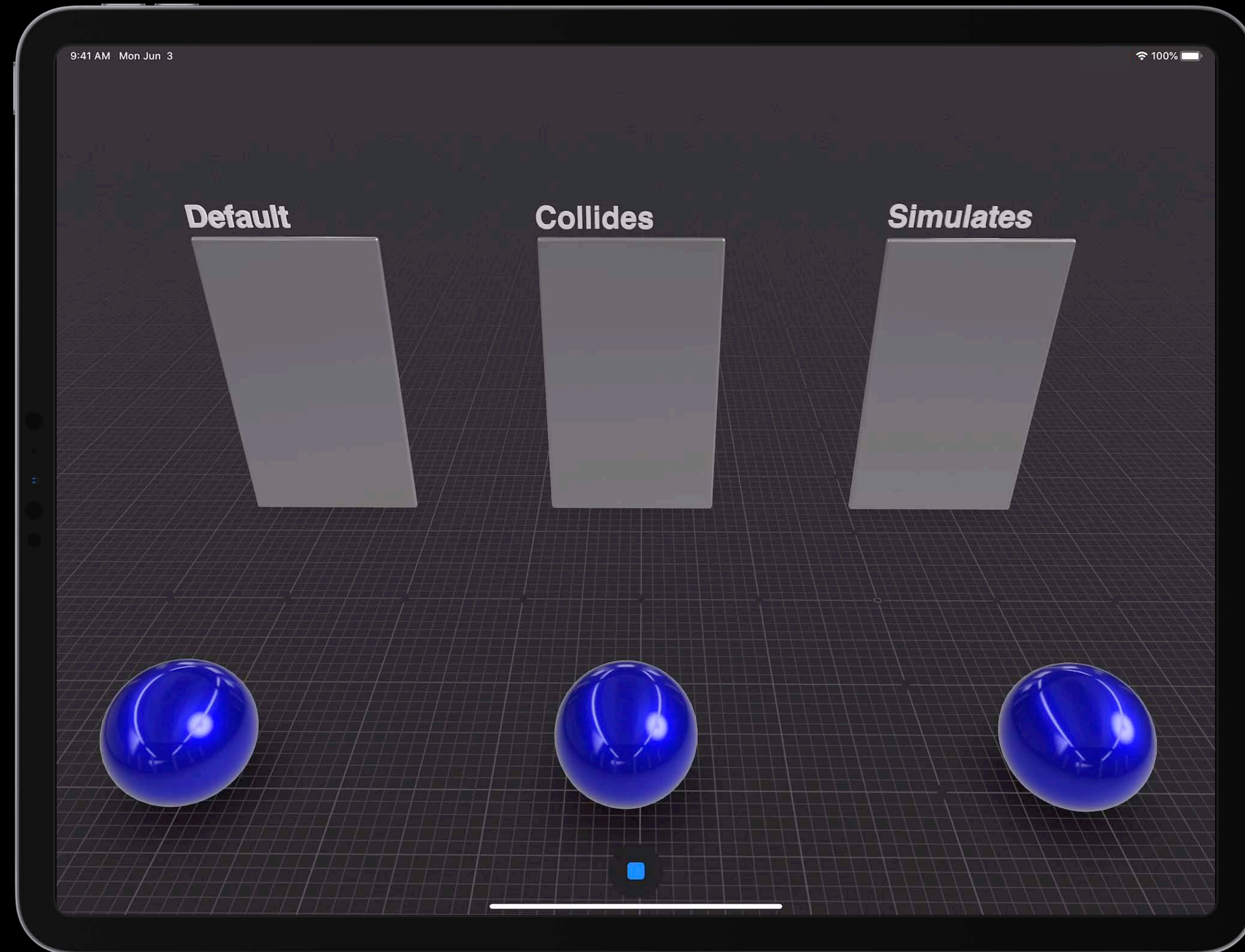
Collisions

Collides vs. simulates



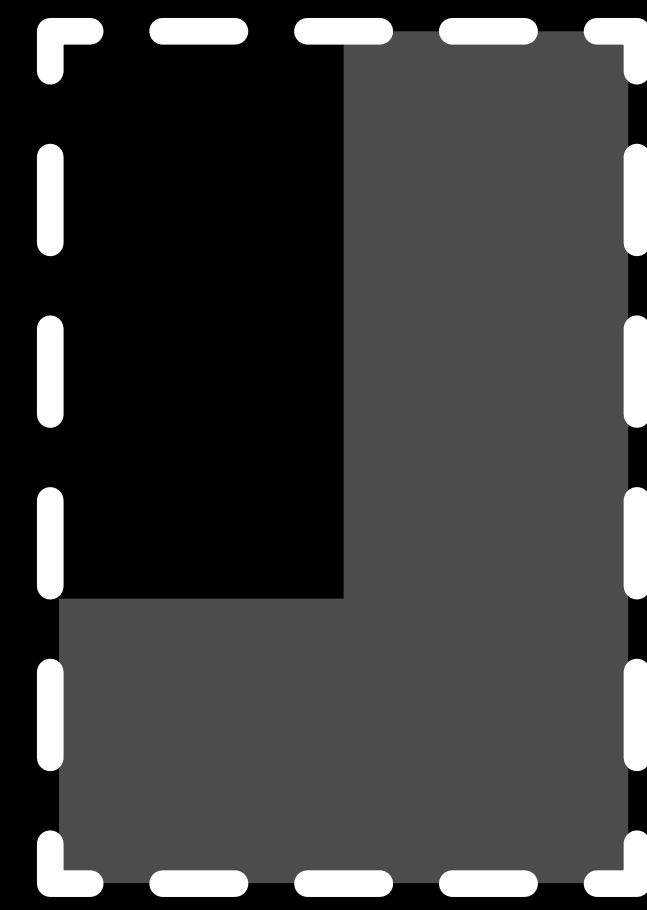
Collisions

Collides vs. simulates

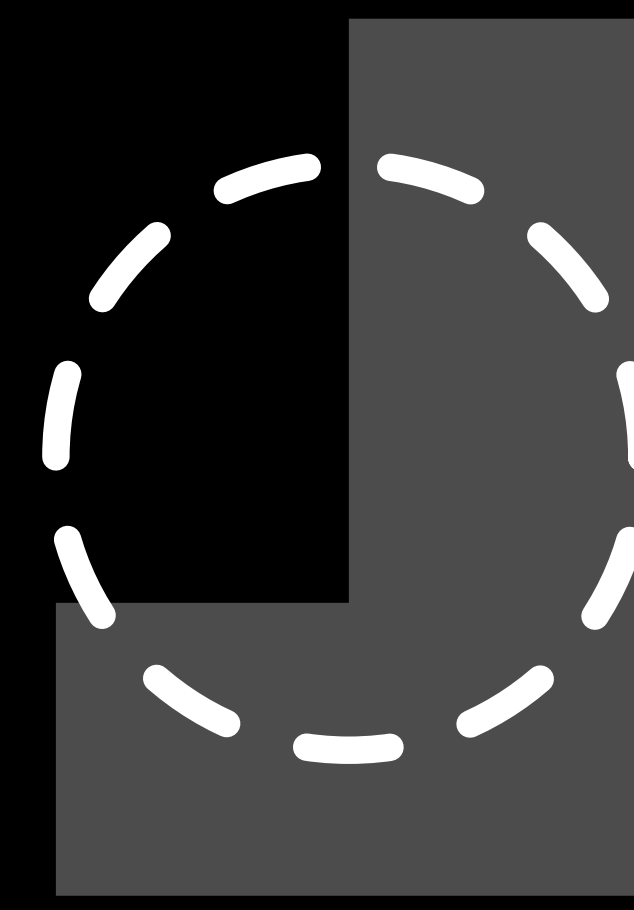


Collisions

Shapes



Box



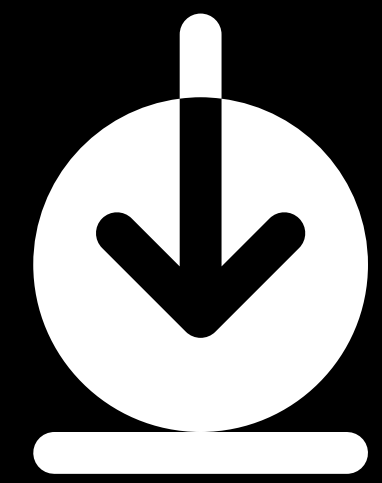
Sphere



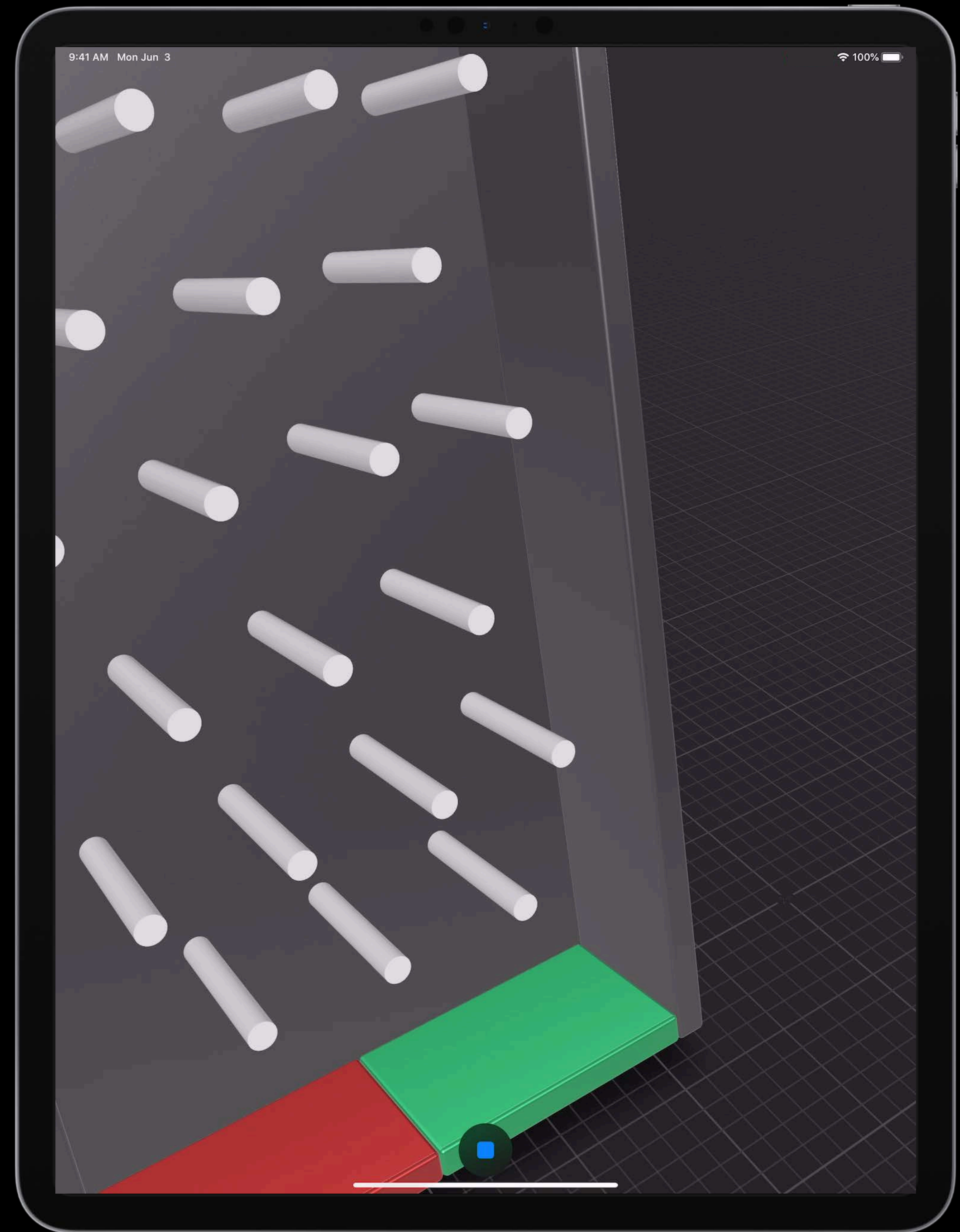
Capsule

Collisions

Trigger

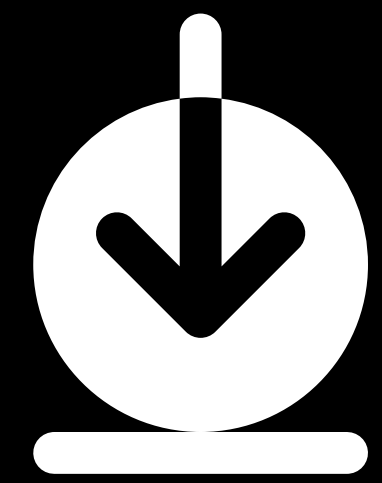


Collide

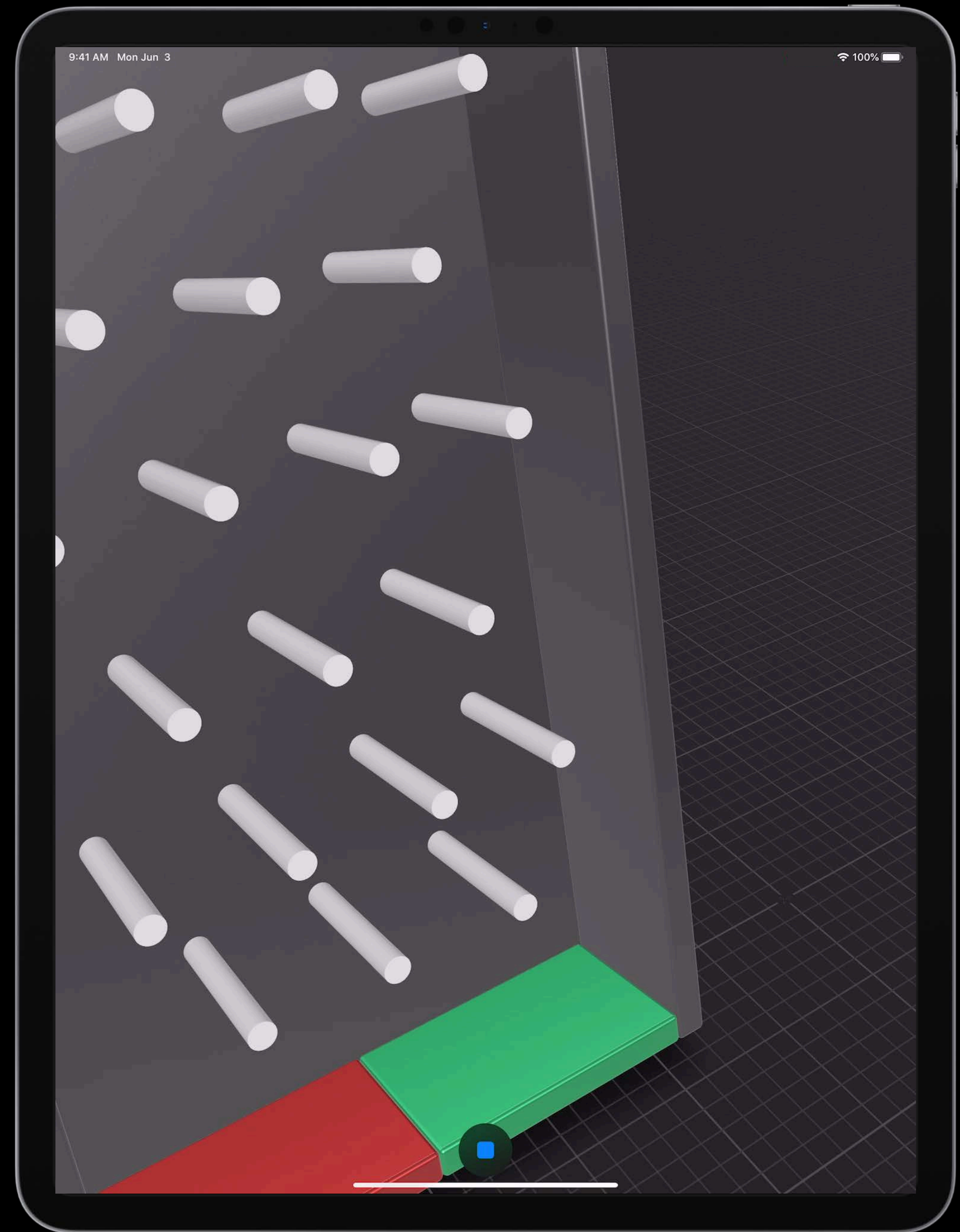


Collisions

Trigger



Collide



Building Scenes

Adding Behaviors

Using Physics

Building Apps

Building Scenes

Adding Behaviors

Using Physics

Building Apps

Building Apps

Abhi Ashutosh, Software Engineer

Getting Started

RealityKit AR or Game template in Xcode

New project from Reality Composer

Reality File exported from Reality Composer



Getting Started

Reality Composer Project



Reality File

Reality Composer Project

Project file for Reality Composer

Included in RealityKit AR and Game templates

Preview available in Xcode

Xcode automatically exports as a Reality File

SolarSystemLesson | Build SolarSystemLesson: **Succeeded**

SolarSystemLesson > SolarSystemLesson > SolarSystemLesson.rcproject

Open in Reality Composer

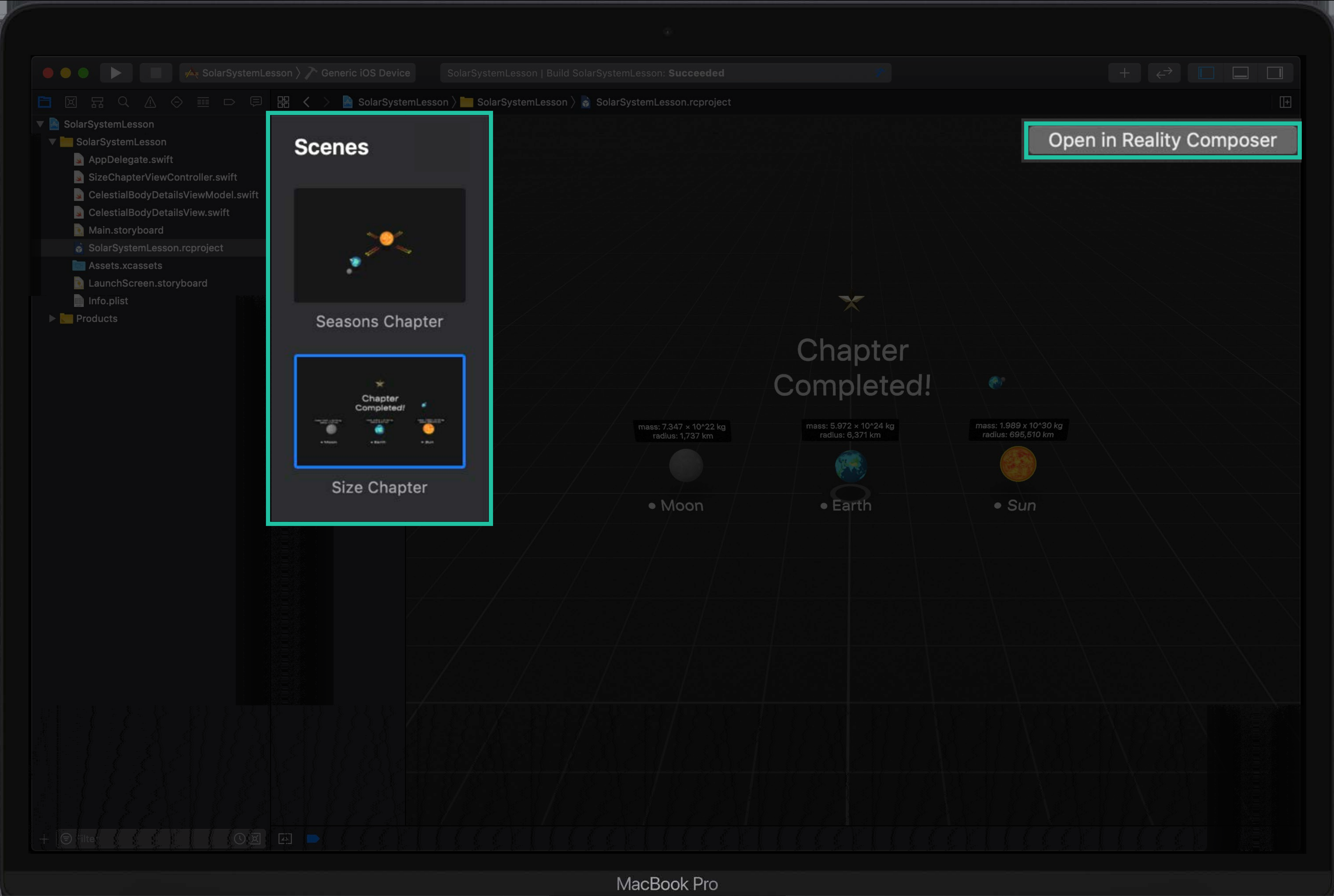
Scenes

- Seasons Chapter
- Size Chapter

Chapter Completed!

Object	mass	radius
Moon	7.347×10^{22} kg	1,737 km
Earth	5.972×10^{24} kg	6,371 km
Sun	1.989×10^{30} kg	695,510 km

MacBook Pro

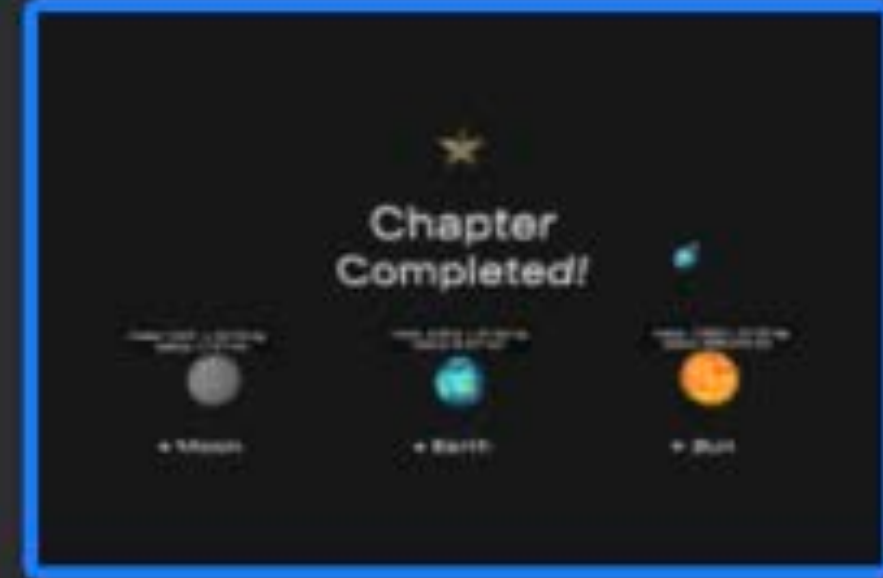


Open in Reality Composer

Scenes



Seasons Chapter



Size Chapter

Chapter Completed!

mass: 7.347×10^{22} kg
radius: 1,737 km

• Moon

mass: 5.972×10^{24} kg
radius: 6,371 km

• Earth

mass: 1.989×10^{30} kg
radius: 695,510 km

• Sun

Reality Composer Project

Automatic export in Xcode



Build in Xcode

SolarSystem.rcproject



SolarSystem.reality

Reality Composer Project

Automatic export in Xcode



Build in Xcode

SolarSystem.rcproject



MyExperience.app

Reality File

Contains all data required for rendering and simulation, optimized for RealityKit

Exported from Reality Composer

Automatically exported by Xcode

Reference directly in an application

Preview in AR Quick Look



Reality File

Contains all data required for rendering and simulation, optimized for RealityKit

Exported from Reality Composer

Automatically exported by Xcode

Reference directly in an application

Preview in AR Quick Look



Reality File

Structure



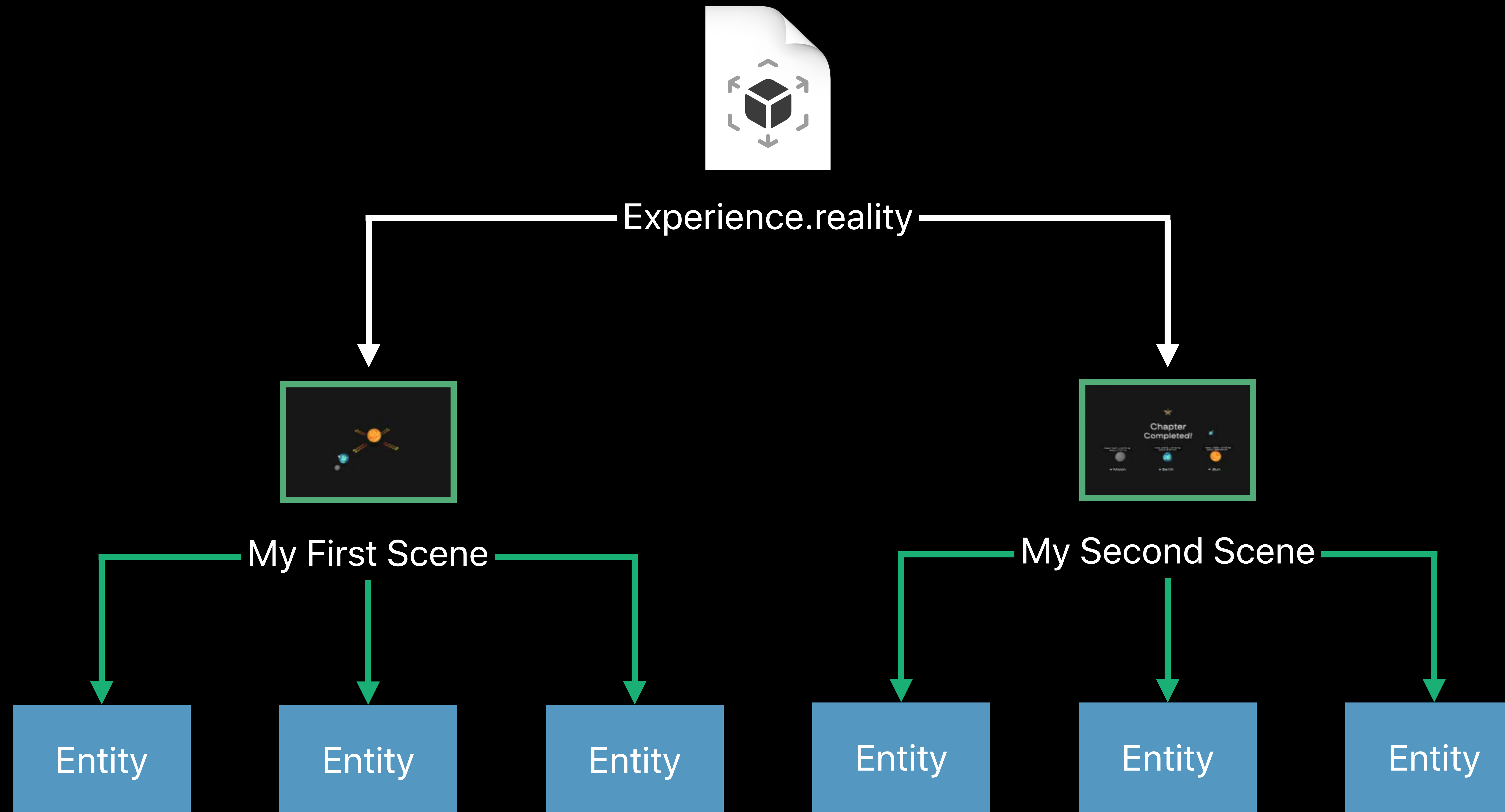
Experience.reality

Reality File Structure



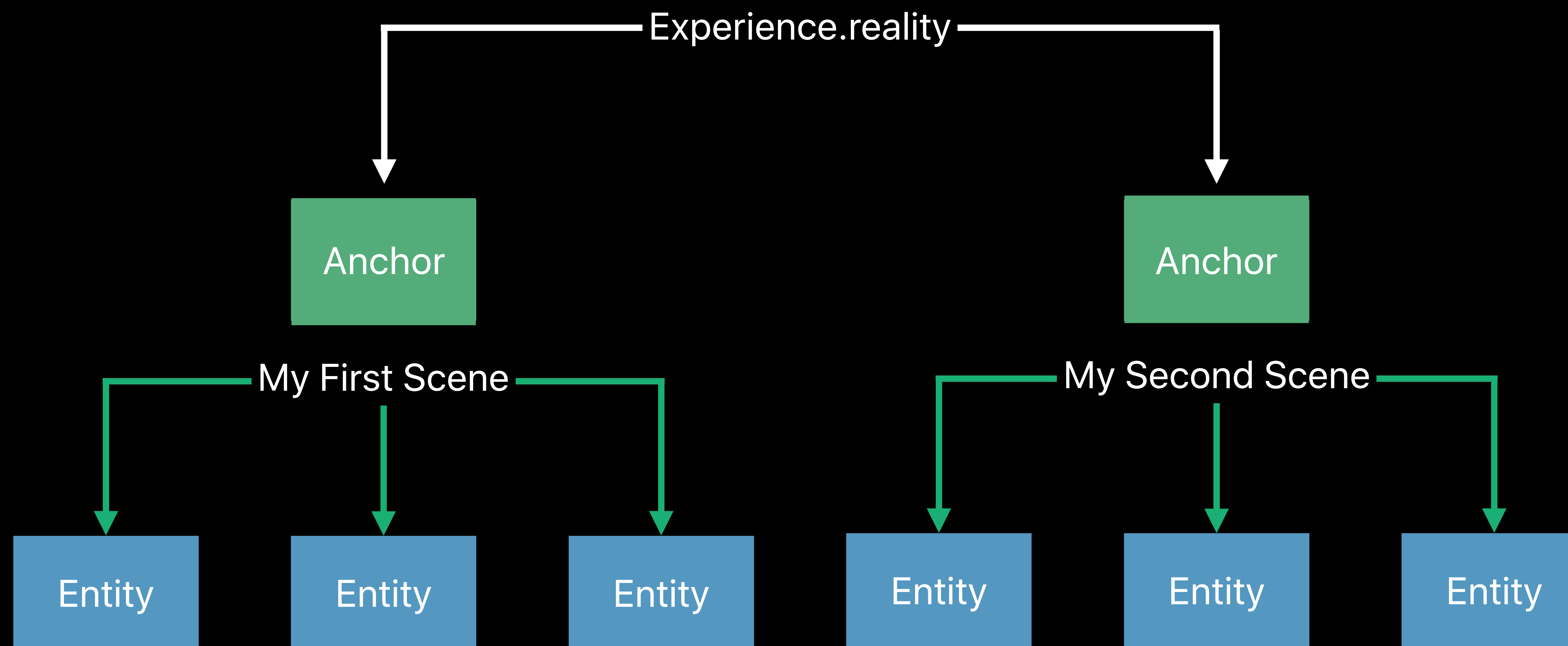
Reality File

Structure



Reality File

Structure



Reality Composer Project in Xcode

SolarSystem.rcproject

→
Build in Xcode

SolarSystem.reality



Code Generation in Xcode

Build in Xcode



SolarSystem.rcproject

SolarSystem.reality

SolarSystem.swift

Code Generation in Xcode

Automatically generated by Xcode

Application-specific, generated API for

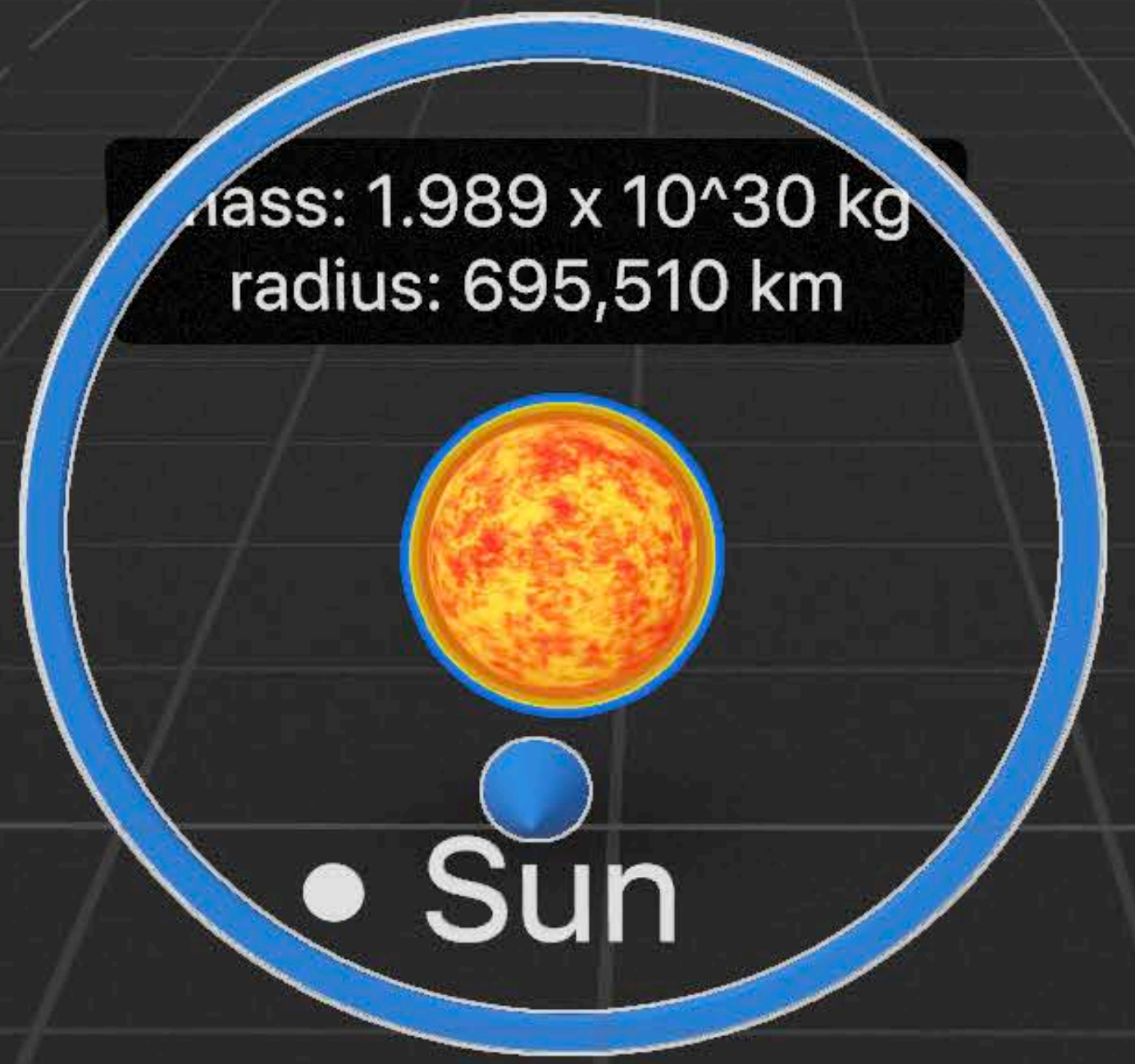
- Scenes
- Named entities
- Notify Actions
- Notification Triggers

Chapter Completed!

mass: 5.972×10^{24} kg
radius: 6,371 km



mass: 1.989×10^{30} kg
radius: 695,510 km



▼ Configure

Name Sun

▼ Physics

Collides Simulates

Material

Concrete Ice Lead Plast

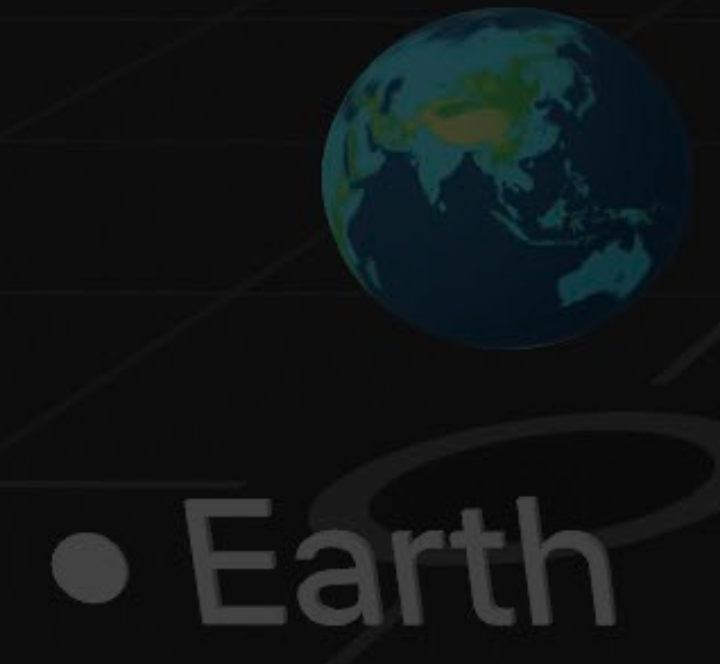
Collision Shape

Box Capsule Sphere

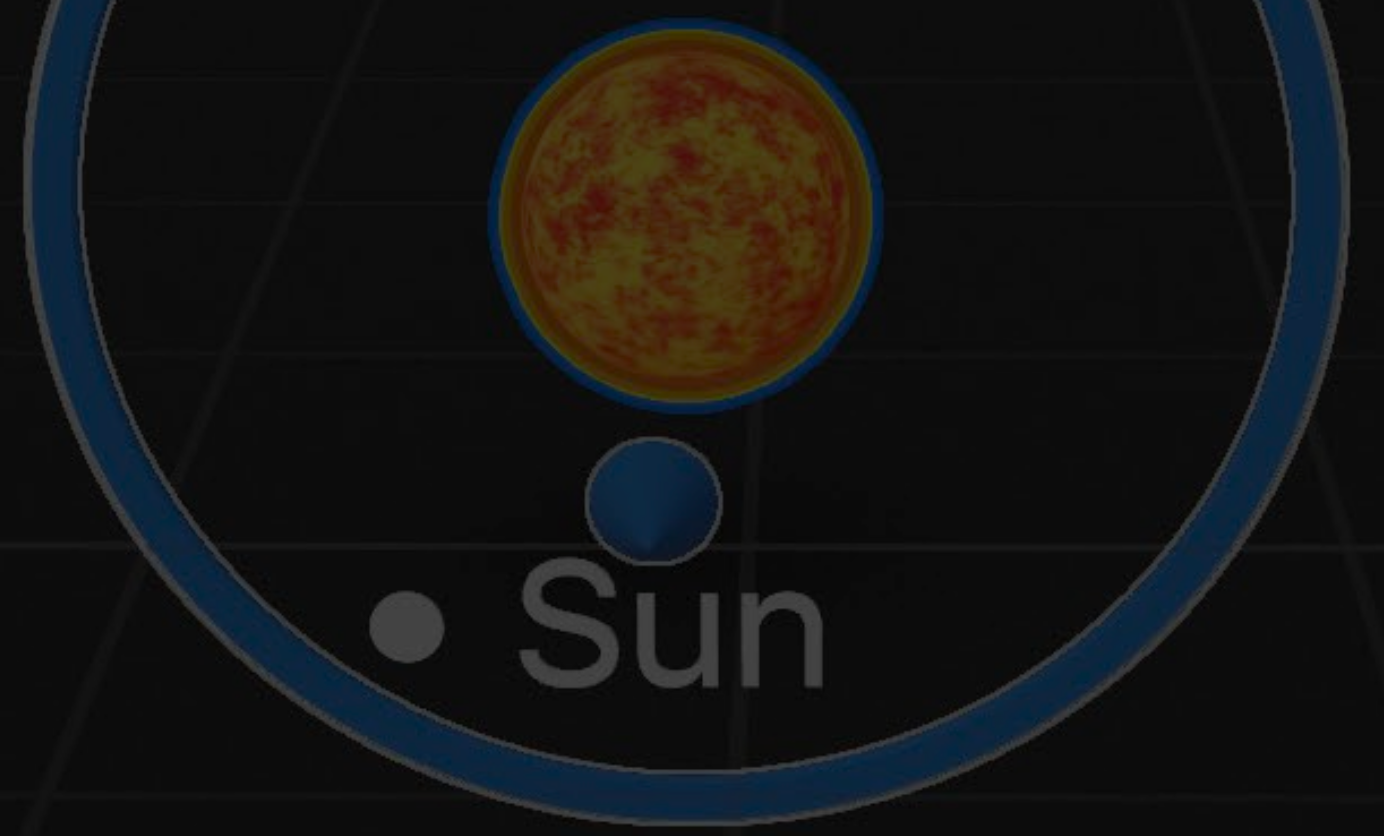


Chapter Completed!

mass: 5.972×10^{24} kg
radius: 6,371 km



mass: 1.989×10^{30} kg
radius: 695,510 km



▼ **Configure**

Name

Material

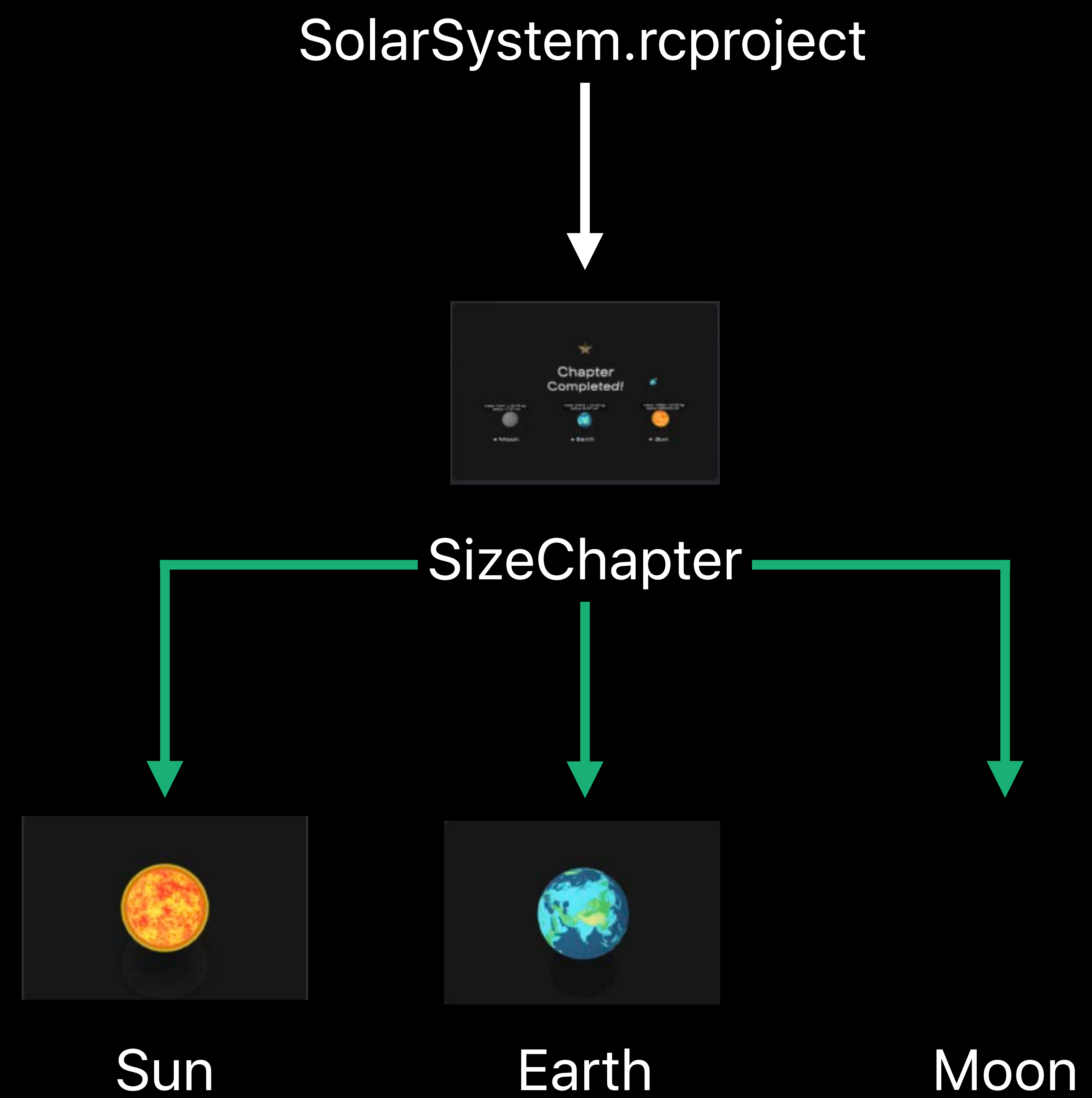
Concrete Ice Lead Plast

Collision Shape

Box Capsule Sphere

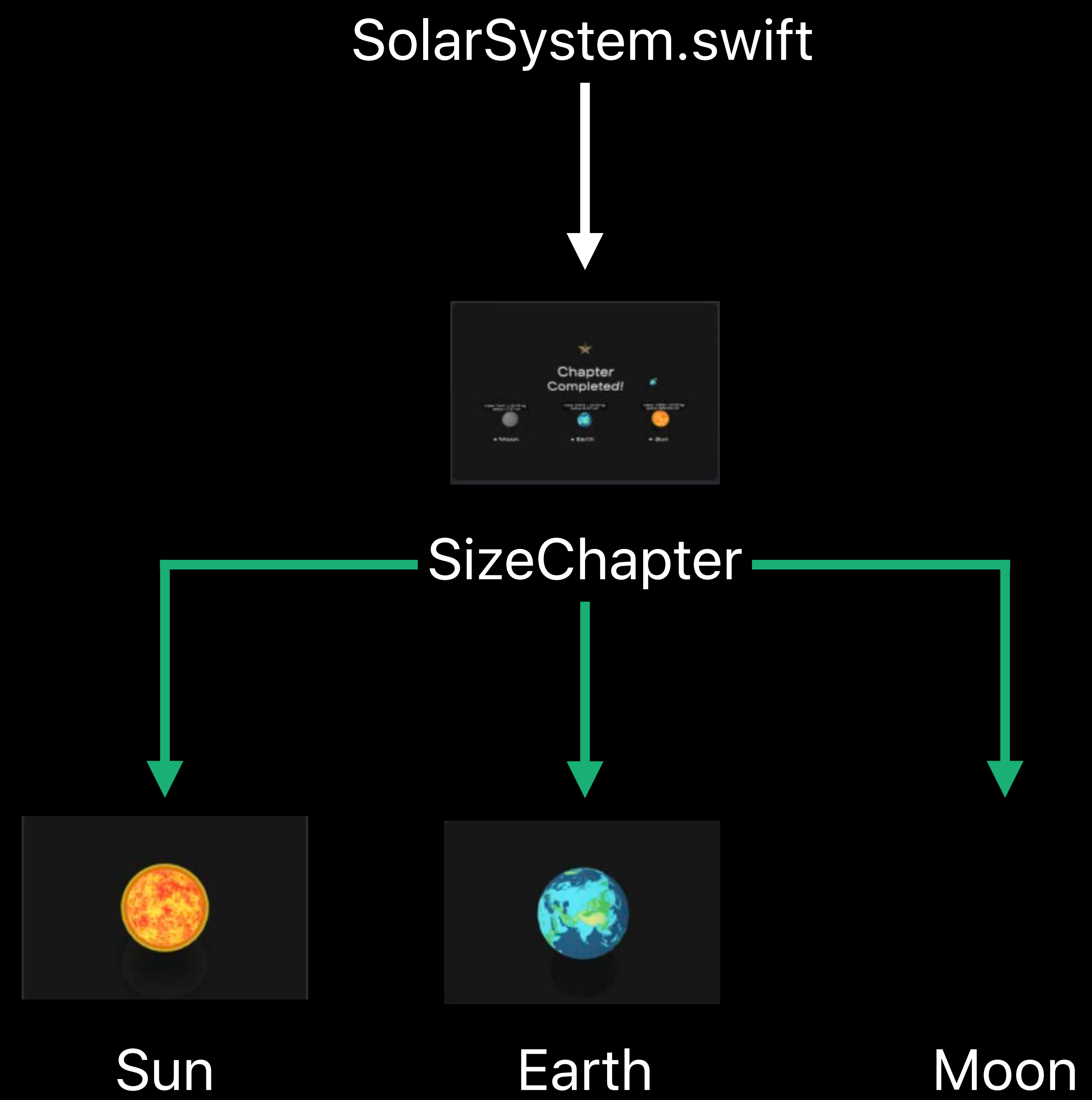
Code Generation in Xcode

Generated code structure



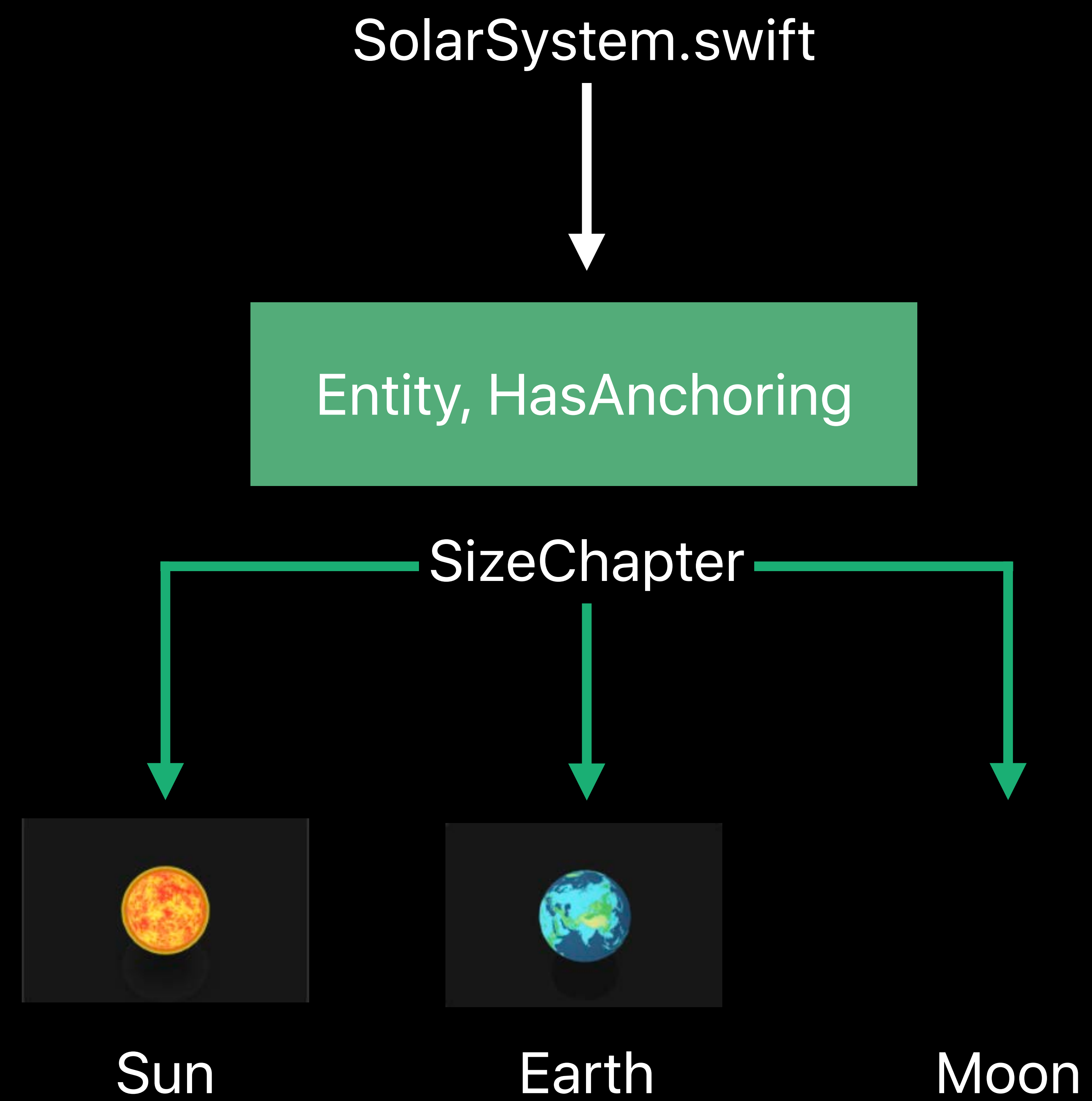
Code Generation in Xcode

Generated code structure



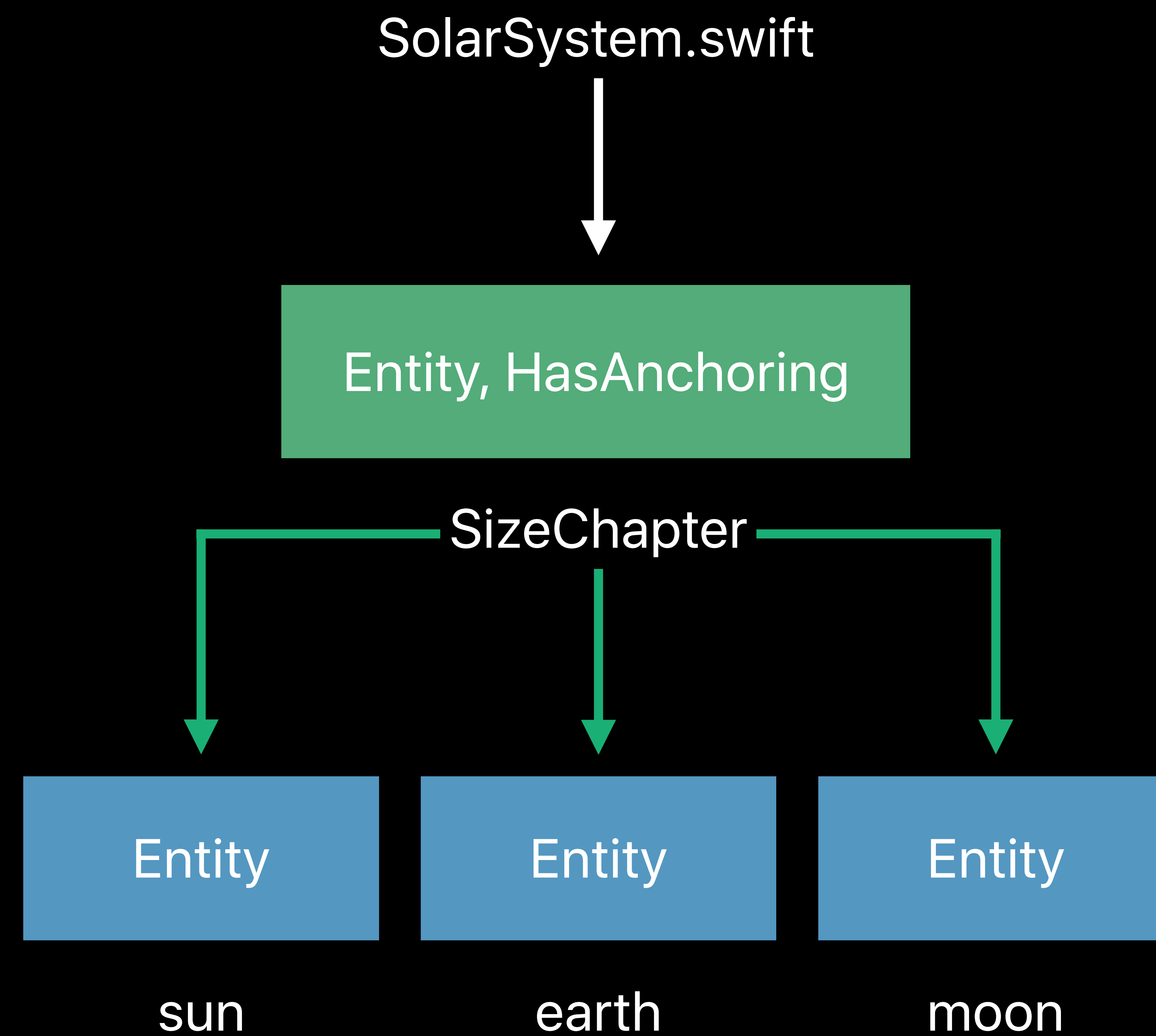
Code Generation in Xcode

Generated code structure



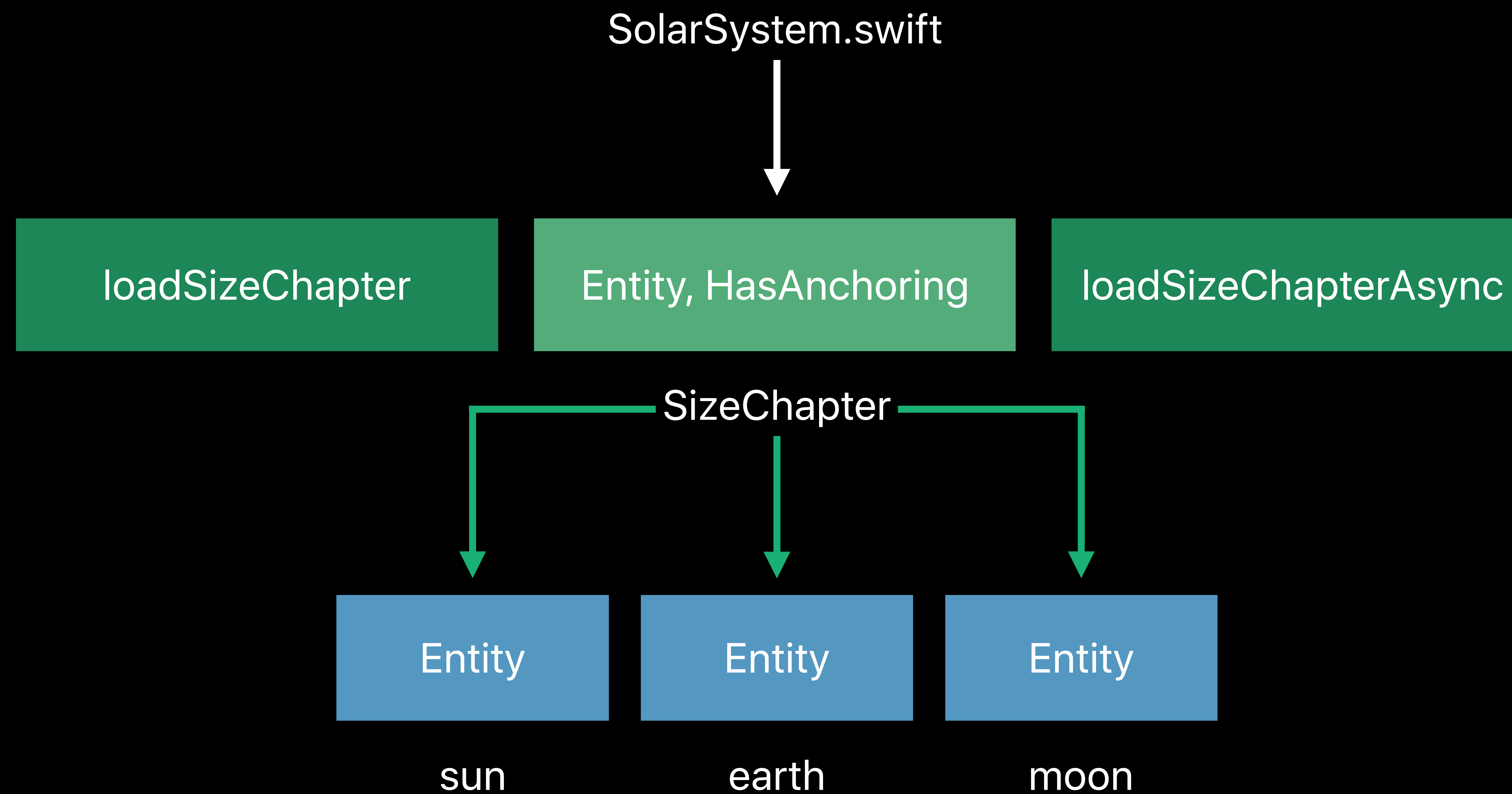
Code Generation in Xcode

Generated code structure



Code Generation in Xcode

Generated code structure



Code Generation in Xcode

Loading a Reality File synchronously

```
let seasonsChapter = try? SolarSystem.loadSeasonsChapter()
```

```
// Use the loaded anchor here
```

Code Generation in Xcode

Loading a Reality File synchronously

```
let seasonsChapter = try? SolarSystem.loadSeasonsChapter()
```

```
// Use the loaded anchor here
```


Code Generation in Xcode

Loading a Reality File asynchronously

```
SolarSystem.loadSeasonsChapterAsync { result in
    switch result {
    case .success(let anchor):
        // Use loaded anchor here
    case .failure(let error):
        // Handle failure
    }
}
```

Code Generation in Xcode

Loading a Reality File asynchronously

```
SolarSystem.loadSeasonsChapterAsync { result in
    switch result {
    case .success(let anchor):
        // Use loaded anchor here
    case .failure(let error):
        // Handle failure
    }
}
```

Code Generation in Xcode

Loading a Reality File asynchronously

```
SolarSystem.loadSeasonsChapterAsync { result in
    switch result {
    case .success(let anchor):
        // Use loaded anchor here
    case .failure(let error):
        // Handle failure
    }
}
```

Code Generation in Xcode

Loading a Reality File asynchronously

```
SolarSystem.loadSeasonsChapterAsync { result in
    switch result {
    case .success(let anchor):
        // Use loaded anchor here
    case .failure(let error):
        // Handle failure
    }
}
```

Code Generation in Xcode

Accessing entities

```
// Load Reality File anchor, `seasonsChapter`, above...
```

```
let sun = seasonsChapter.sun
```

```
let earth = seasonsChapter.earth
```

```
let moon = seasonsChapter.moon
```

Code Generation in Xcode

Accessing entities

```
// Load Reality File anchor, `seasonsChapter`, above...
```

```
let sun = seasonsChapter.sun
```

```
let earth = seasonsChapter.earth
```

```
let moon = seasonsChapter.moon
```

Code Generation in Xcode

Getting started

```
// Load the "Seasons Chapter" scene from the "SolarSystem" Reality File
let seasonsChapter = try! SolarSystem.loadSeasonsChapter()

// Add the seasons chapter anchor into the scene
arView.scene.anchors.append(seasonsChapter)
```

Code Generation in Xcode

Getting started

```
// Load the "Seasons Chapter" scene from the "SolarSystem" Reality File
let seasonsChapter = try! SolarSystem.loadSeasonsChapter()

// Add the seasons chapter anchor into the scene
arView.scene.anchors.append(seasonsChapter)
```

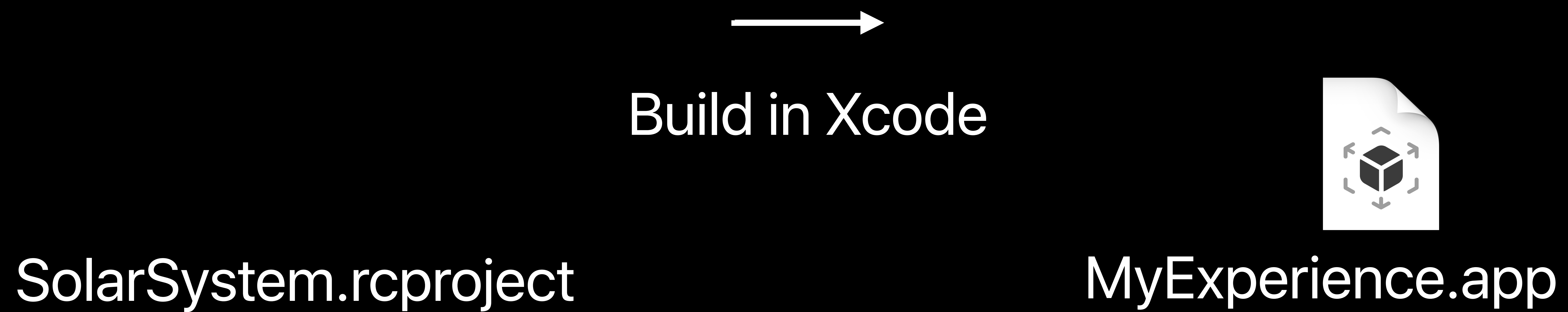

Code Generation in Xcode

Getting started

```
// Load the "Seasons Chapter" scene from the "SolarSystem" Reality File  
let seasonsChapter = try! SolarSystem.loadSeasonsChapter()
```

```
// Add the seasons chapter anchor into the scene  
arView.scene.anchors.append(seasonsChapter)
```

Code Generation in Xcode



Reality File

Loading synchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {
    return
}

let anchor = try? Entity.loadAnchor(contentsOf: url, withName: "SeasonsChapter")

// Use the loaded anchor here...
```

Reality File

Loading synchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {  
    return  
}
```

```
let anchor = try? Entity.loadAnchor(contentsOf: url, withName: "SeasonsChapter")
```

```
// Use the loaded anchor here...
```

Reality File

Loading synchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {  
    return  
}
```

```
let anchor = try? Entity.loadAnchor(contentsOf: url, withName: "SeasonsChapter")
```

```
// Use the loaded anchor here...
```

Reality File

Loading asynchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {
    return
}

let loadRequest = Entity.loadAnchorAsync(contentsOf: url, withName: "SeasonsChapter")
_ = loadRequest.sink(receiveCompletion: { completion in
    // Handle completion state
}, receiveValue: { anchor in
    // Use loaded anchor here
})
```

Reality File

Loading asynchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {  
    return  
}
```

```
let loadRequest = Entity.loadAnchorAsync(contentsOf: url, withName: "SeasonsChapter")  
_ = loadRequest.sink(receiveCompletion: { completion in  
    // Handle completion state  
}, receiveValue: { anchor in  
    // Use loaded anchor here  
})
```

Reality File

Loading asynchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {
    return
}
```

```
let loadRequest = Entity.loadAnchorAsync(contentsOf: url, withName: "SeasonsChapter")
_ = loadRequest.sink(receiveCompletion: { completion in
    // Handle completion state
}, receiveValue: { anchor in
    // Use loaded anchor here
})
```


Reality File

Loading asynchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {
    return
}

let loadRequest = Entity.loadAnchorAsync(contentsOf: url, withName: "SeasonsChapter")
_ = loadRequest.sink(receiveCompletion: { completion in
    // Handle completion state
}, receiveValue: { anchor in
    // Use loaded anchor here
})
```

Reality File

Loading asynchronously

```
guard let url = Bundle.main.url(forResource: "SolarSystem", withExtension: "reality") else {
    return
}

let loadRequest = Entity.loadAnchorAsync(contentsOf: url, withName: "SeasonsChapter")
_ = loadRequest.sink(receiveCompletion: { completion in
    // Handle completion state
}, receiveValue: { anchor in
    // Use loaded anchor here
})
```

Reality File

Accessing entities

```
// Load Reality File anchor above...  
  
let sun = anchor.findEntity(named: "Sun")  
let earth = anchor.findEntity(named: "Earth")  
let moon = anchor.findEntity(named: "Moon")  
  
// Use fetched entities below...
```

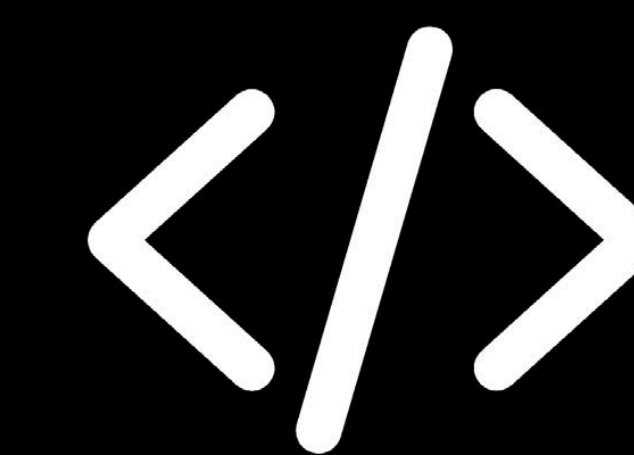
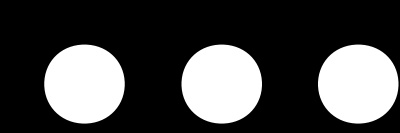
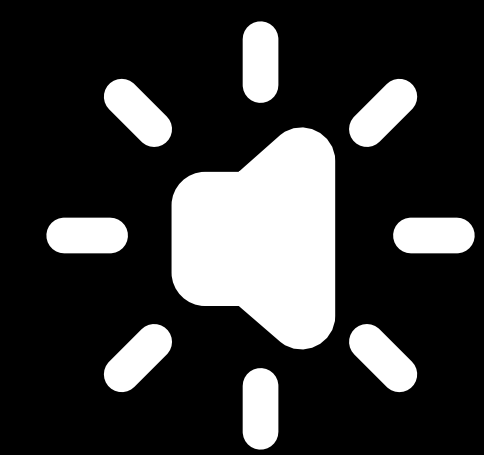
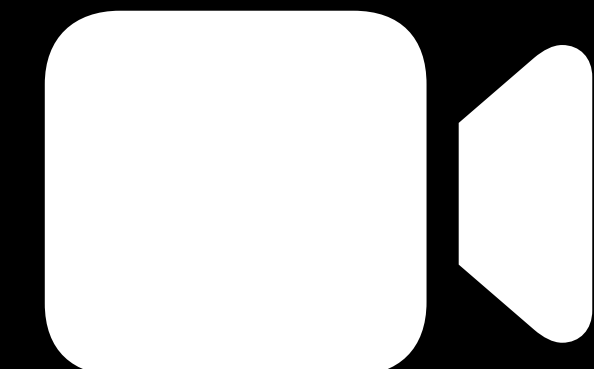
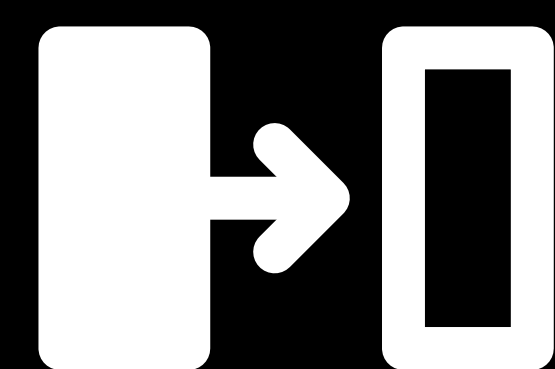
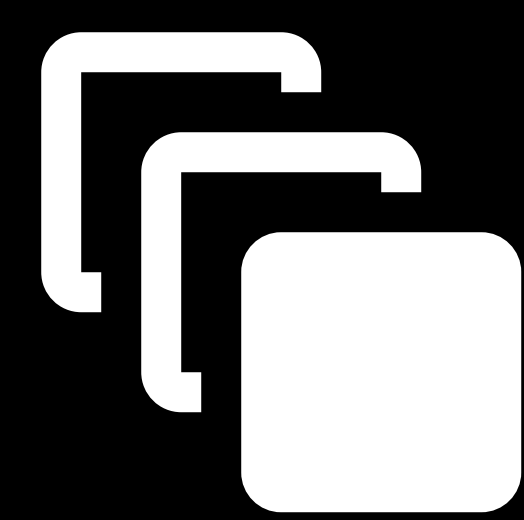
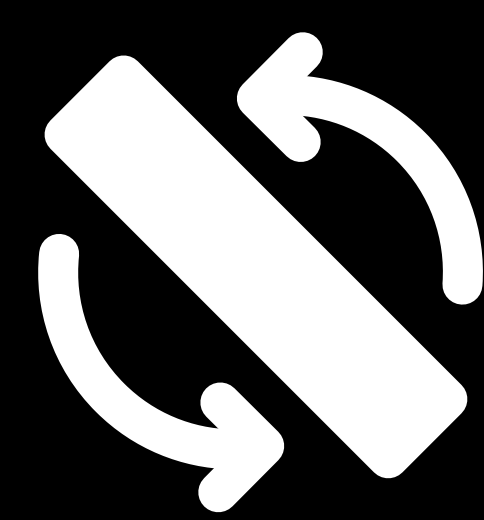
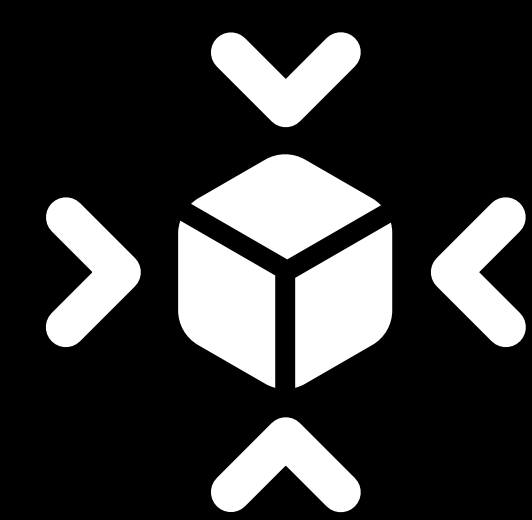
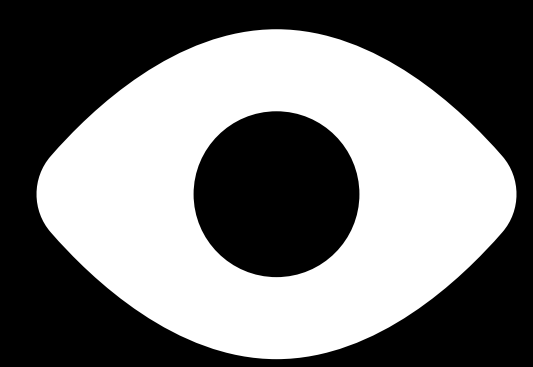
Reality File

Accessing entities

```
// Load Reality File anchor above...  
  
let sun = anchor.findEntity(named: "Sun")  
let earth = anchor.findEntity(named: "Earth")  
let moon = anchor.findEntity(named: "Moon")  
  
// Use fetched entities below...
```

UI Applications and AR Content

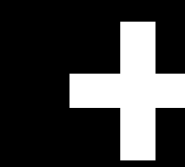
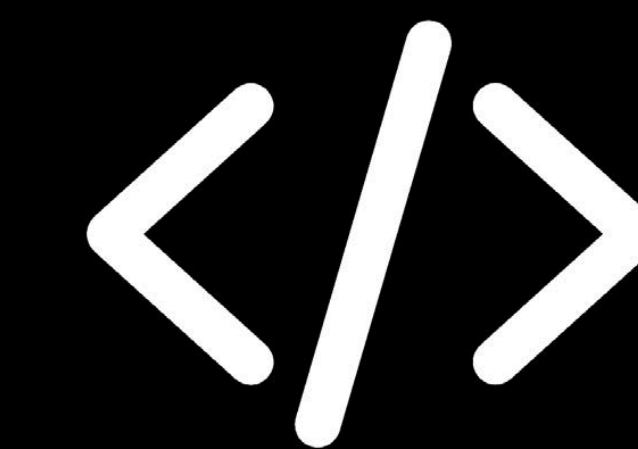
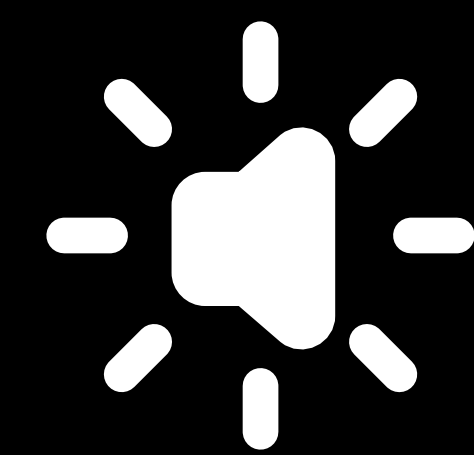
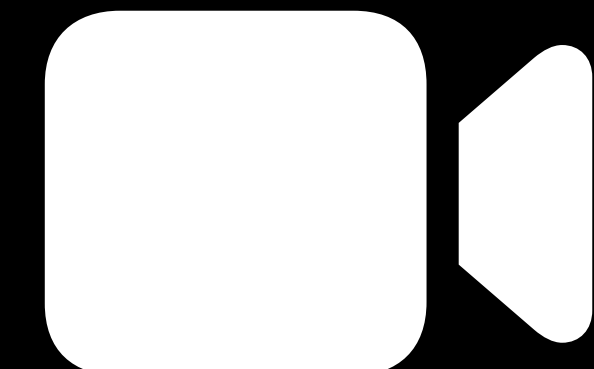
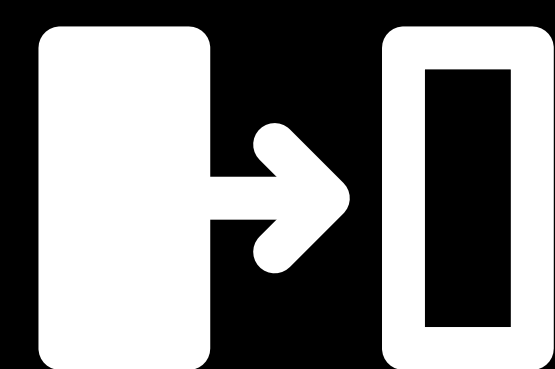
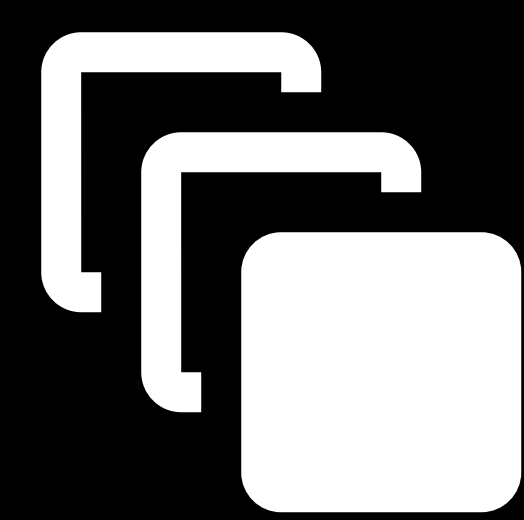
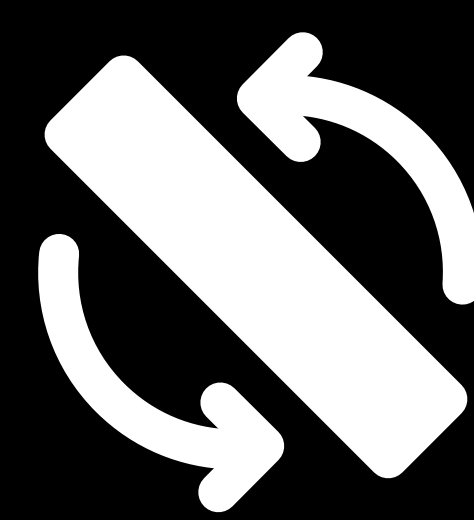
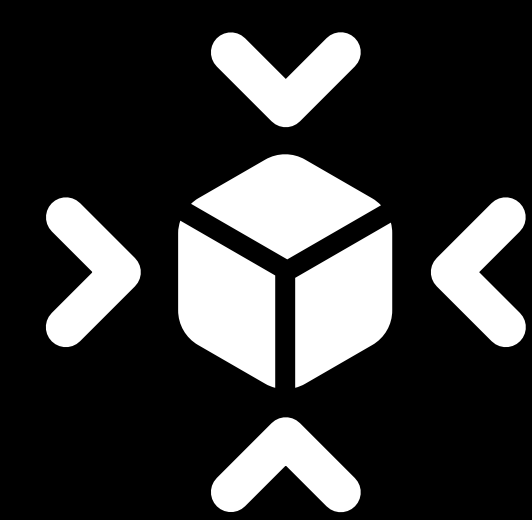
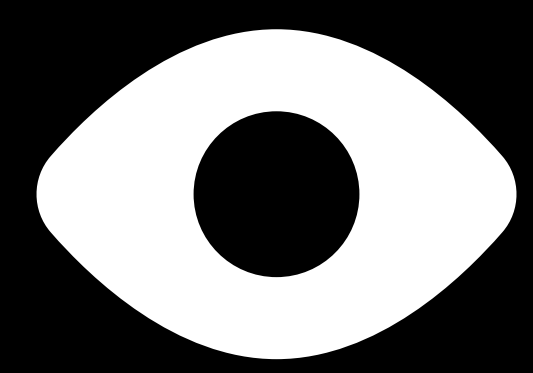
Actions



Application Logic

UI Applications and AR Content

Actions



Application Logic

Notify Action

Notify Action



Set up in Reality Composer


Invoked in action sequences

Settable closure in application code

Accessible by name in code


MacBook Pro

 **Notify** 



Affected Objects 

1 Object

Identifier



Action will notify your application code using the name above when it occurs (Xcode)

 Notify 

Affected Objects

Choose

1 Object

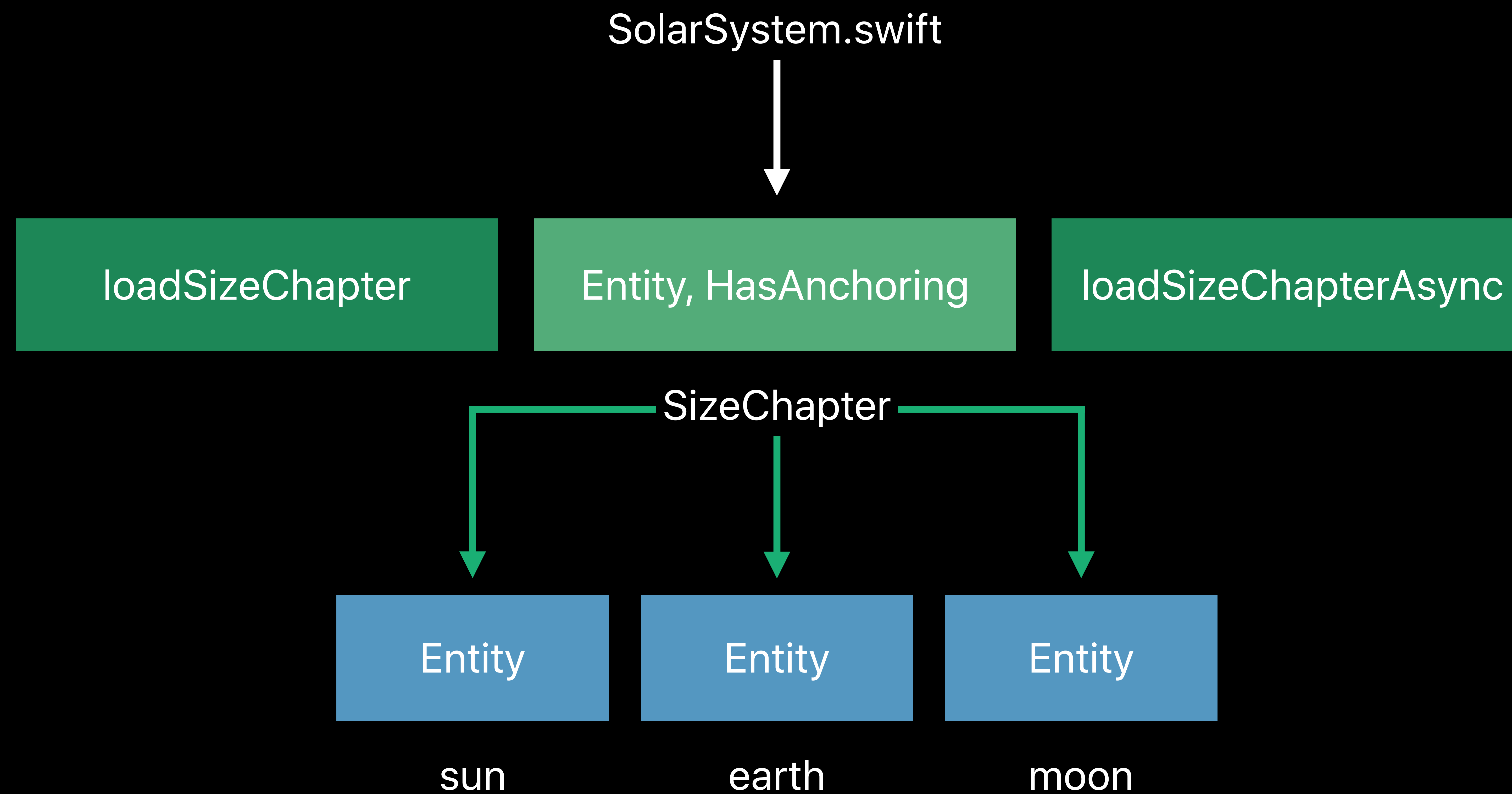
Identifier

Display Earth Details

Action will notify your application code using the name above when it occurs (Xcode)

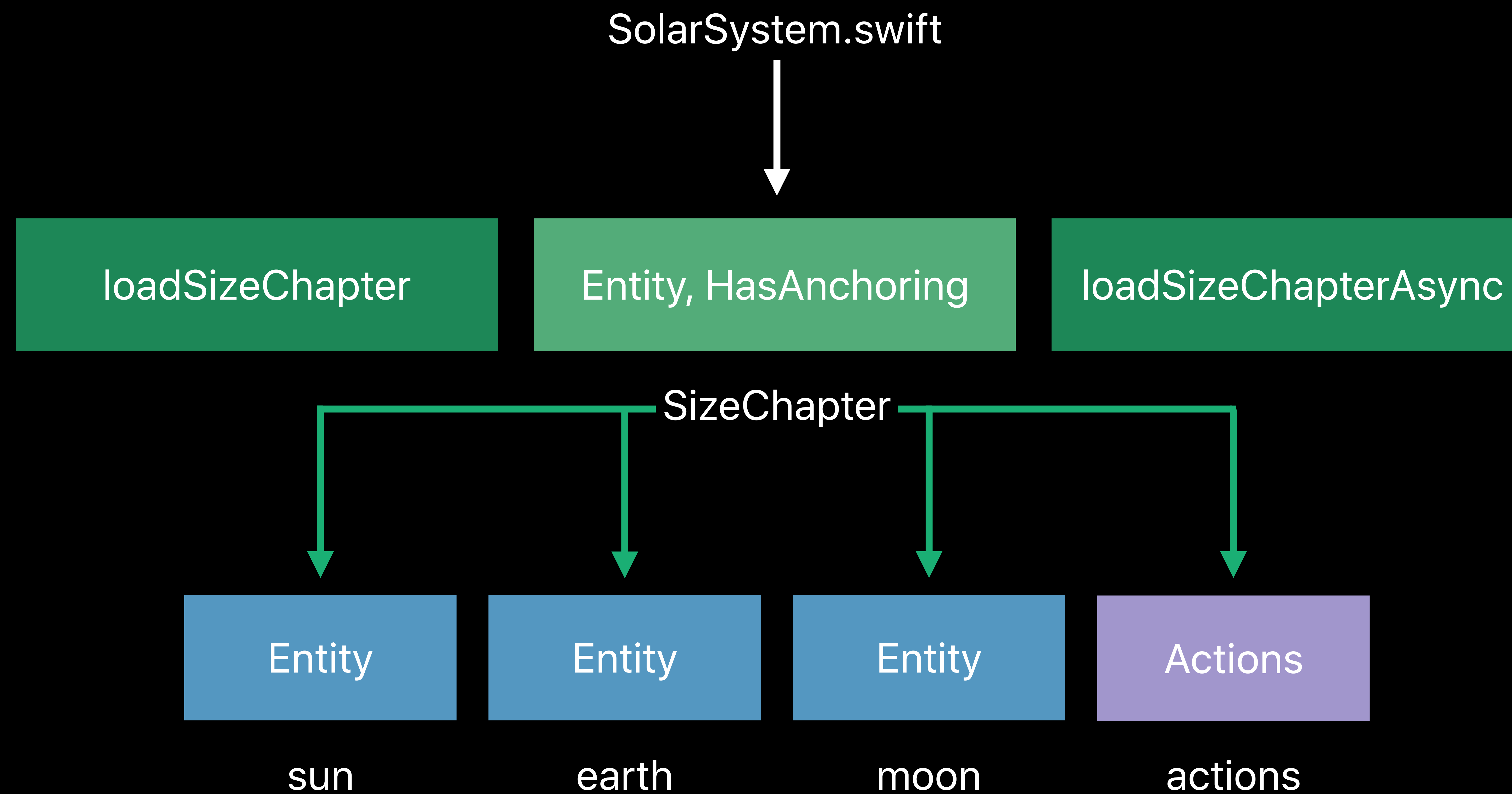
Notify Action

Generated code structure



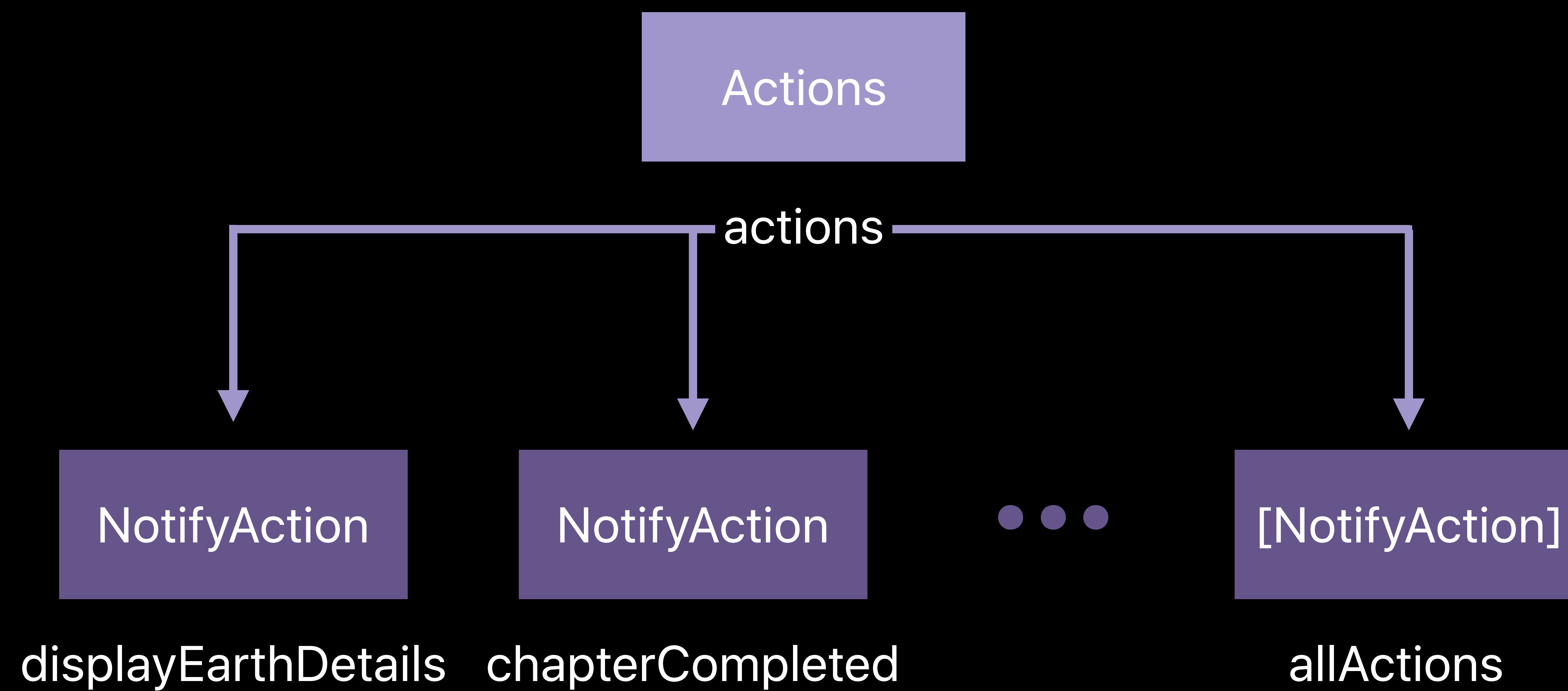
Notify Action

Generated code structure



Notify Action

Generated code structure



Notify Action

Defining closure

```
seasonsChapter.actions.displayEarthDetails.onAction = { entity in  
    // Display details about Fall  
}
```

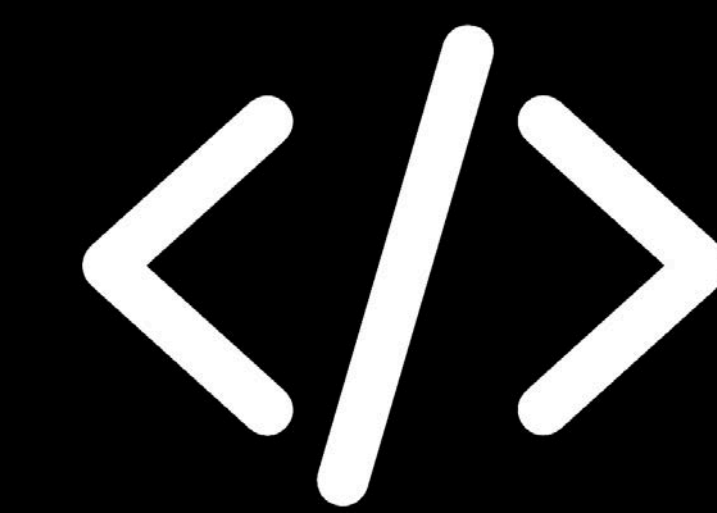
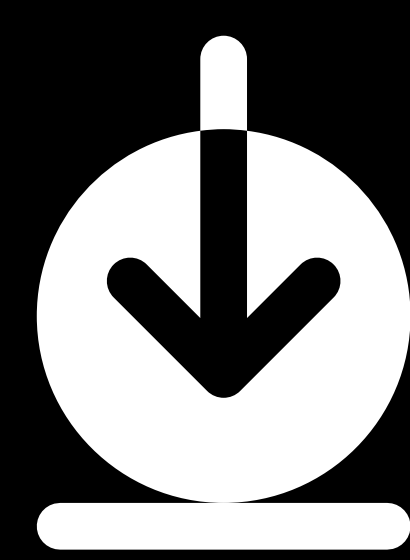
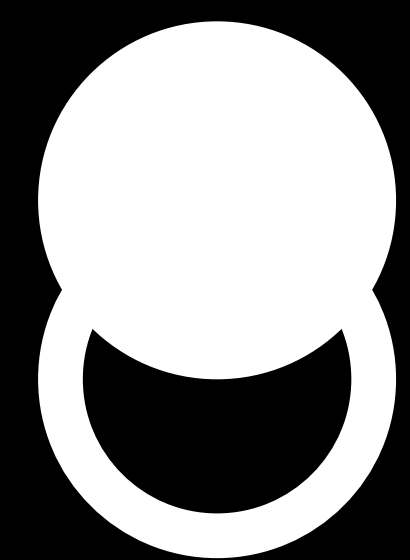
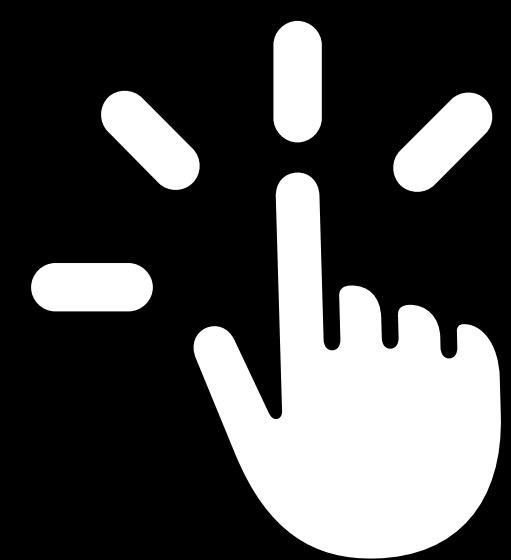
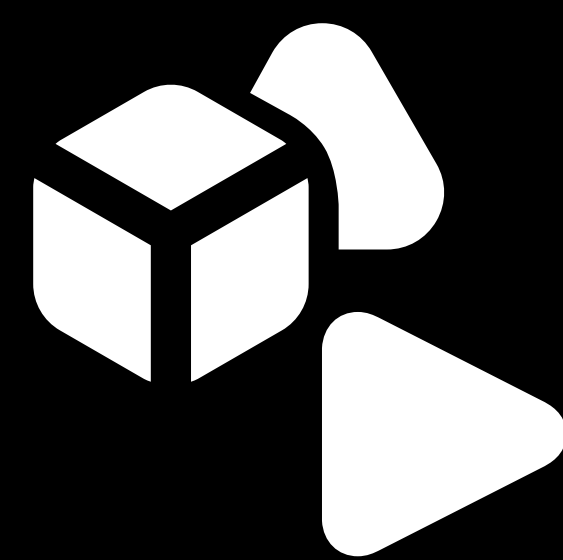
Notify Action

Defining closure

```
seasonsChapter.actions.displayEarthDetails.onAction = { entity in  
    // Display details about Fall  
}
```

UI Applications and AR Content

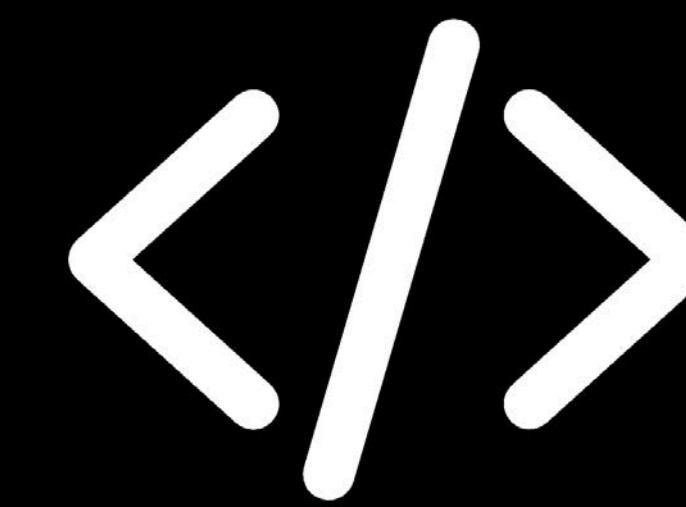
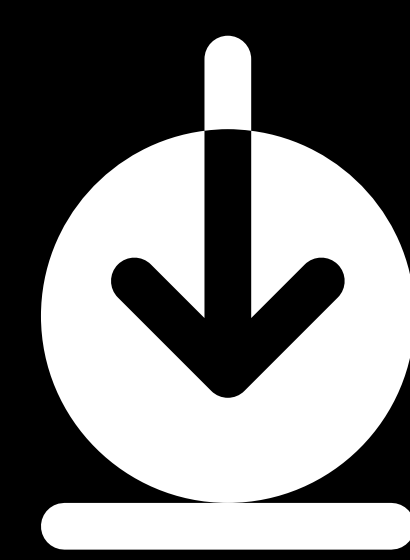
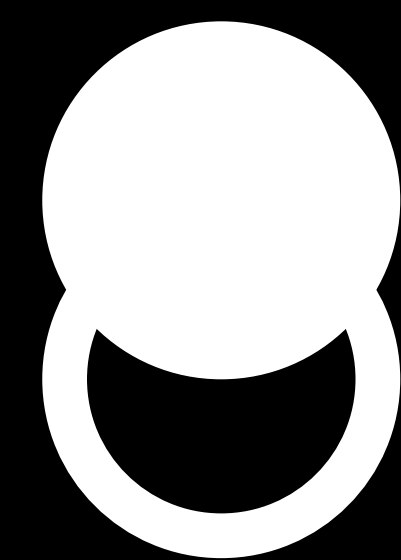
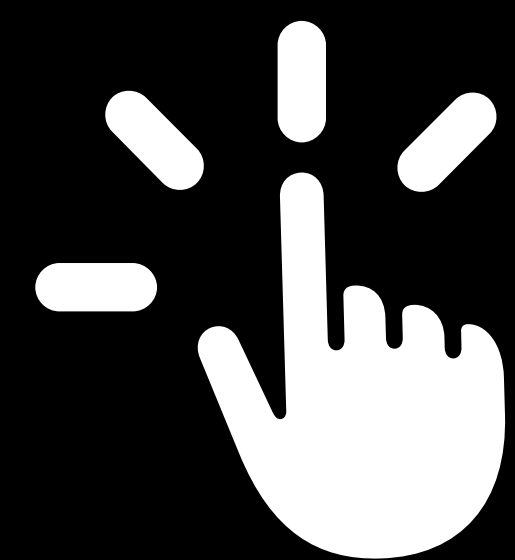
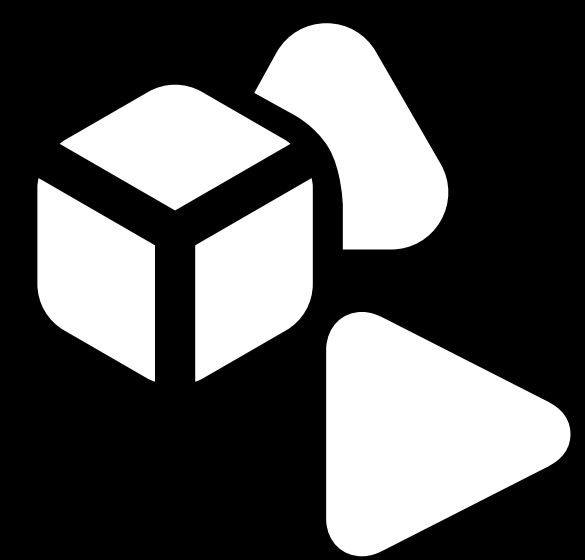
Triggers



Application Logic

UI Applications and AR Content

Triggers



+



Application Logic

Notification Trigger

Notification Trigger

Set up in Reality Composer

Starts action sequences

Posted from the application

Accessible by name in code

 Notification

Identifier

Show Gold Star

Action Sequence begins when the above notification is posted from your application code (Xcode).

MacBook Pro

 Notification

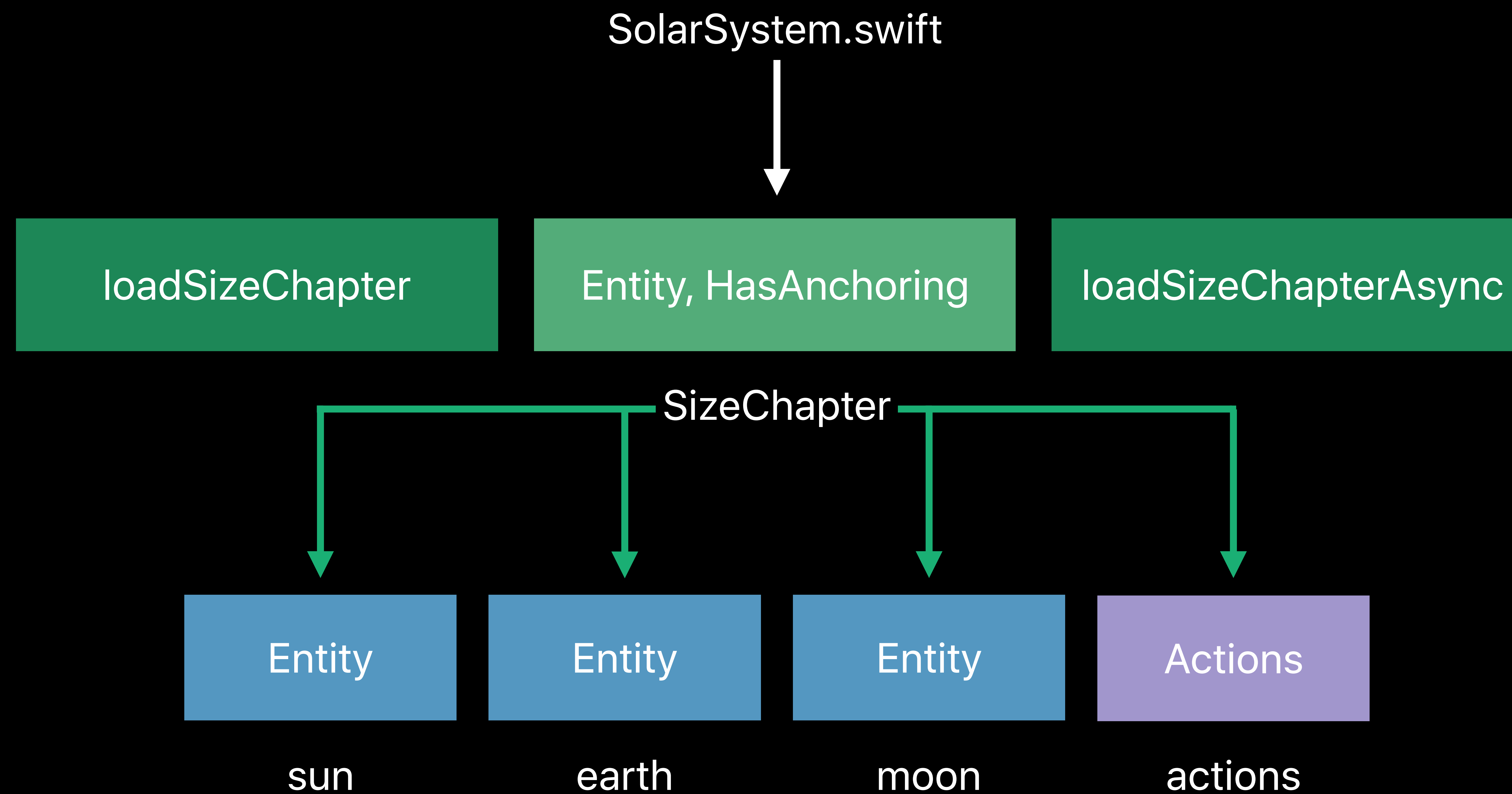
Identifier

Show Gold Star

Action Sequence begins when the above notification is posted from your application code (Xcode).

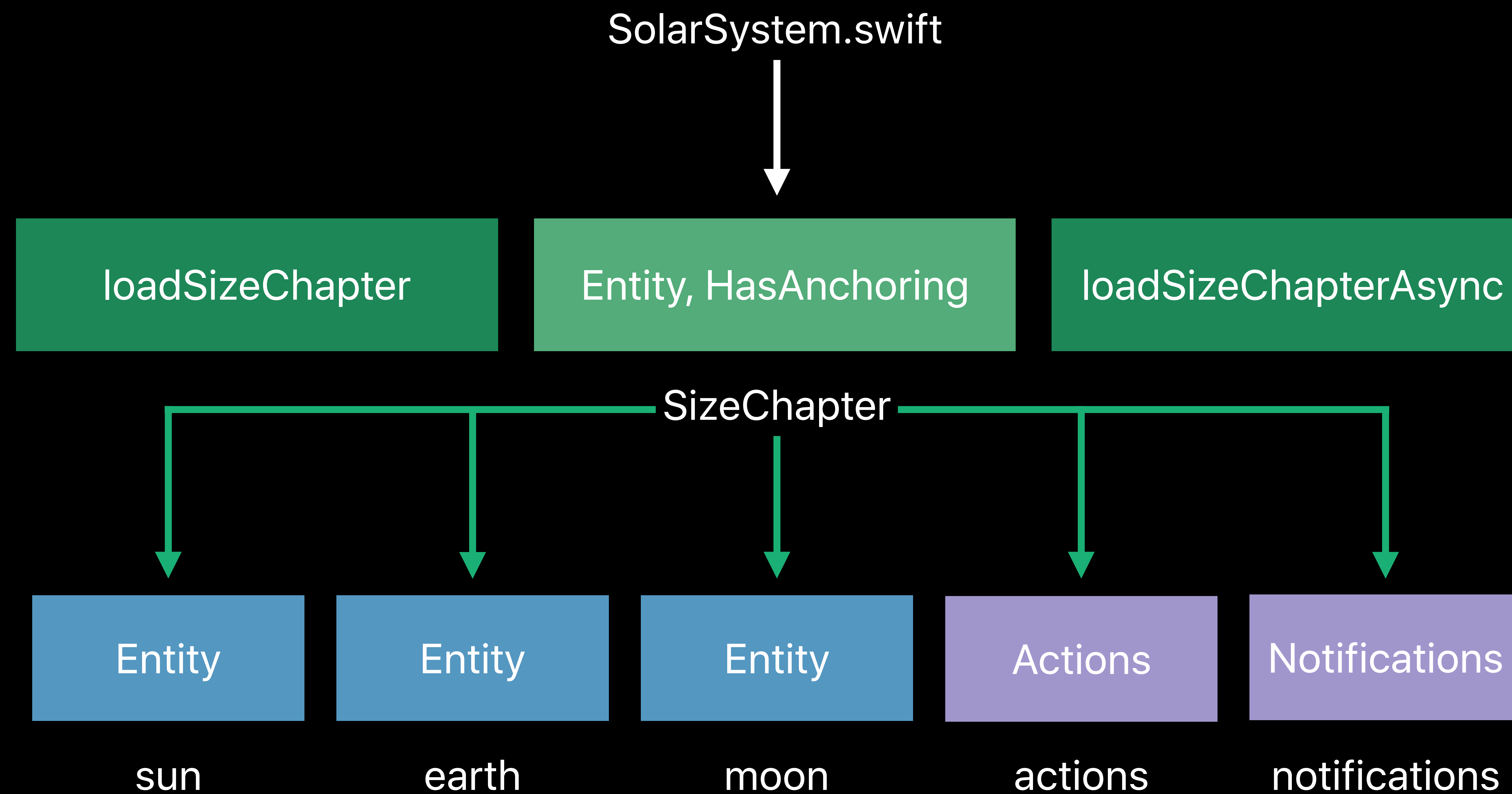
Notification Trigger

Generated code structure



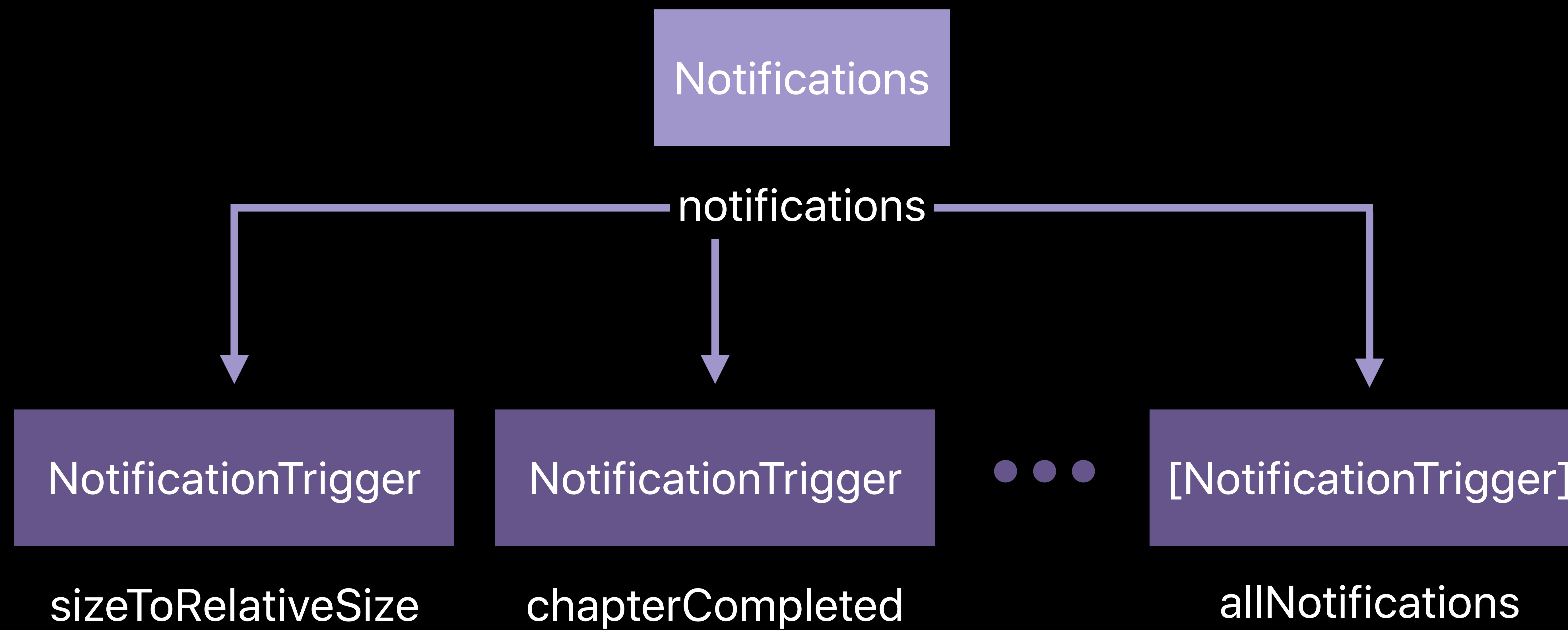
Notification Trigger

Generated code structure



Notification Trigger

Generated code structure



Notification Trigger

Starting an action sequence

```
seasonsChapter.notifications.showGoldStar.post()
```

```
// Replaces targets in the action sequence named `originalTarget.name` with `newTarget`  
seasonsChapter.notifications.showGoldStar.post(overrides: [originalTarget.name: newTarget])
```

Notification Trigger

Starting an action sequence

```
seasonsChapter.notifications.showGoldStar.post()
```

```
// Replaces targets in the action sequence named `originalTarget.name` with `newTarget`  
seasonsChapter.notifications.showGoldStar.post(overrides: [originalTarget.name: newTarget])
```


Notification Trigger

Starting an action sequence

```
seasonsChapter.notifications.showGoldStar.post()
```

```
// Replaces targets in the action sequence named `originalTarget.name` with `newTarget`
```

```
seasonsChapter.notifications.showGoldStar.post(overrides: [originalTarget.name: newTarget])
```

Demo

Reality Composer

Get started with AR and 3D

Layout and pre-visualization

Content library

Simple interactions

Xcode integration

More Information

developer.apple.com/wwdc19/609

RealityKit and Reality Composer Lab

Thursday, 3:00

Advances in AR Quick Look

Friday, 9:00

AR Quick Look and Reality Composer Lab

Friday, 11:00
