

# SceneKit in Swift Playgrounds

Session 605

Michael DeWitt, Gem Collector  
Lemont Washington, Draw Call Blaster

**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.

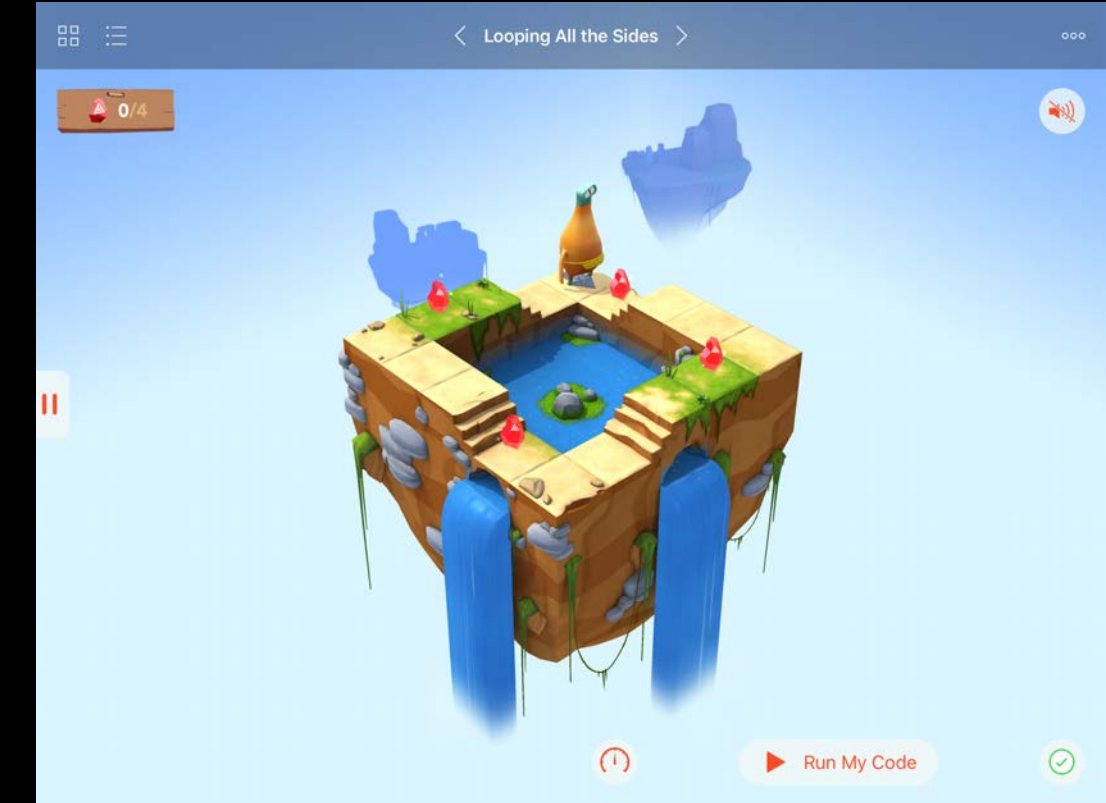
- 1 Drag a for loop from the code library, then drop it above the existing code.
- 2 Tap the bottom curly brace to select the loop.
- 3 Tap and hold that curly brace, then drag it downward to pull the existing code into the loop.

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





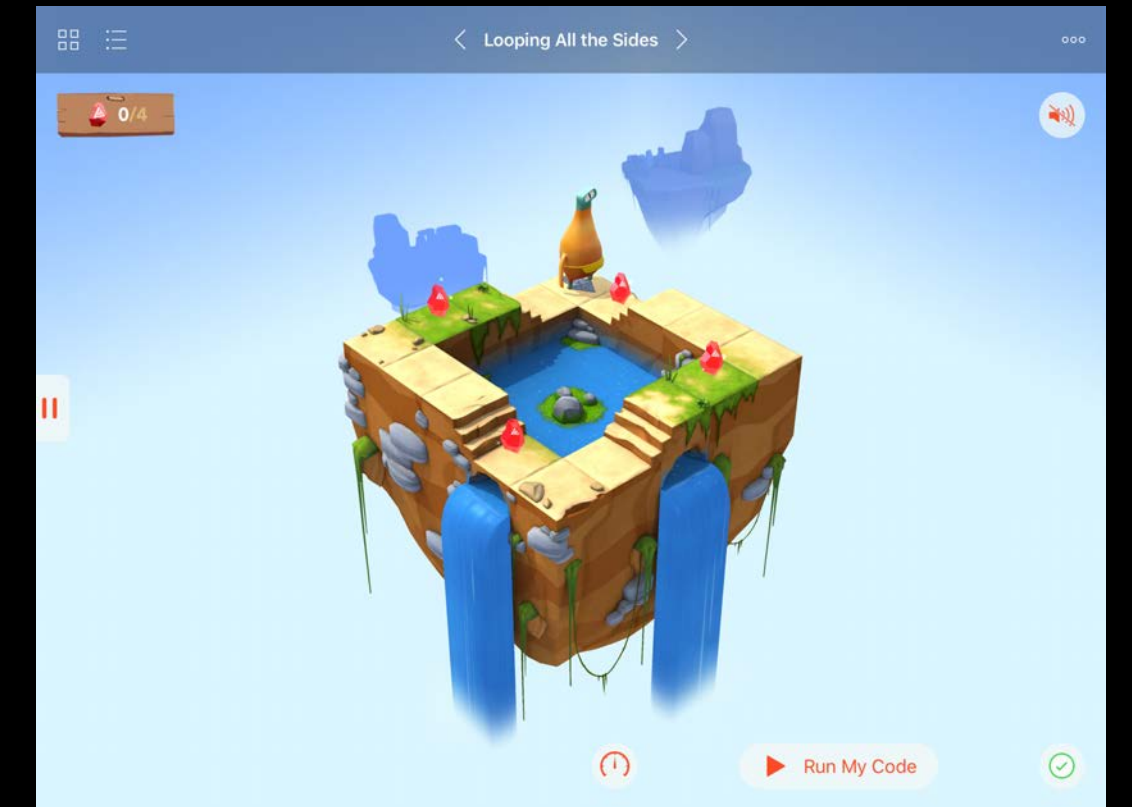
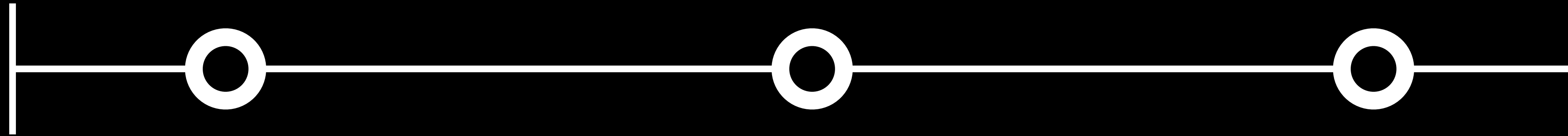




Prototyping

Iterating

Tuning



# Prototyping

**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 ... number {  
  moveForward()  
  collectGem()  
  moveForward()  
  moveForward()  
}
```

```
moveForward()  
turnRight()
```





# Early Feedback



# Early Feedback

Change gem

Add border

Pivot camera at the end



**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 ... number {  
  moveForward()  
  collectGem()  
  moveForward()  
  moveForward()  
}
```

```
moveForward()  
turnRight()
```





SpriteKit

SceneKit





SpriteKit

SKScene

SKNode

SKAction

SceneKit





SpriteKit

SKScene

SKNode

SKAction

SceneKit

SCNScene

SCNNode

SCNAction





SpriteKit

SKScene

SKNode

SKAction

SceneKit

SCNScene

SCNNode

SCNAction



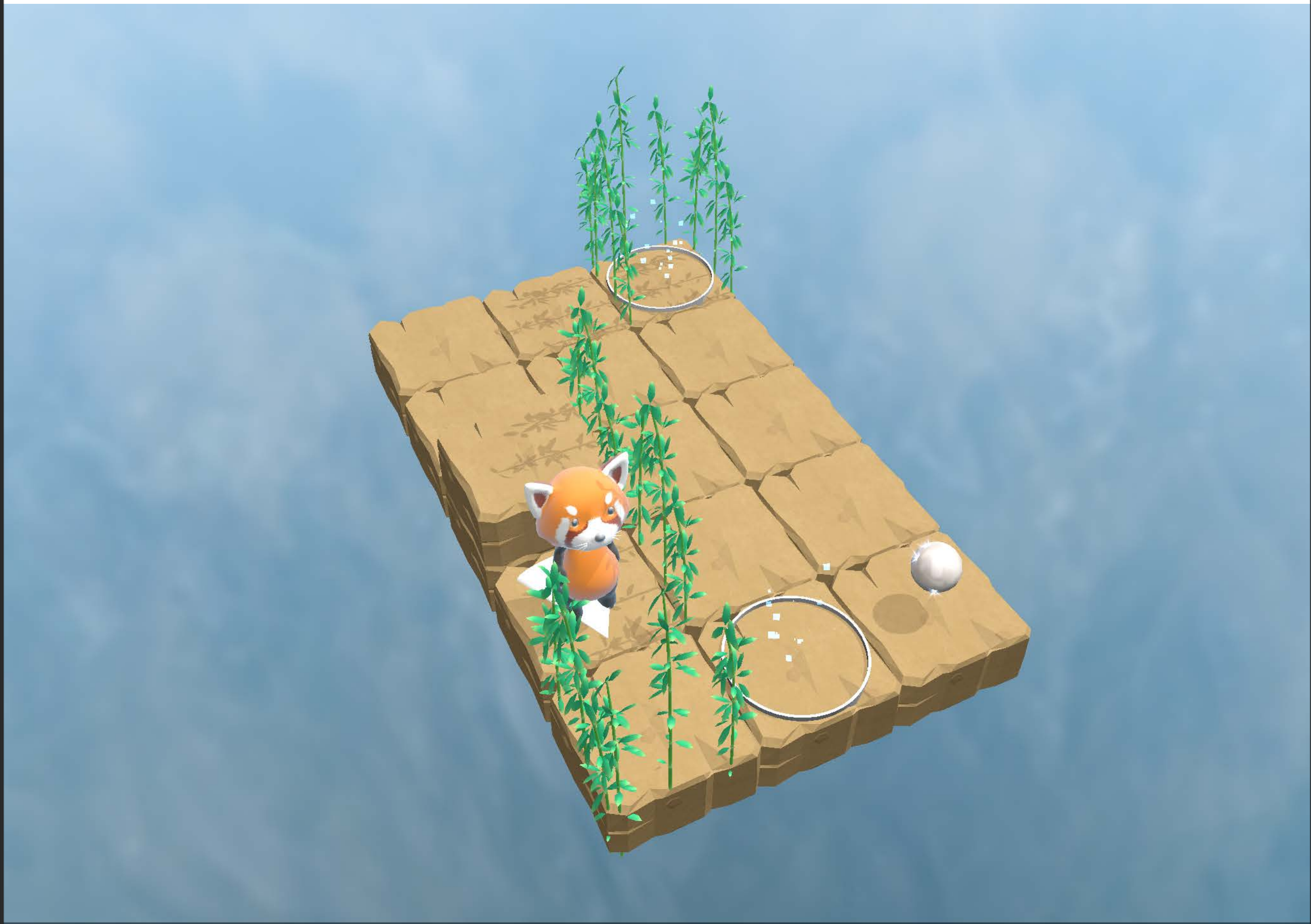




iPad

9:41 AM

100%



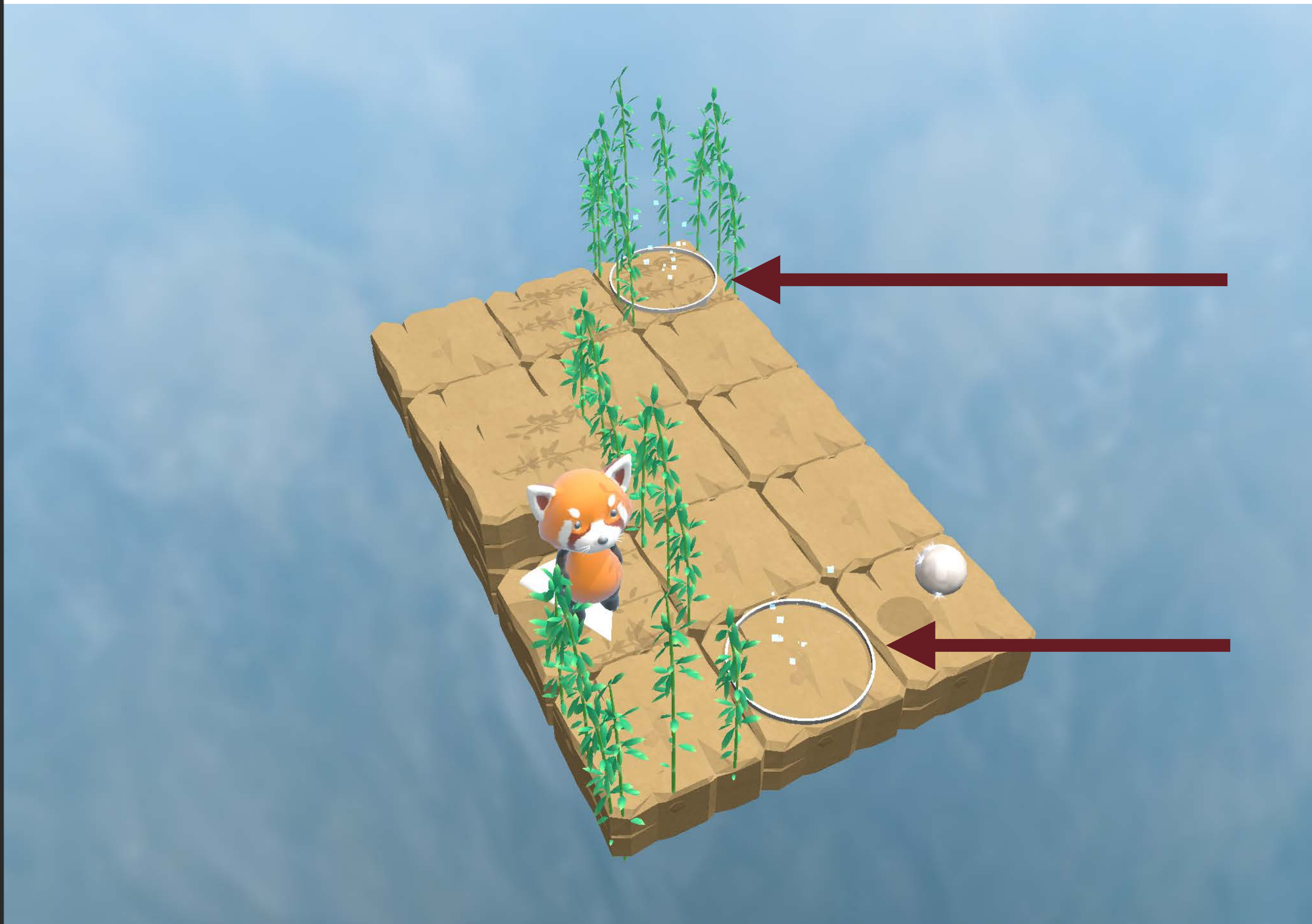
+  Mt 60fps

◆119 ▲47.4K

iPad

9:41 AM

100%



+  Mt 60fps

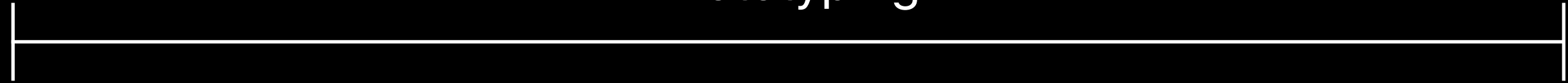
119 47.4K

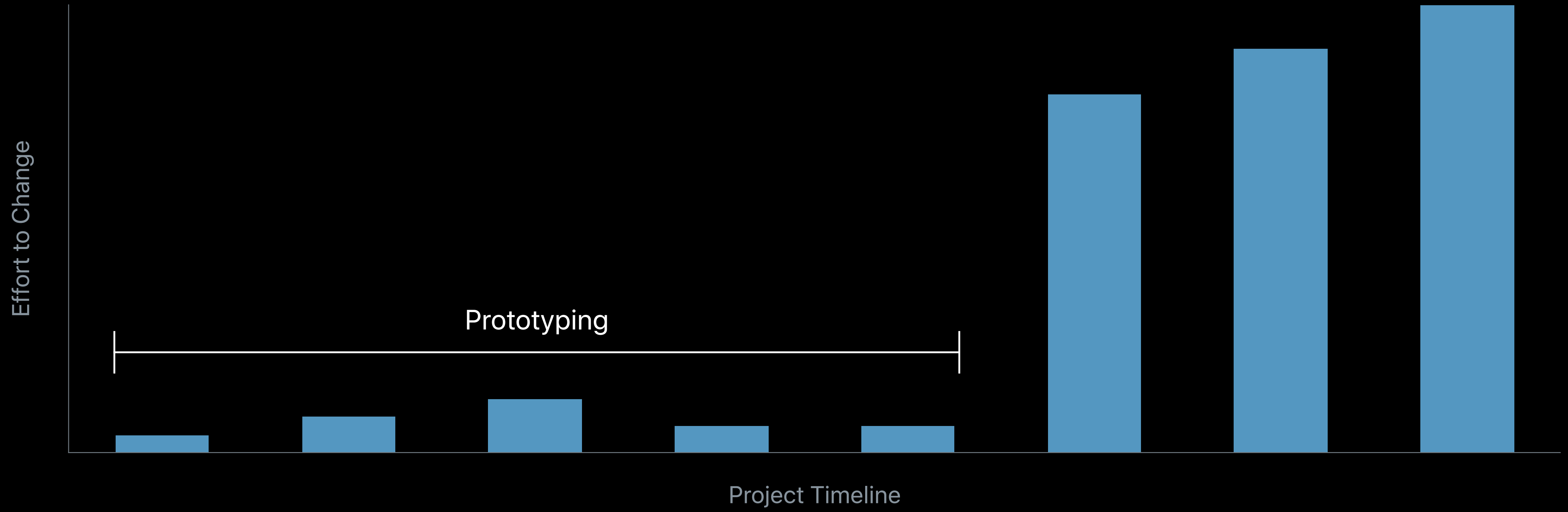


Effort to Change

Prototyping

Project Timeline





# Prototyping



# Prototyping

Test interaction model



# Prototyping

Test interaction model

Interpret feedback





# Prototyping

Test interaction model

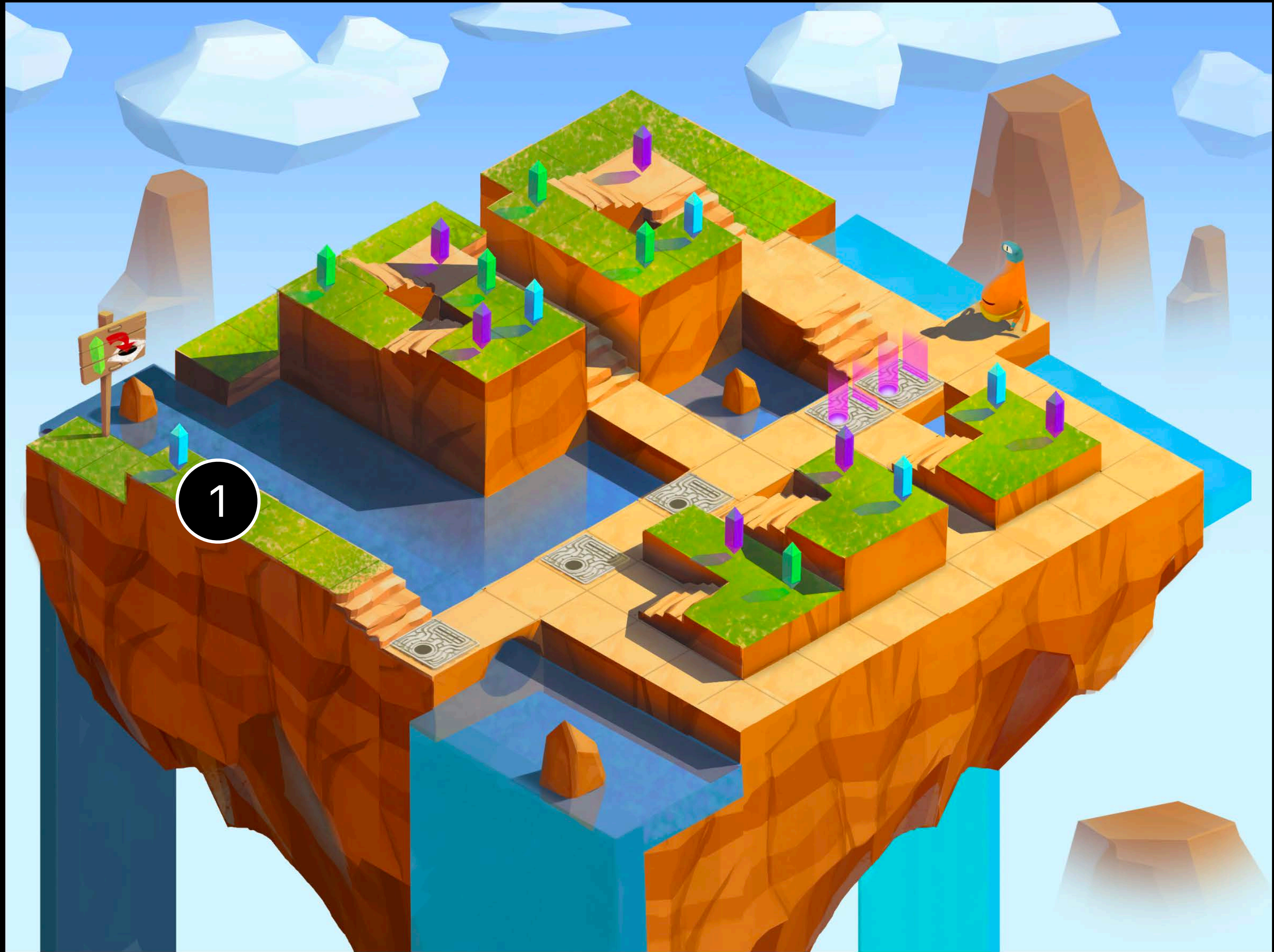
Interpret feedback

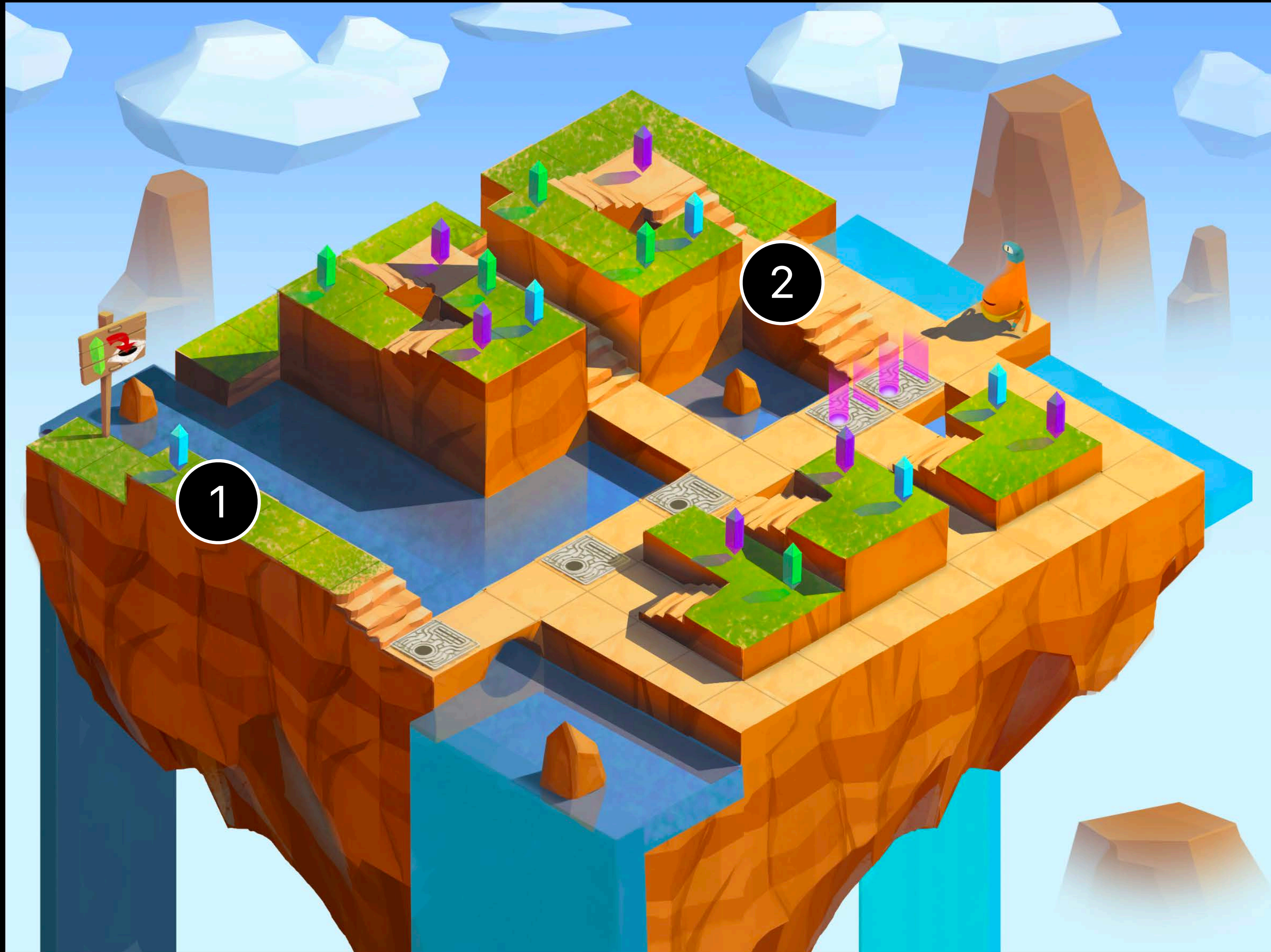
Take the insights, leave the code



**Iterating**











1

2

3

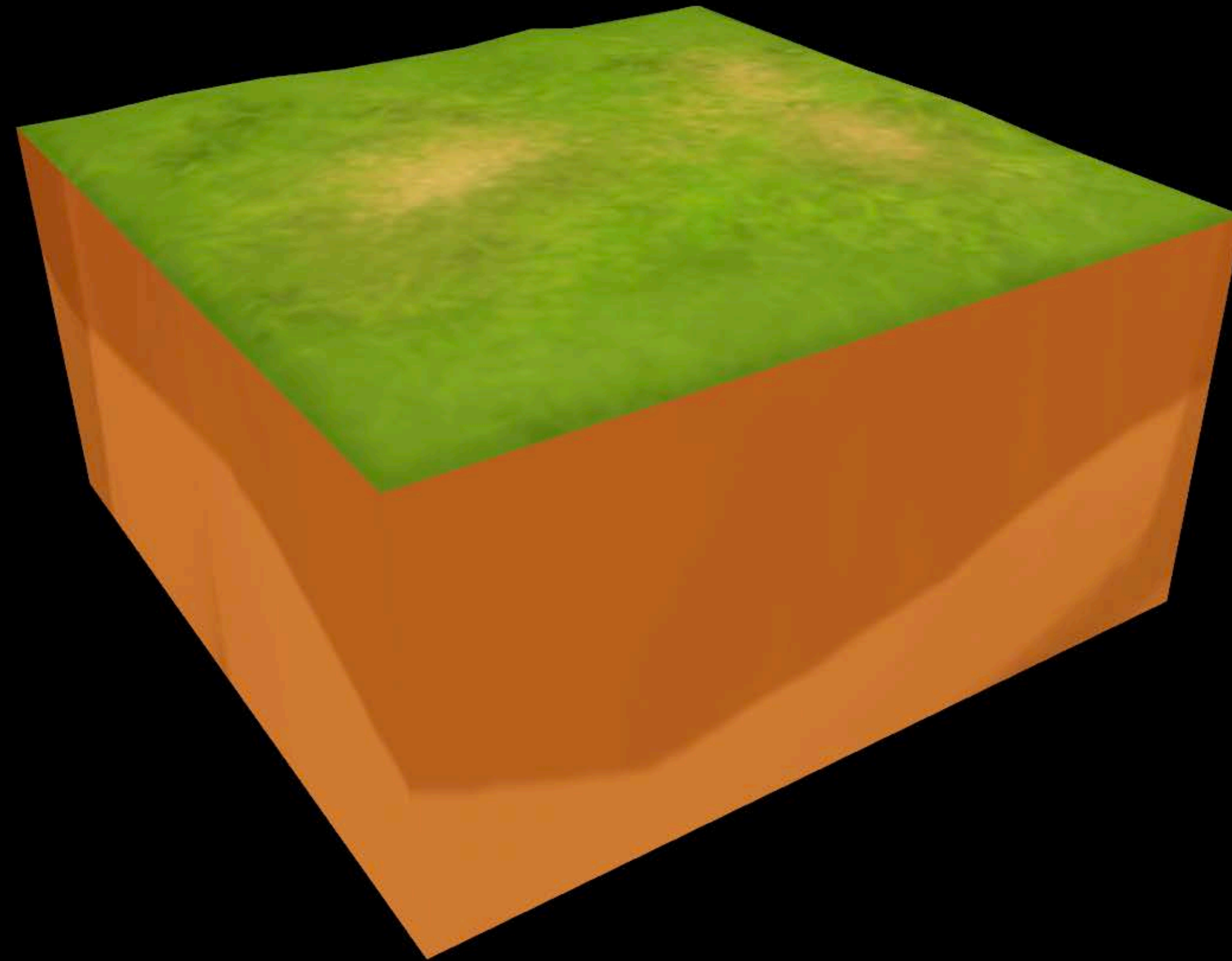
4

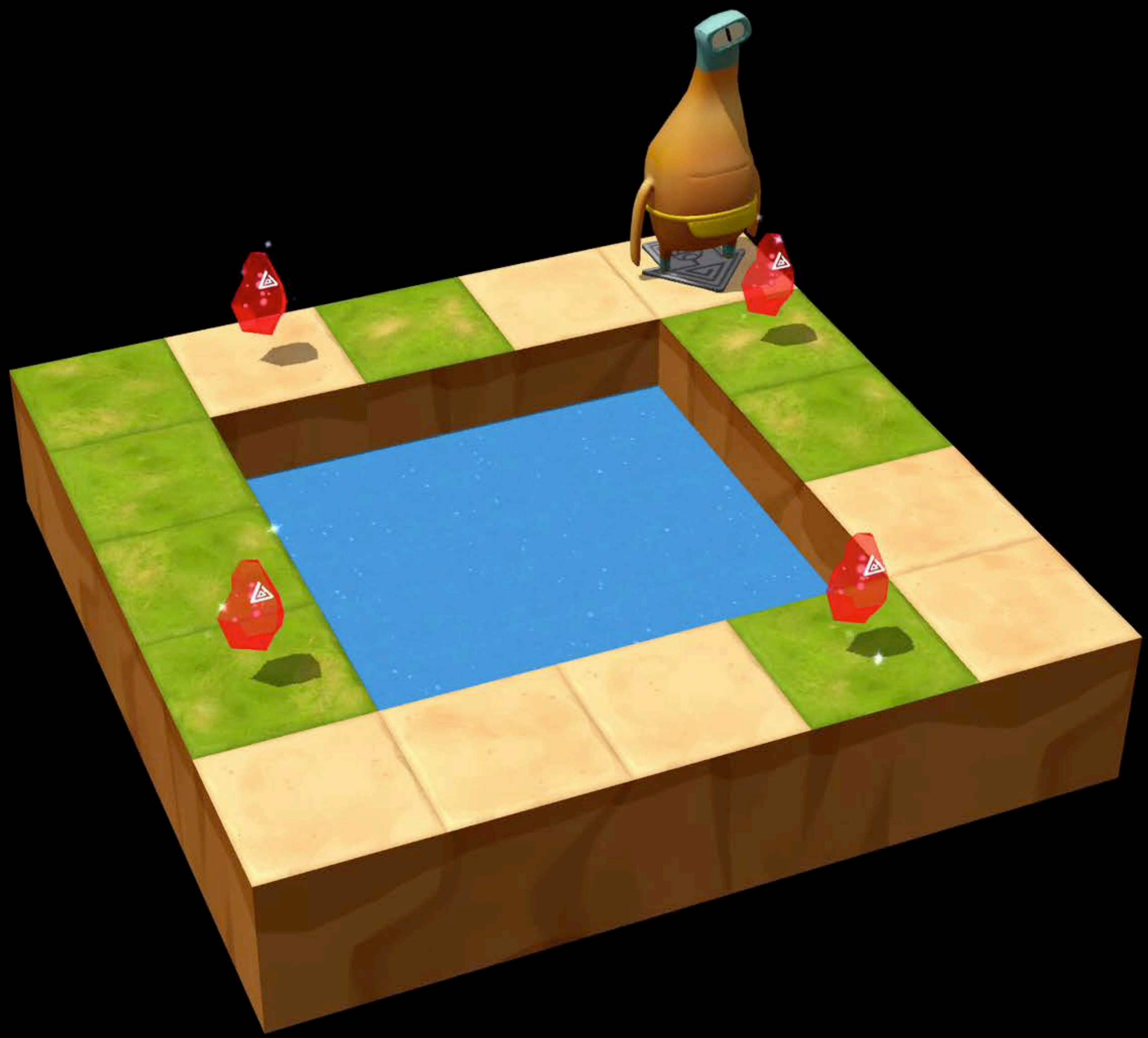
# Modeling the World





# Modeling the World







```
// World building API
```

```
let world = GridWorld(columns: 5, rows: 5)
```

```
// Add an actor to the scene.
```

```
let actor = Actor()
```

```
world.place(actor, facing: north, at: Coordinate(column: 0, row: 0))
```

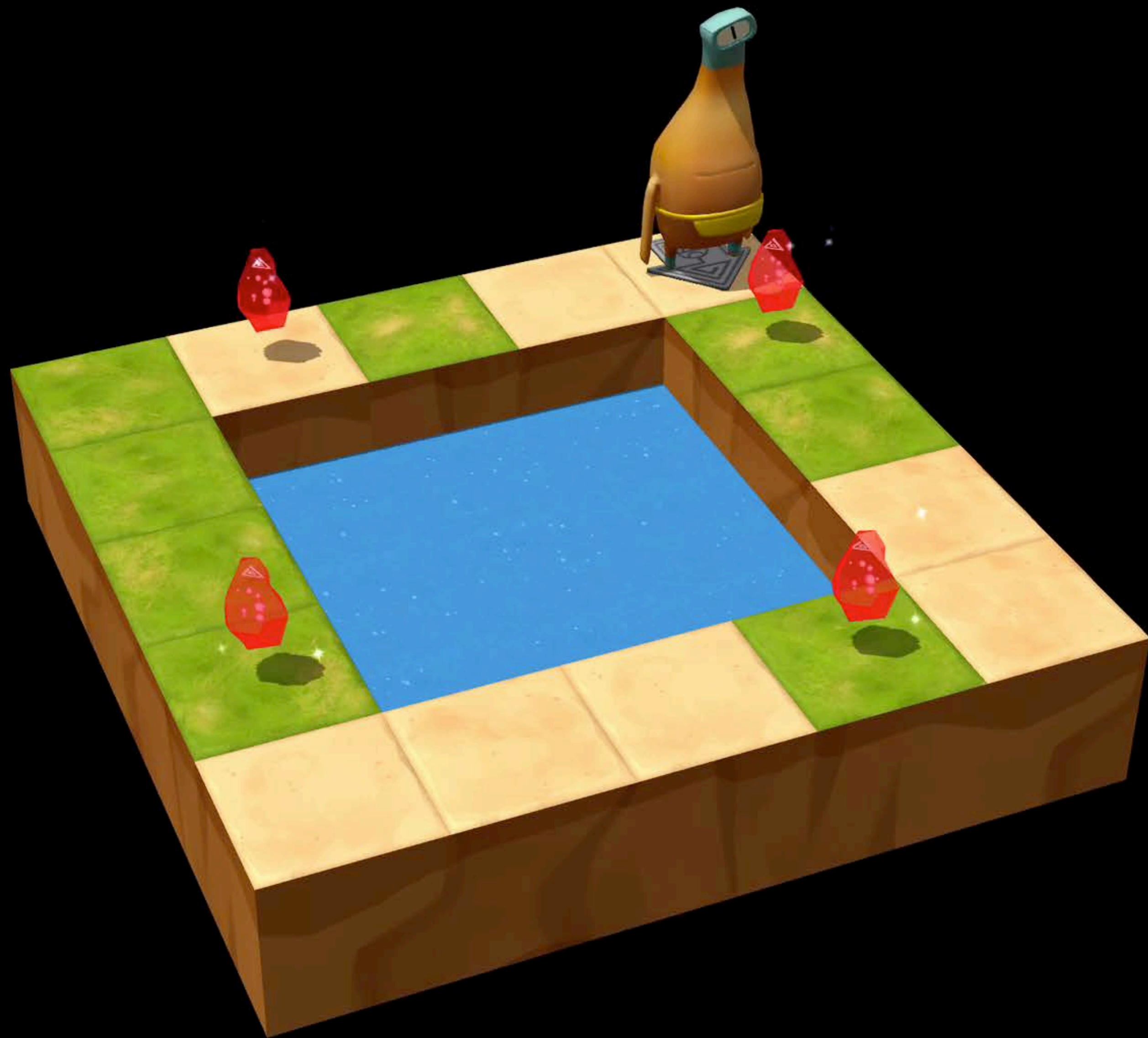
```
// World building API

let world = GridWorld(columns: 5, rows: 5)

// Add an actor to the scene.
let actor = Actor()
world.place(actor, facing: north, at: Coordinate(column: 0, row: 0))

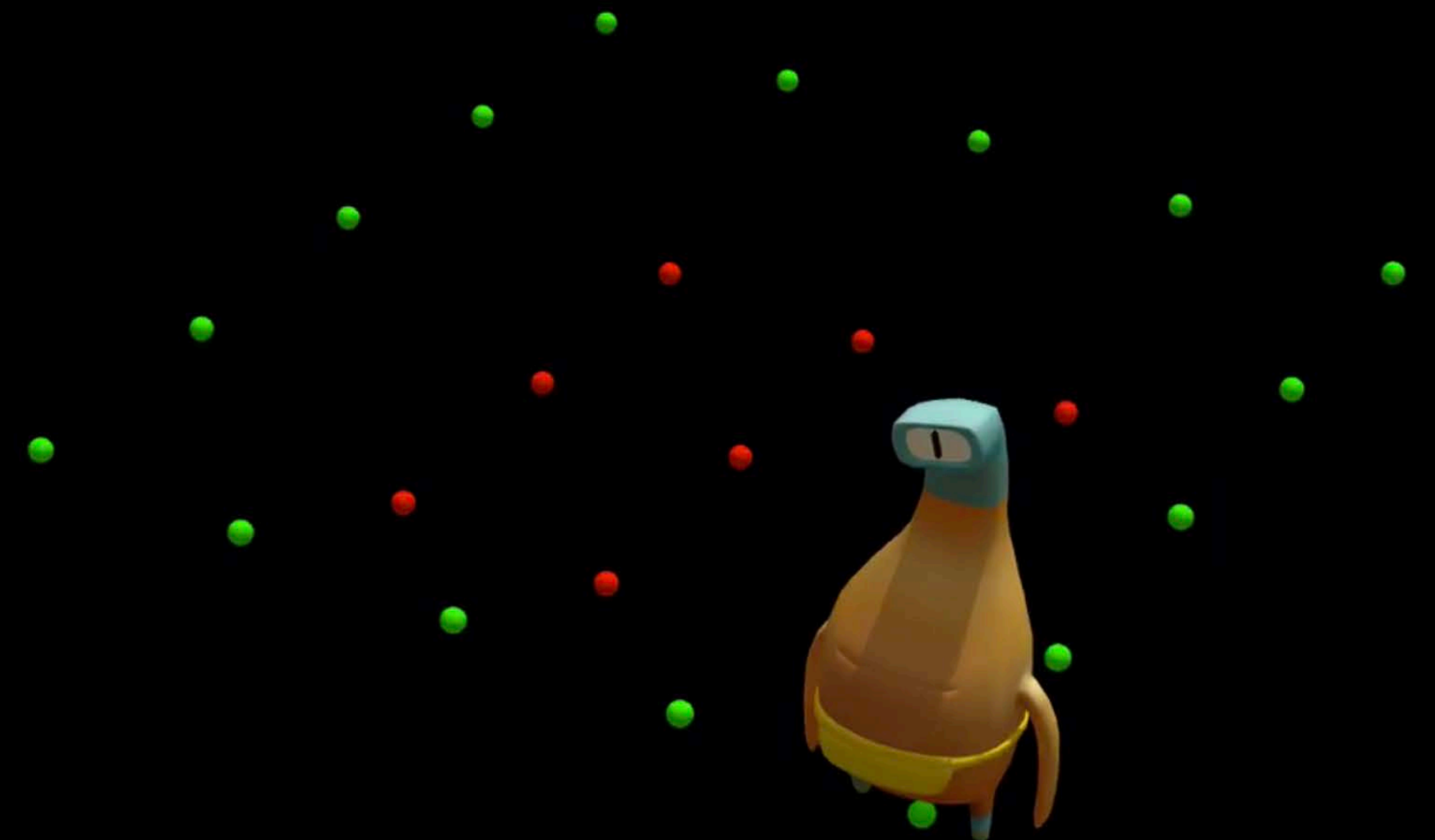
// Place additional items.
let gems = [
    Coordinate(column: 0, row: 1),
    Coordinate(column: 1, row: 4),
    Coordinate(column: 3, row: 0),
    Coordinate(column: 4, row: 3)
]
world.placeGems(at: gems)

let water = world.coordinates(inColumns: 1..<4, intersectingRows: 1..<4)
world.placeWater(at: water)
```



# Modeling the World

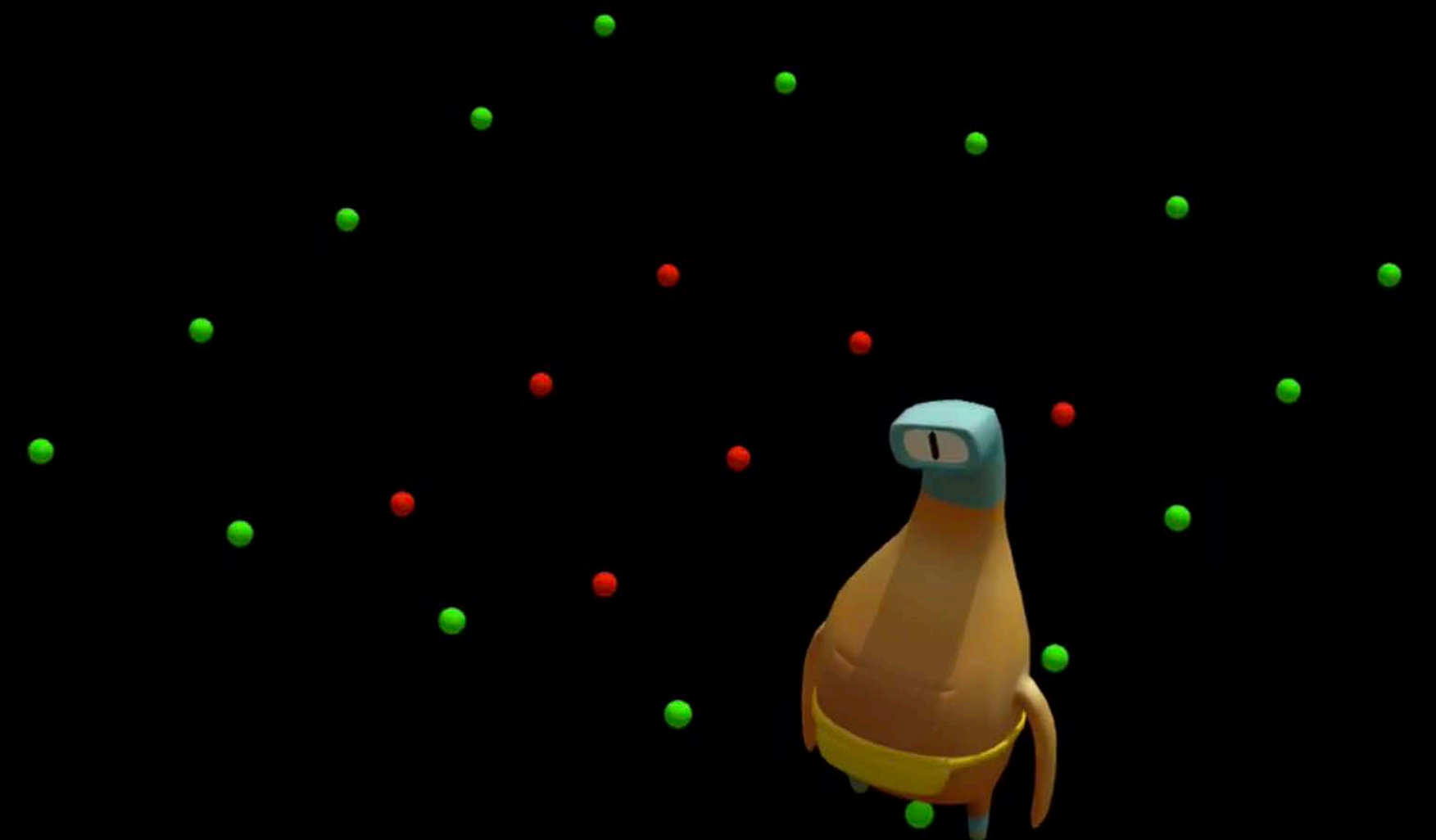
Separate data and visuals



# Modeling the World

Separate data and visuals

Swap assets easily



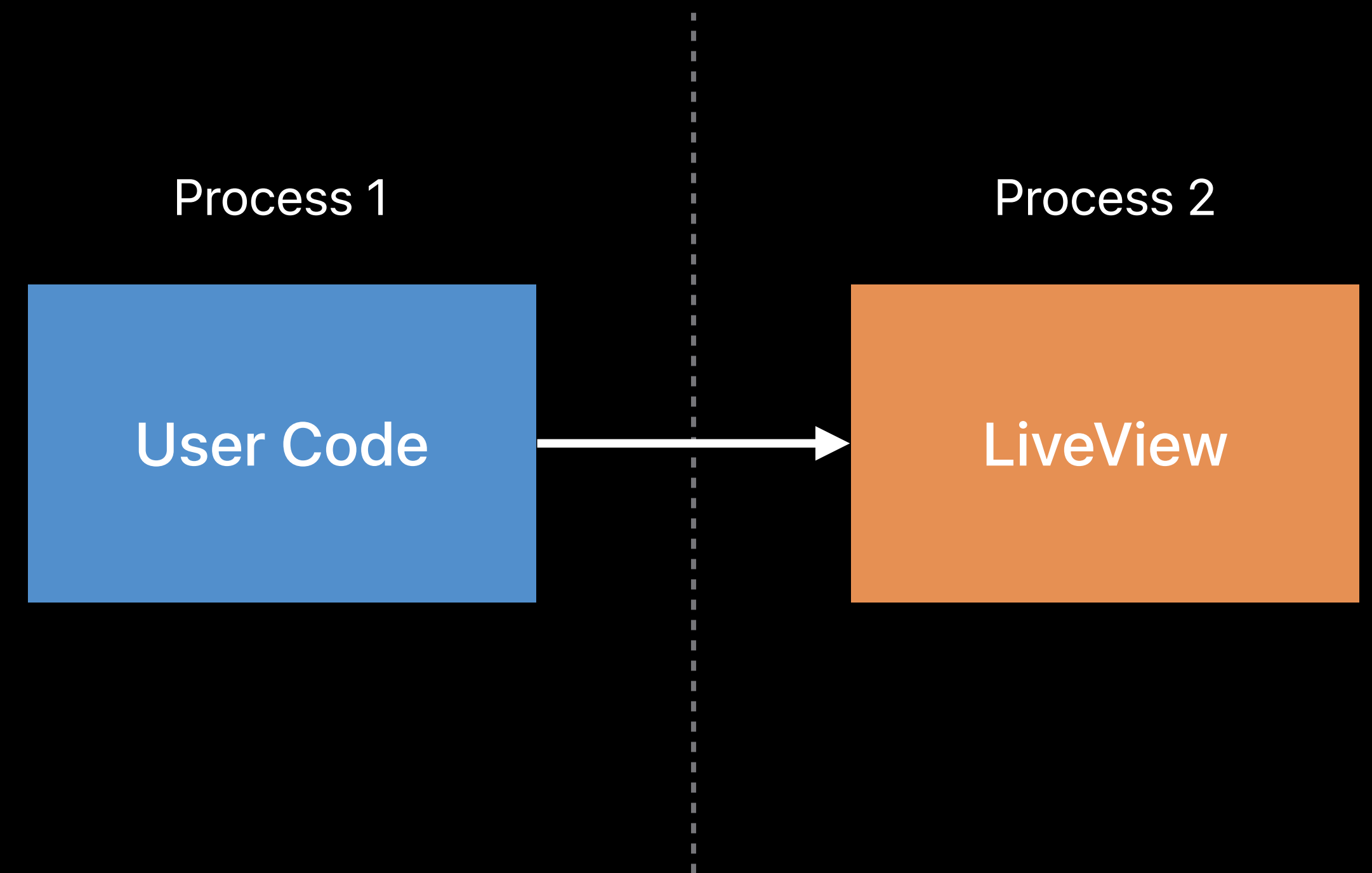


# Modeling the World

Separate data and visuals

Swap assets easily

Send gameplay logic



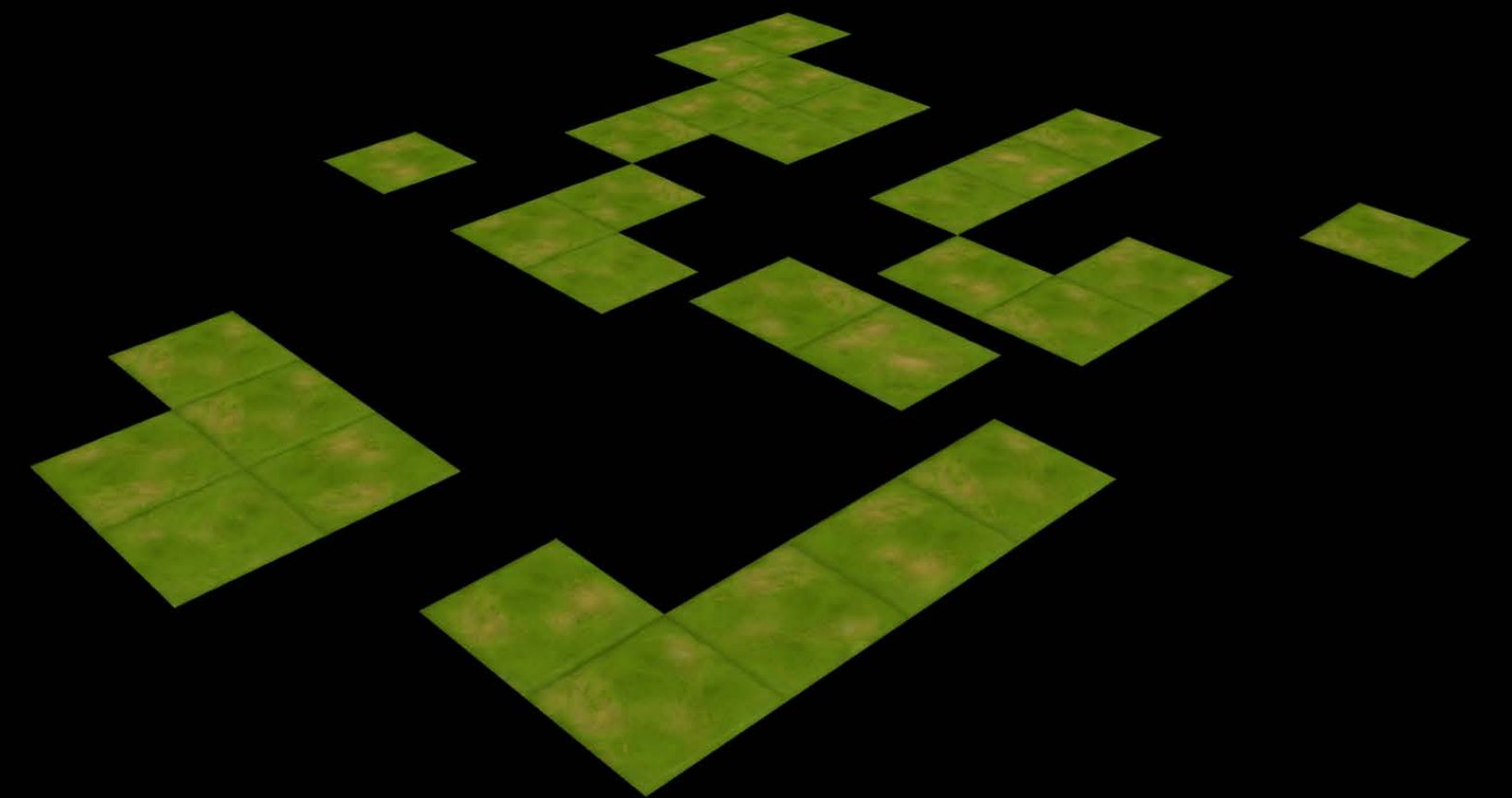
# Modeling the World

Separate data and visuals

Swap assets easily

Send gameplay logic

Optimize the geometry



# Modeling the World

Separate data and visuals

Swap assets easily

Send gameplay logic

Optimize the geometry

Need debugging tools ⚠

# Modeling the World

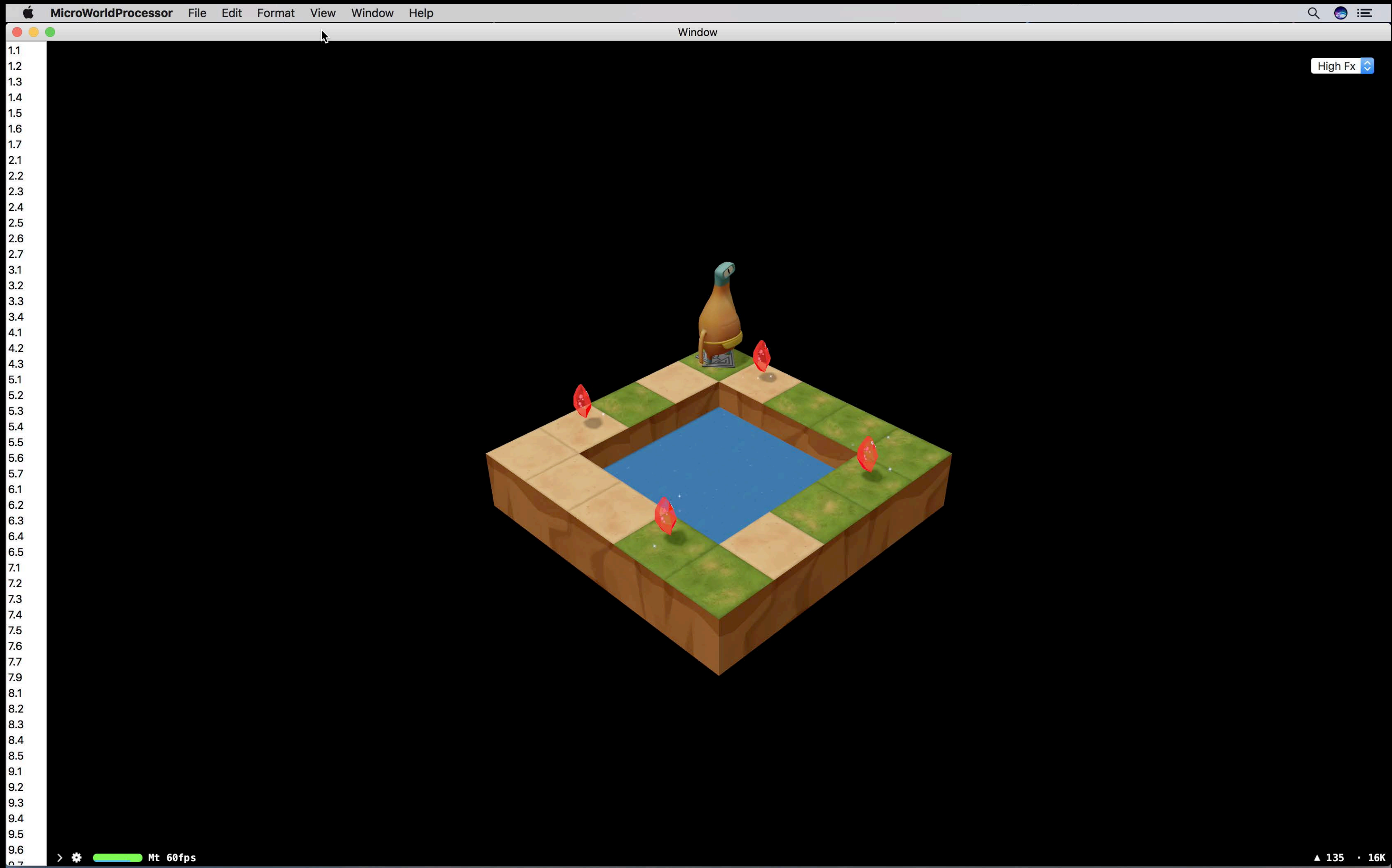
Separate data and visuals

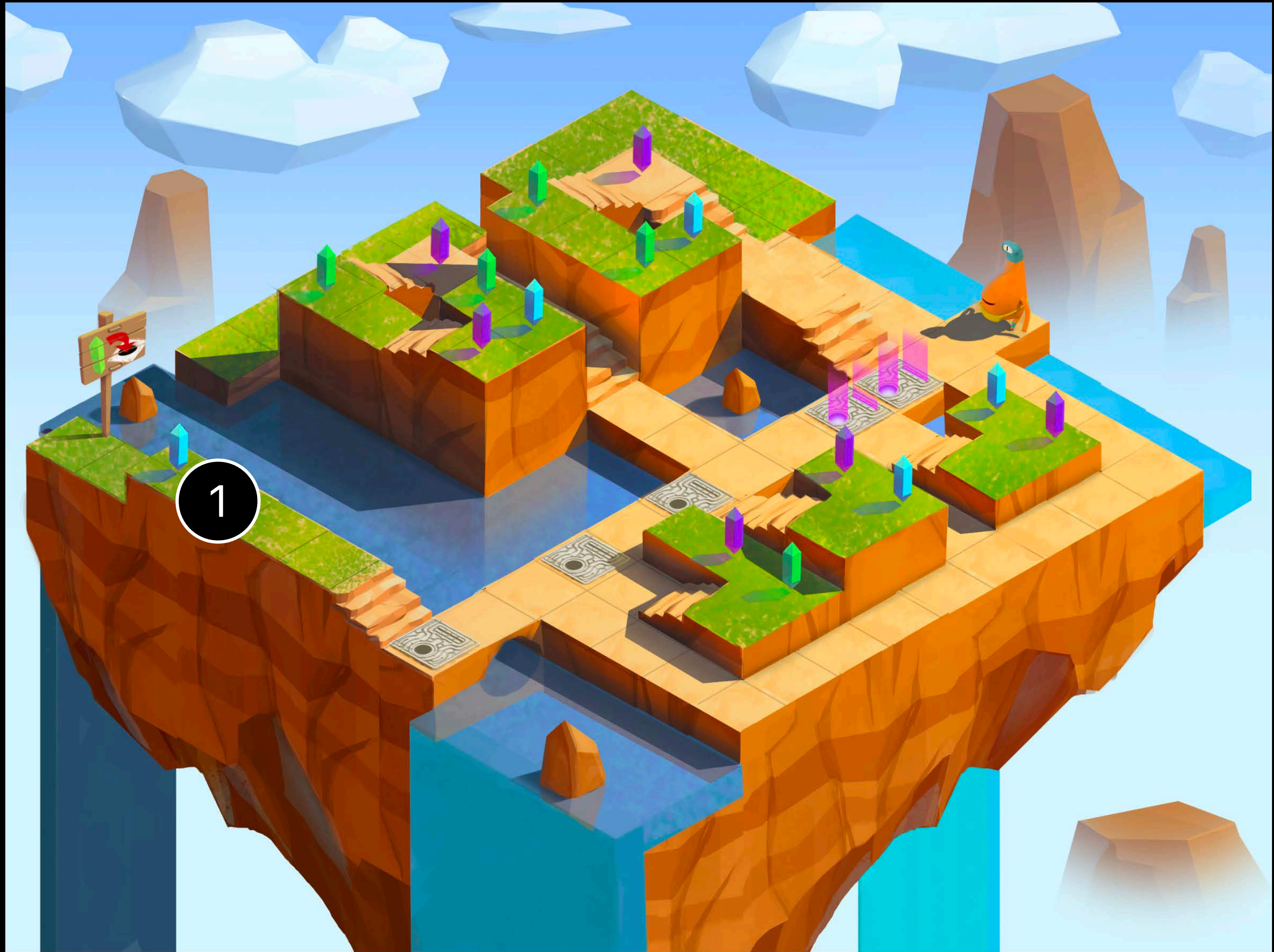
Swap assets easily

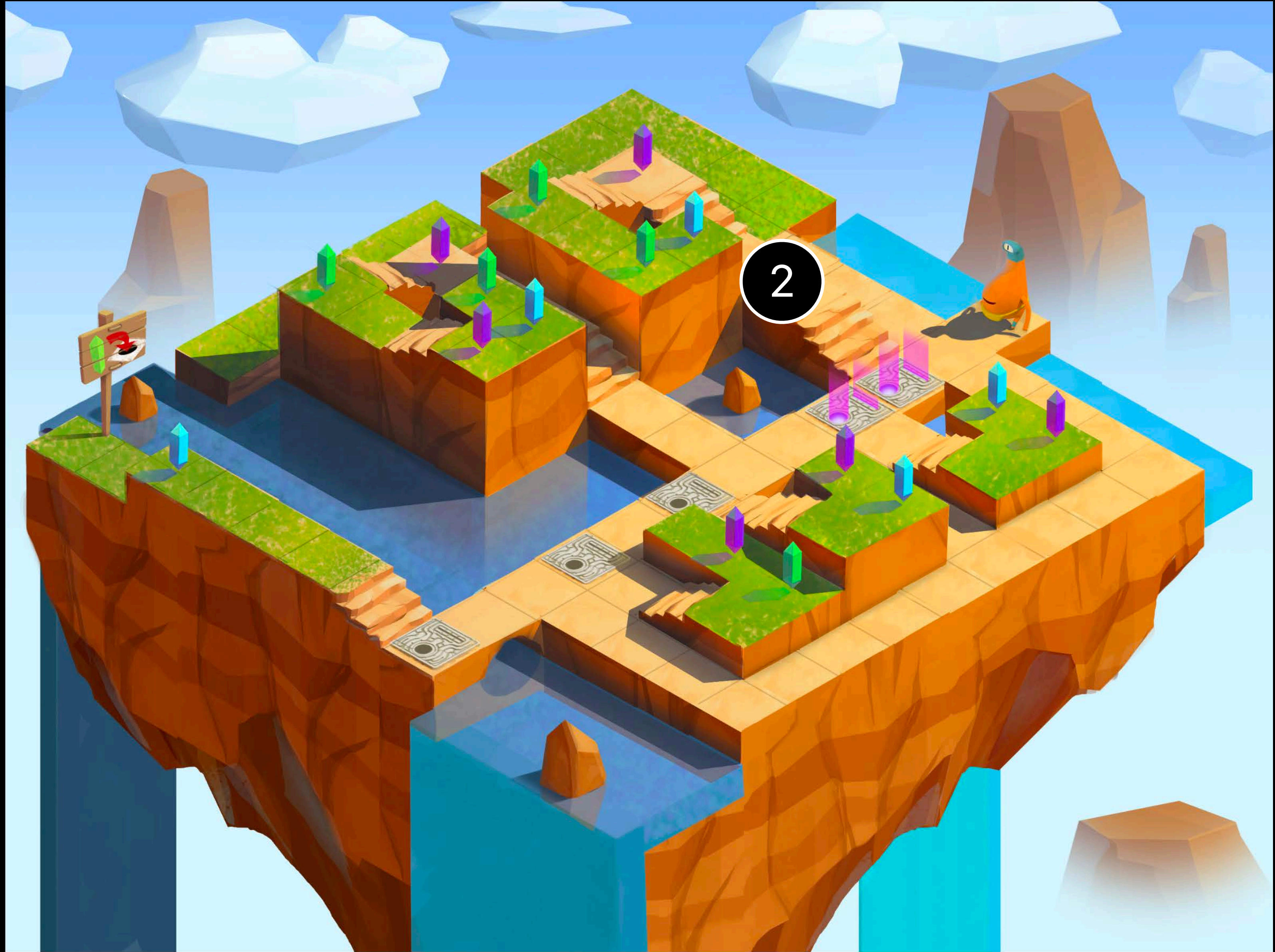
Send gameplay logic

Optimize the geometry

Need debugging tools ⚠





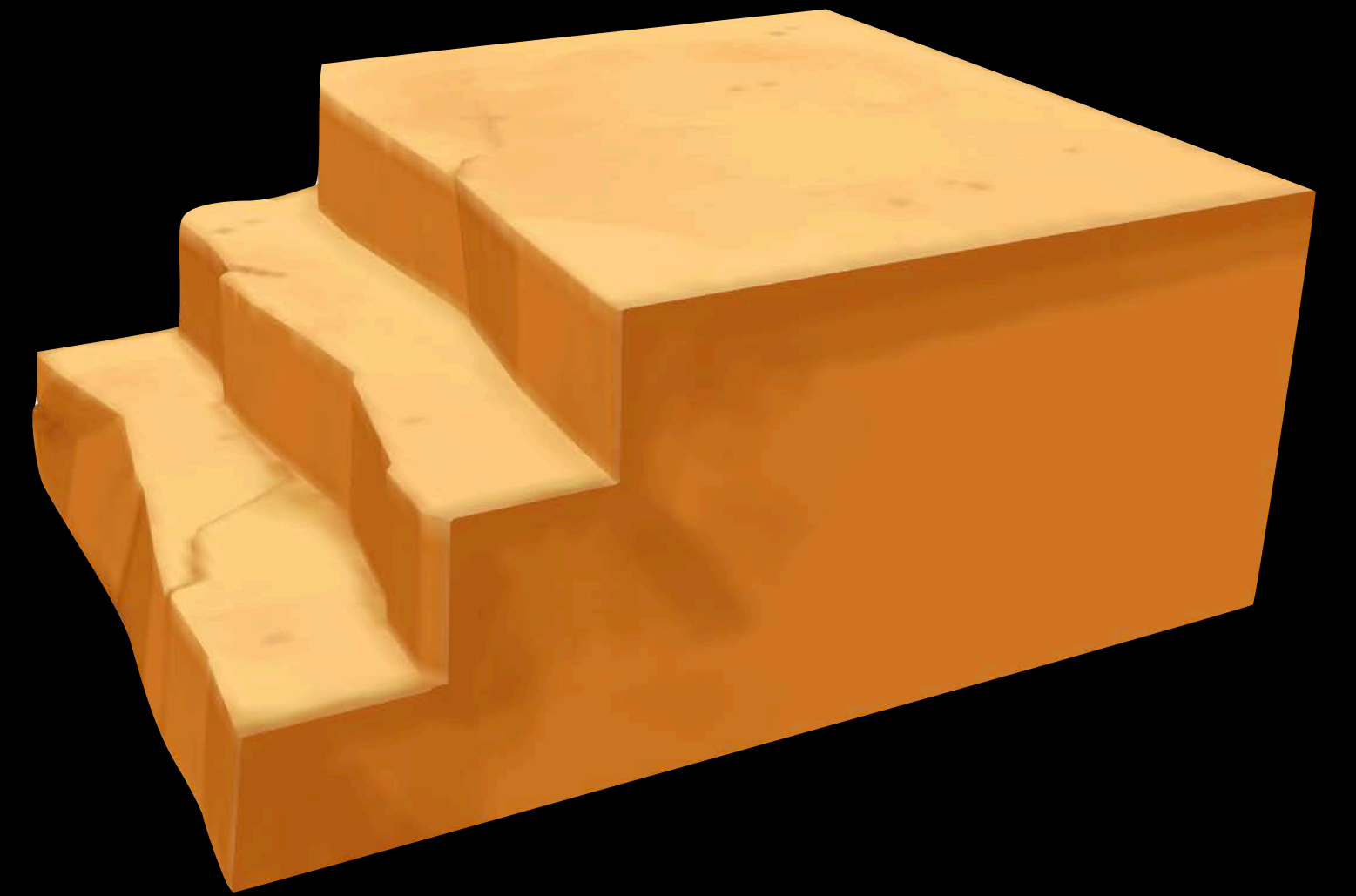


# Animating Up



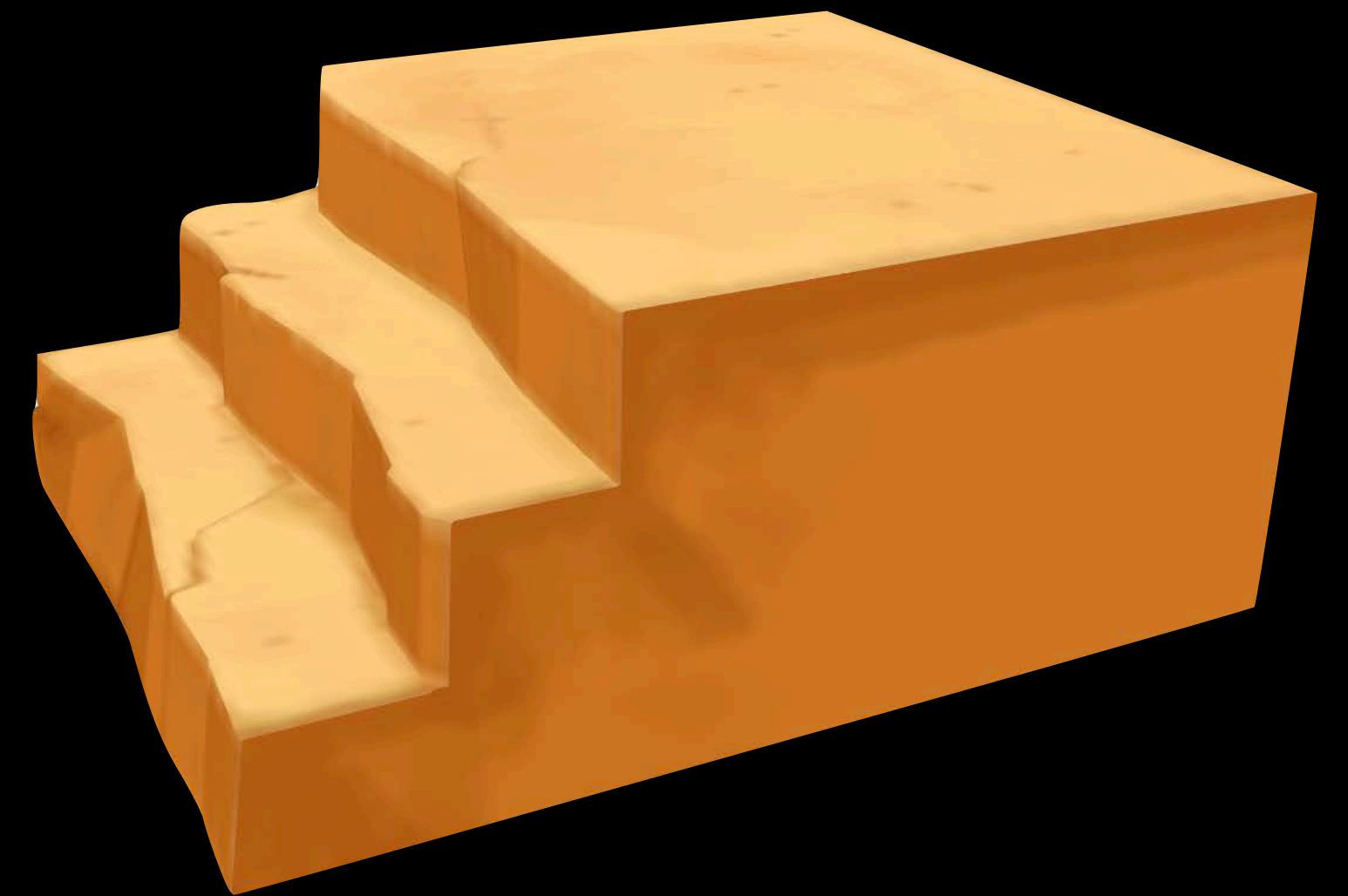


# Animating Up



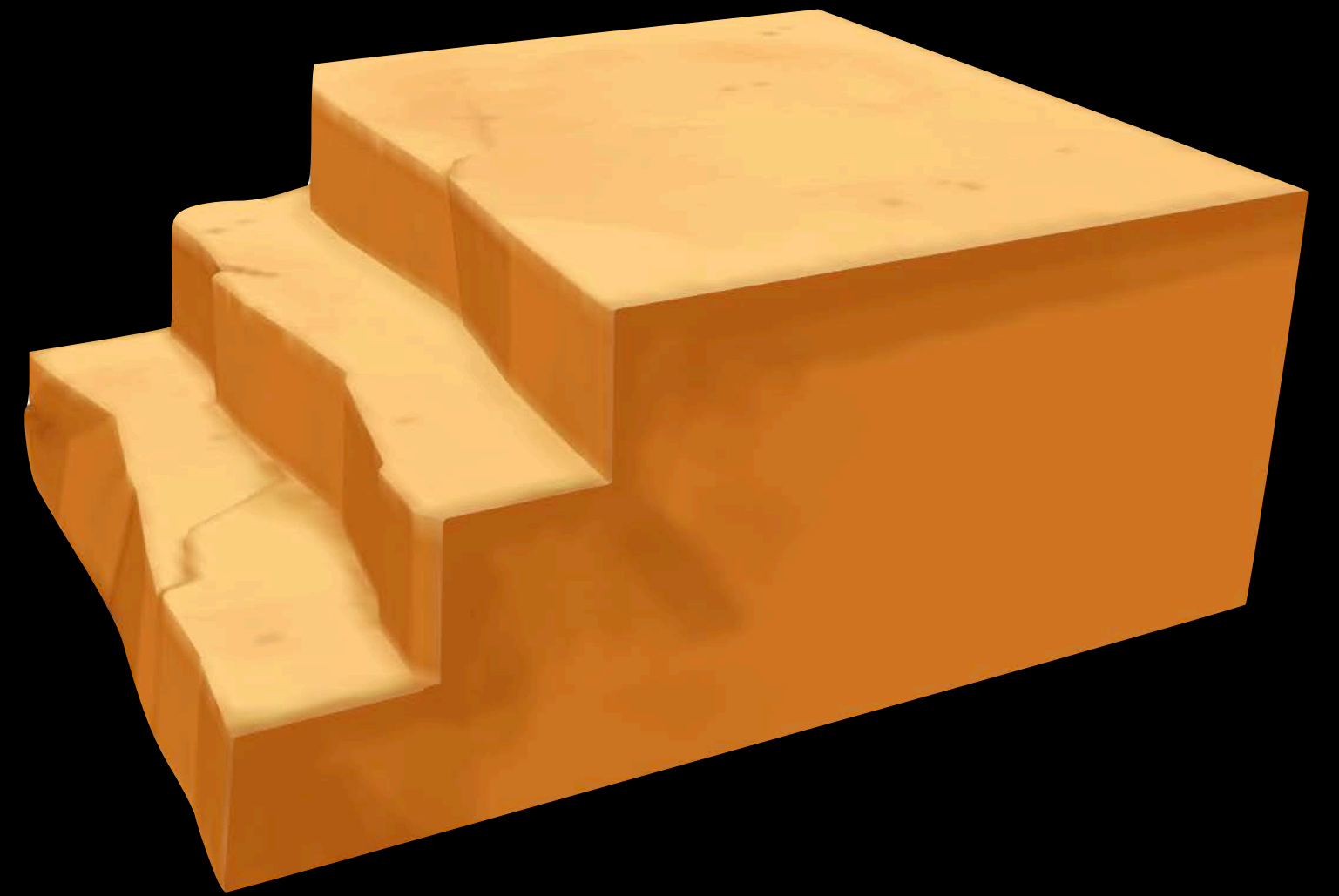
# Animating Up

Use a ramp



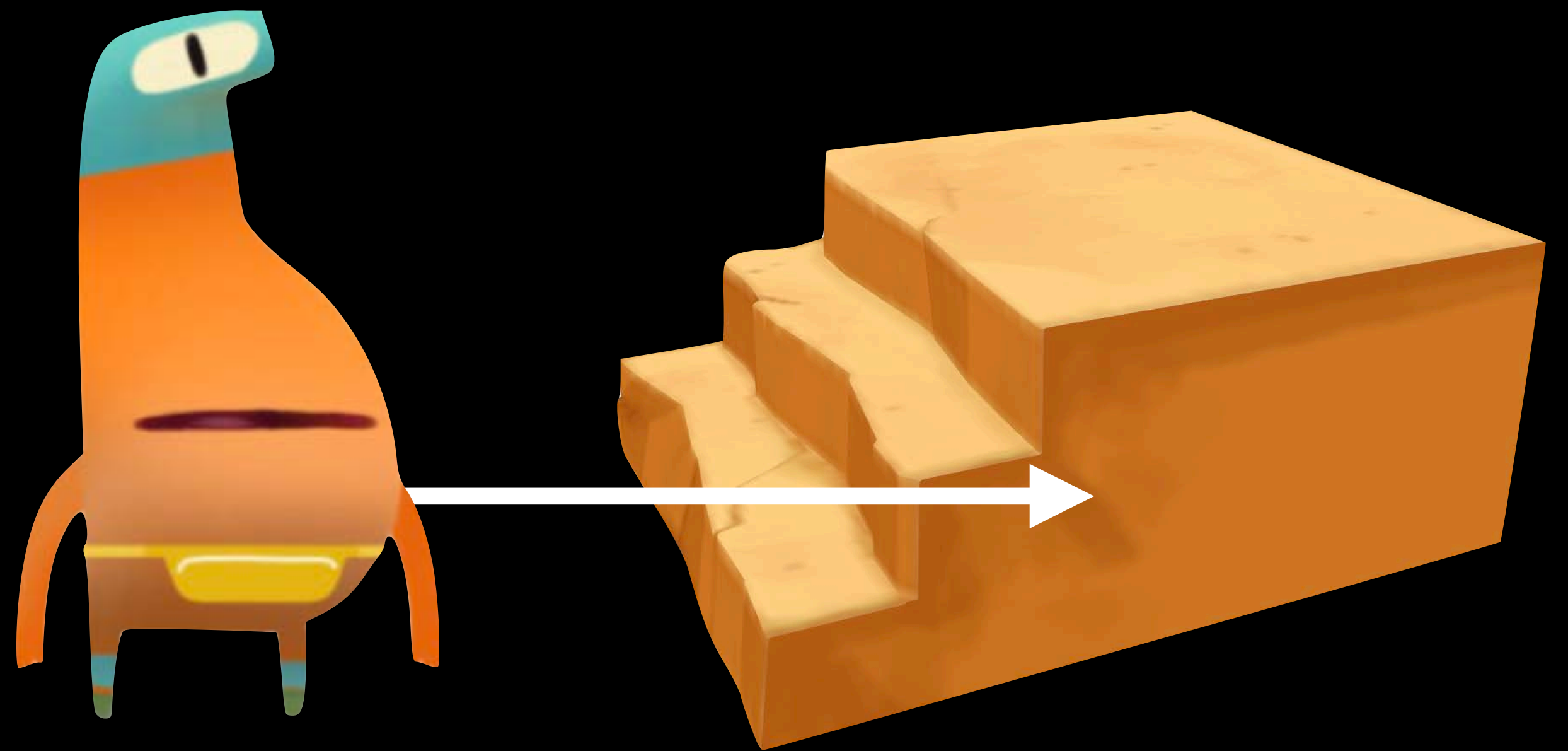
# Animating Up

Use a ramp



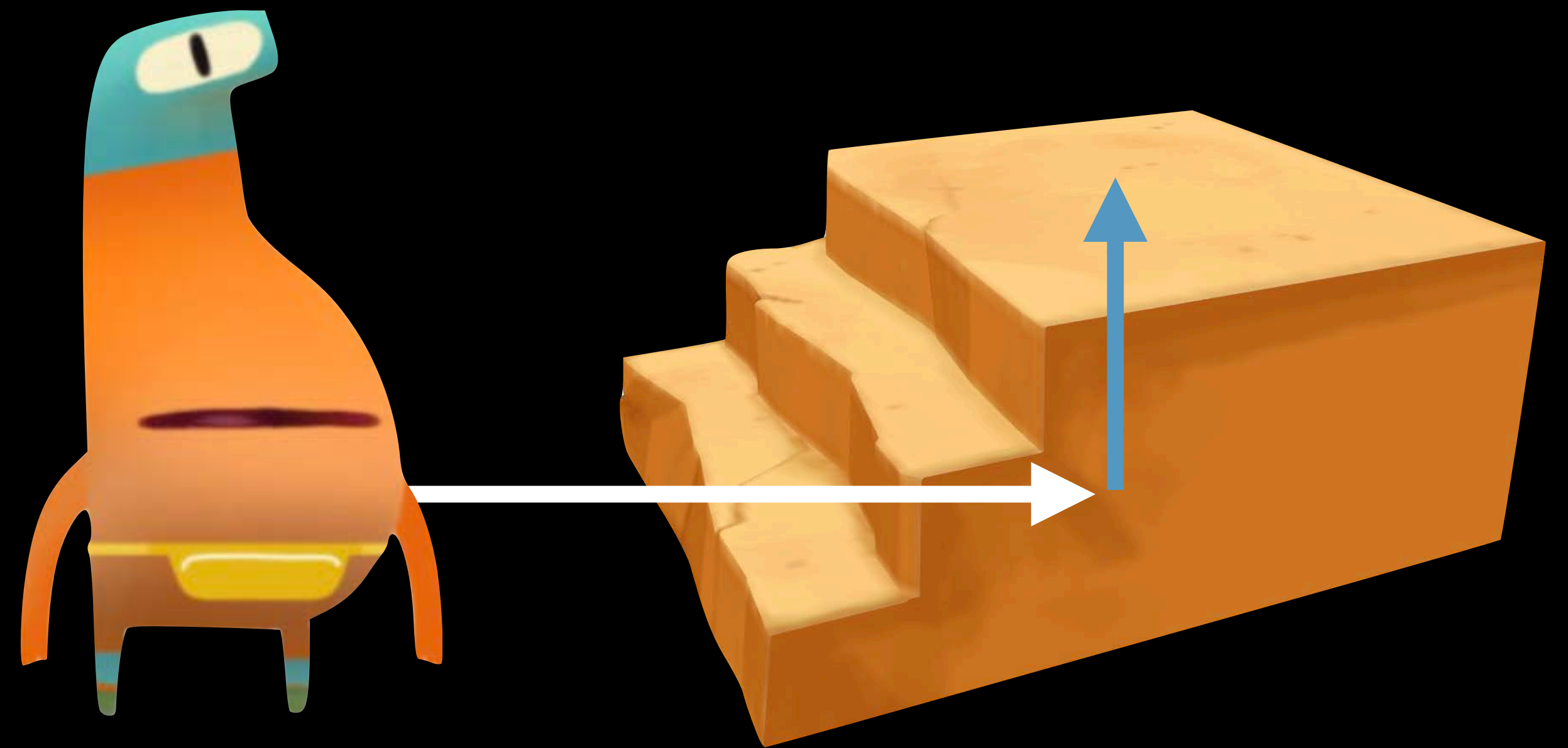
# Animating Up

Use a ramp



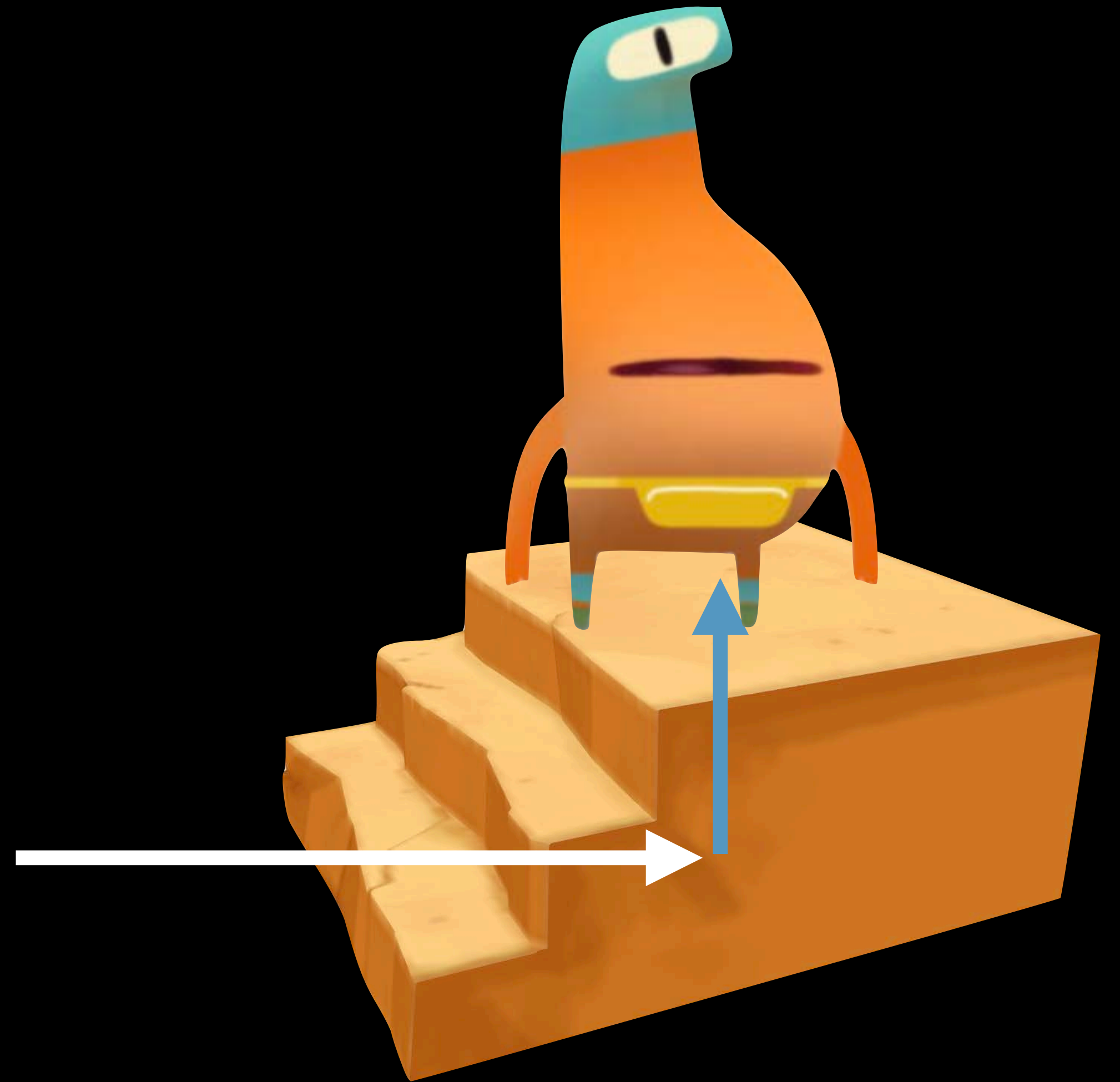
# Animating Up

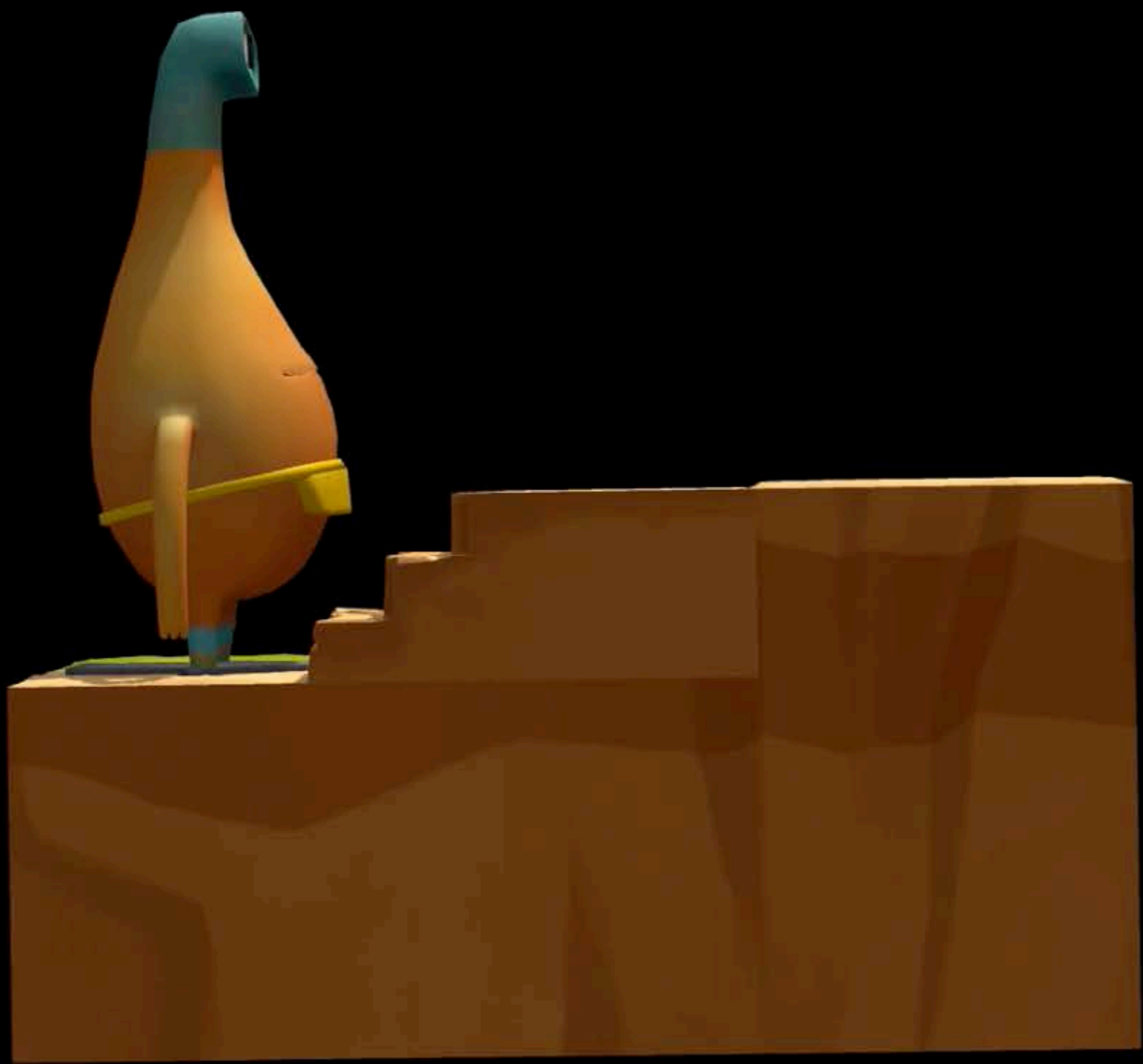
Use a ramp

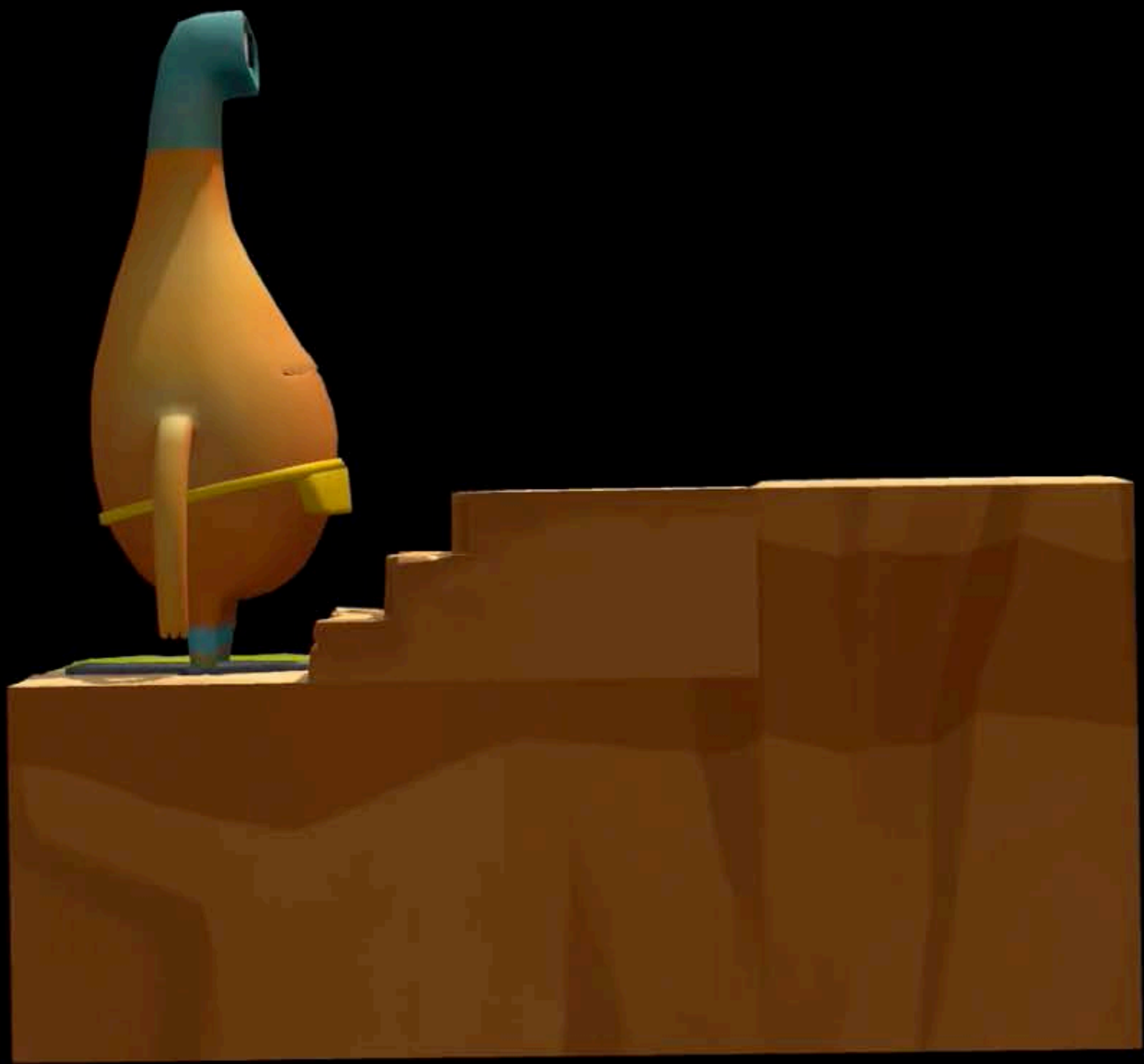


# Animating Up

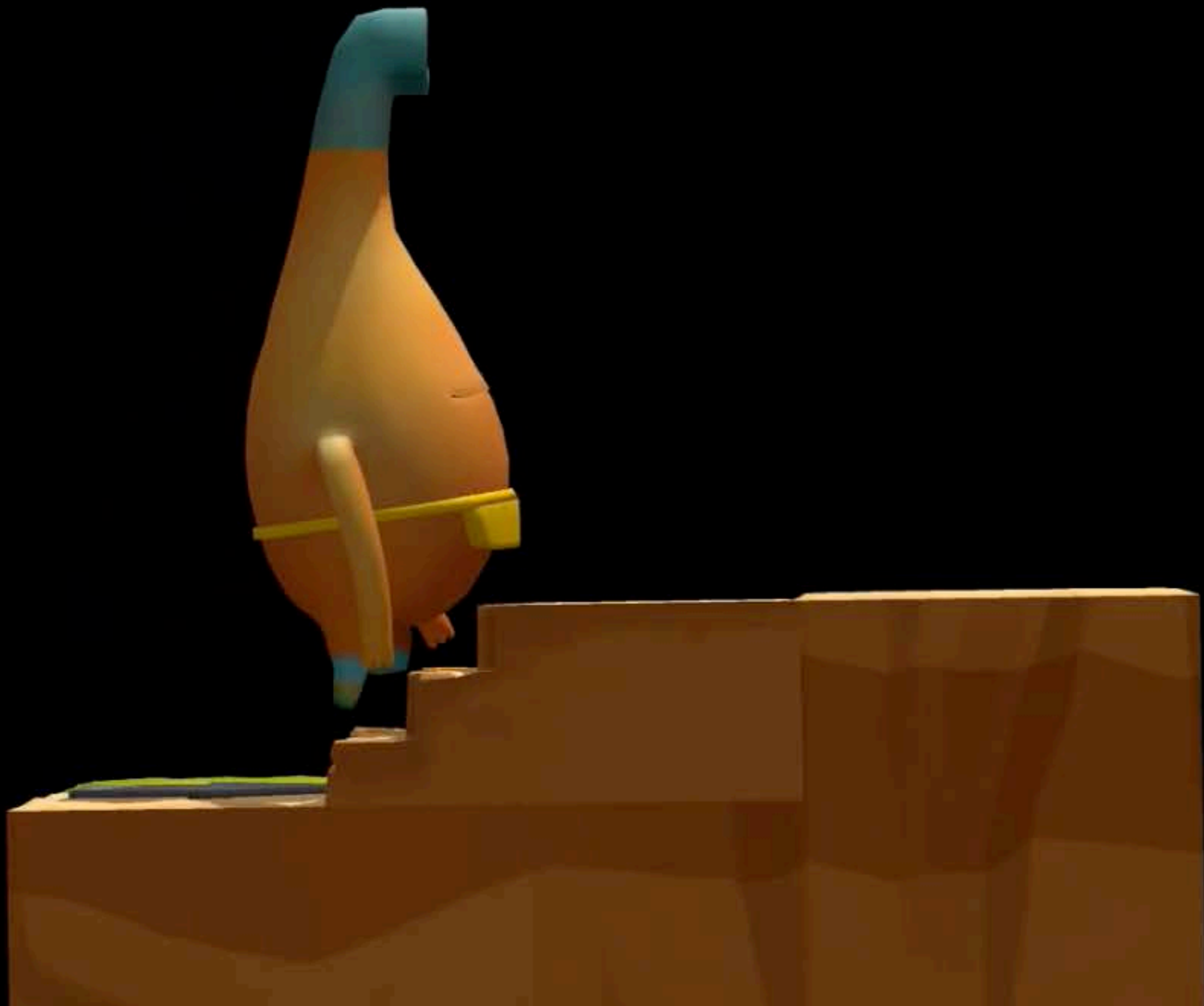
Use a ramp

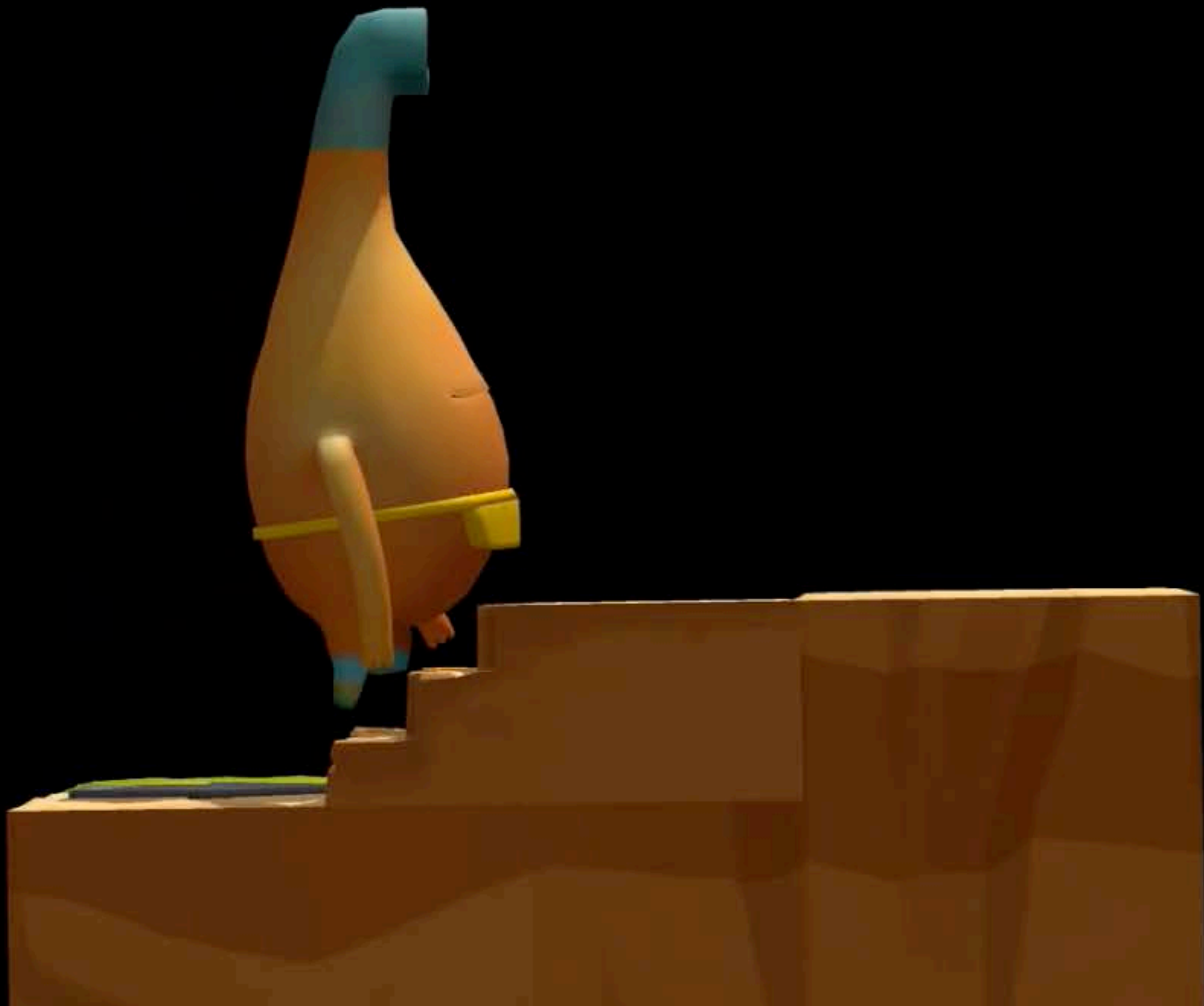






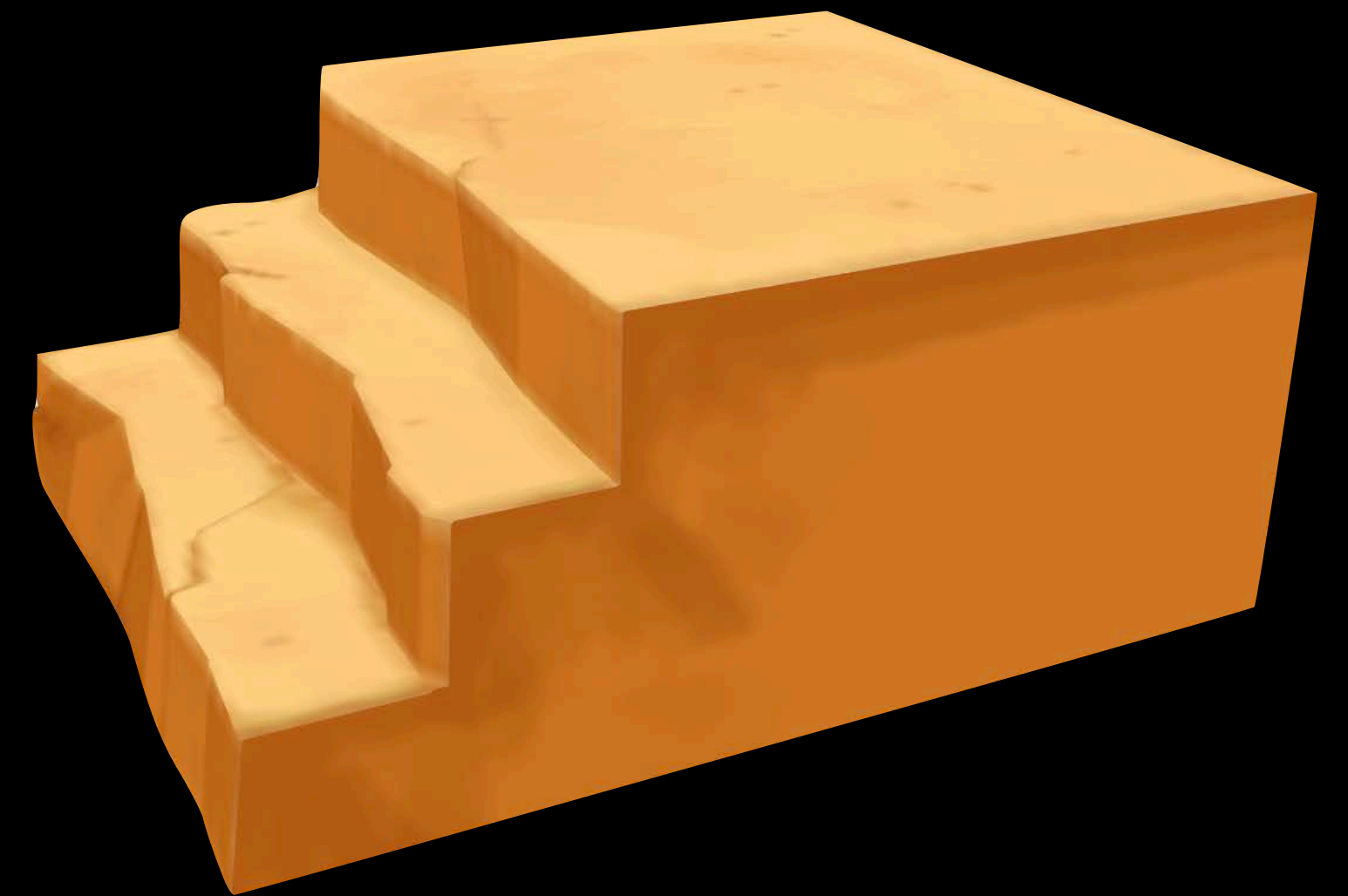






# Animating Up

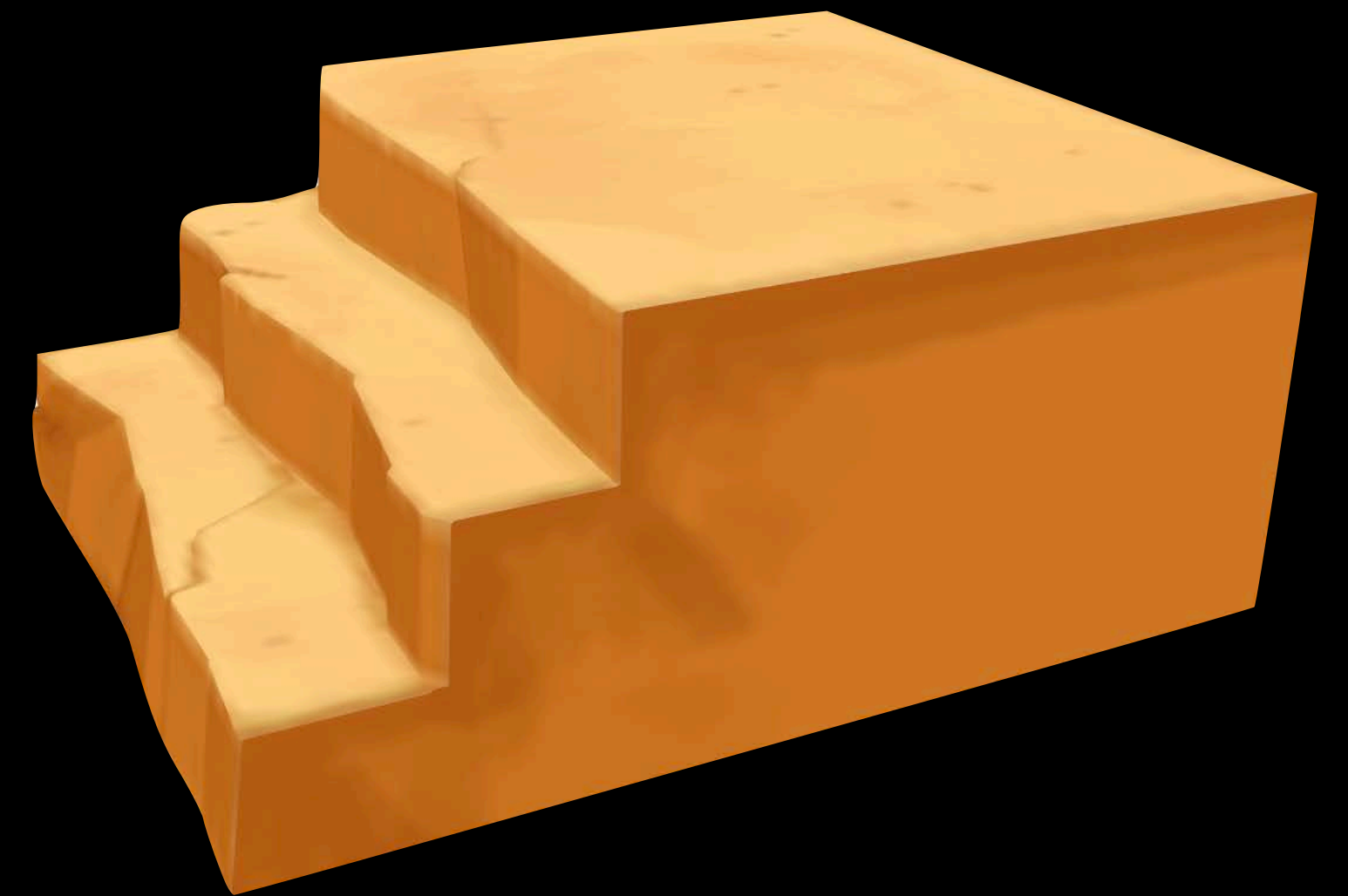
Use a ramp



# Animating Up

Use a ramp

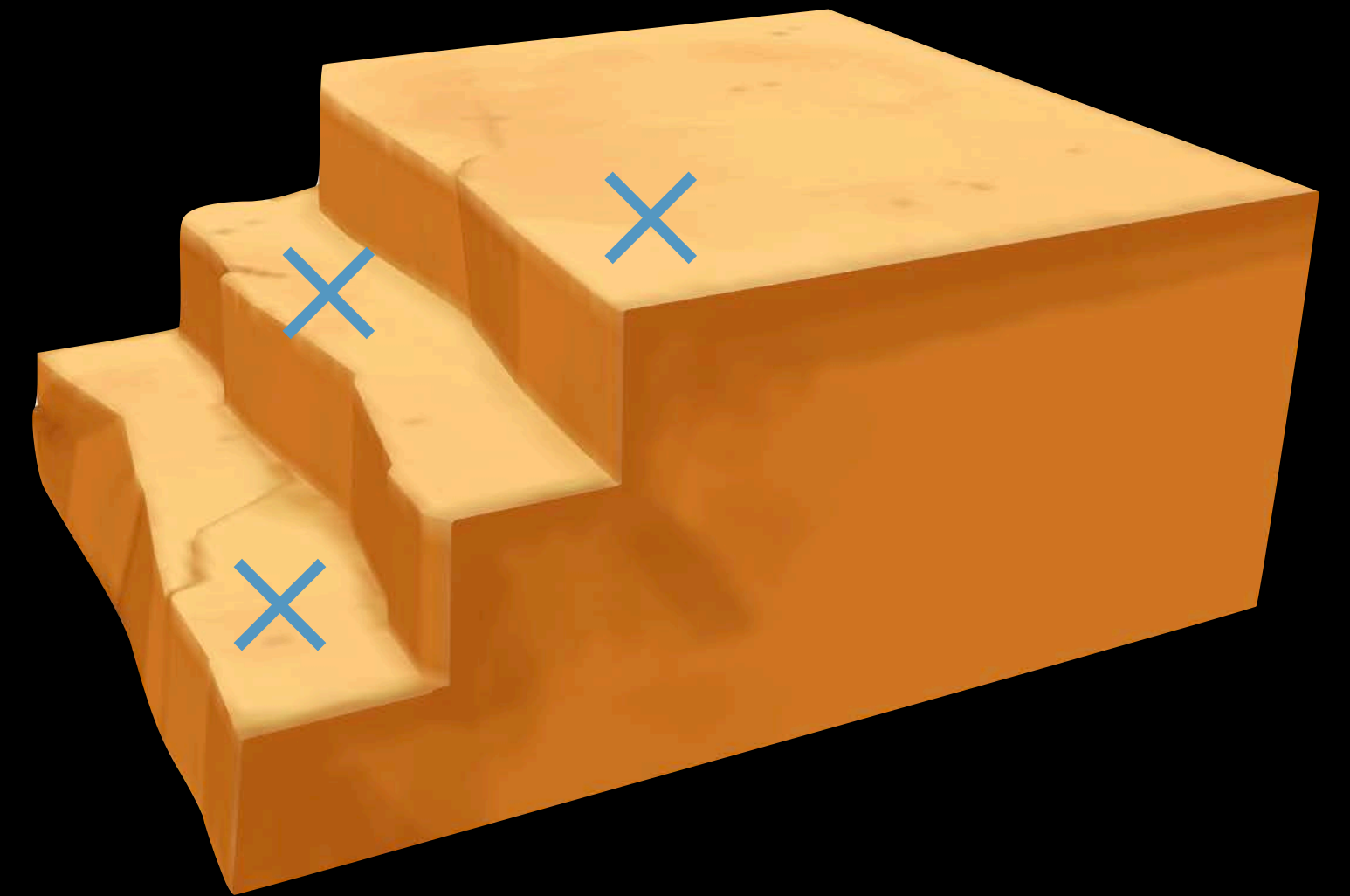
Inverse kinematics



# Animating Up

Use a ramp

Inverse kinematics



# Animating Up

Use a ramp

Inverse kinematics

Displace Animation

(0, 0, 0)

Node

Geometry



(0, 0, 0)

Node



Geometry



(1, 0, 0)

Node

# Displace Animation

Geometry



(0, 0, 0)

Node

# Displace Animation



(0, 0, 0)

Node

Geometry



# Displace Animation

Geometry



(1, 0, 0)

Node

# Displace Animation

Geometry



(1, 0, 0)

Node

# Displace Animation

**SCNTransaction**









```
//Synchronize animation completion and node position.
```

```
SCNTransaction.begin()
```

```
SCNTransaction.animationDuration = 0.0
```

```
actor.position = newPosition
```

```
// Remove all animations.
```

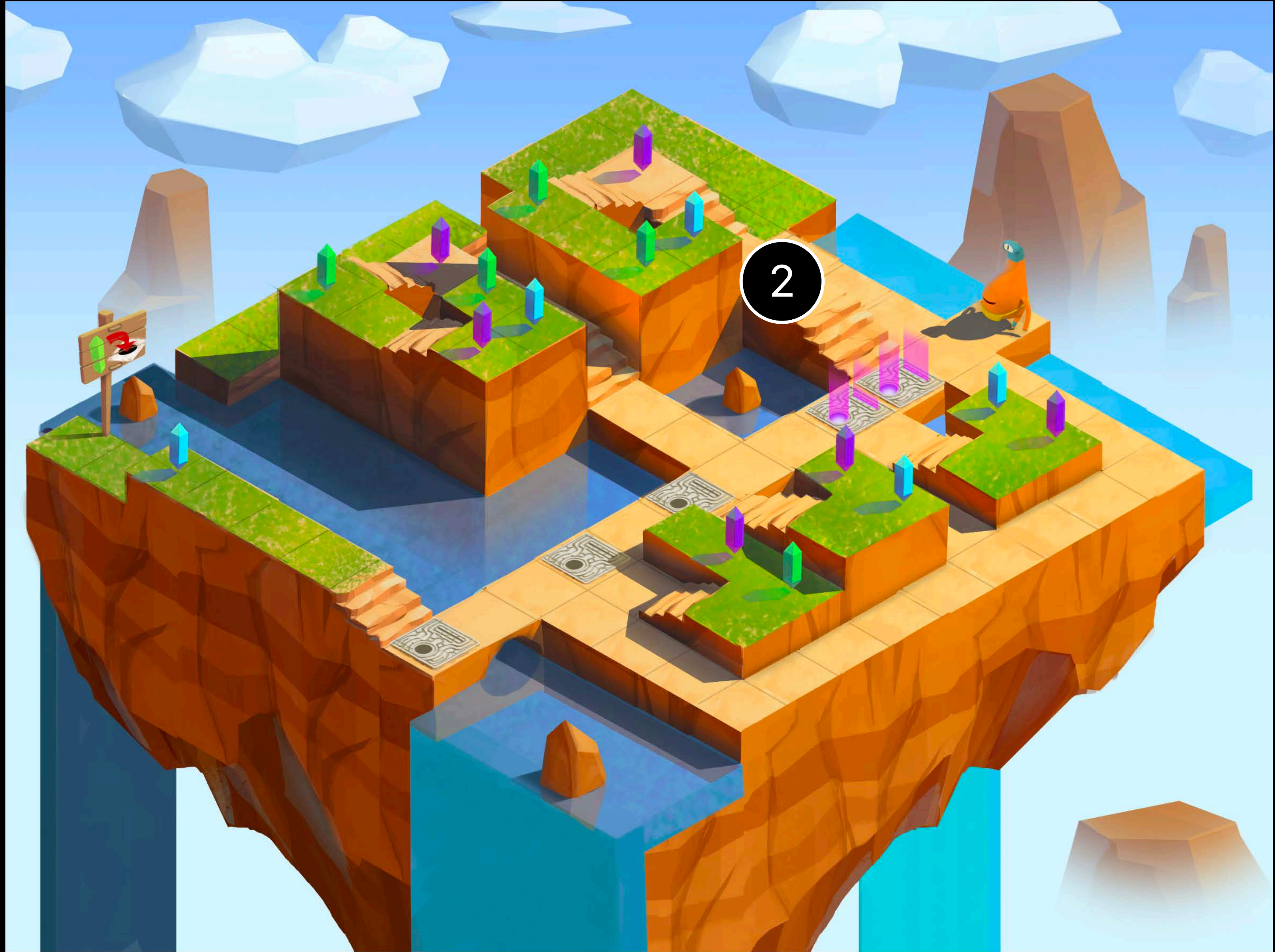
```
for key in node.animationKeys {
```

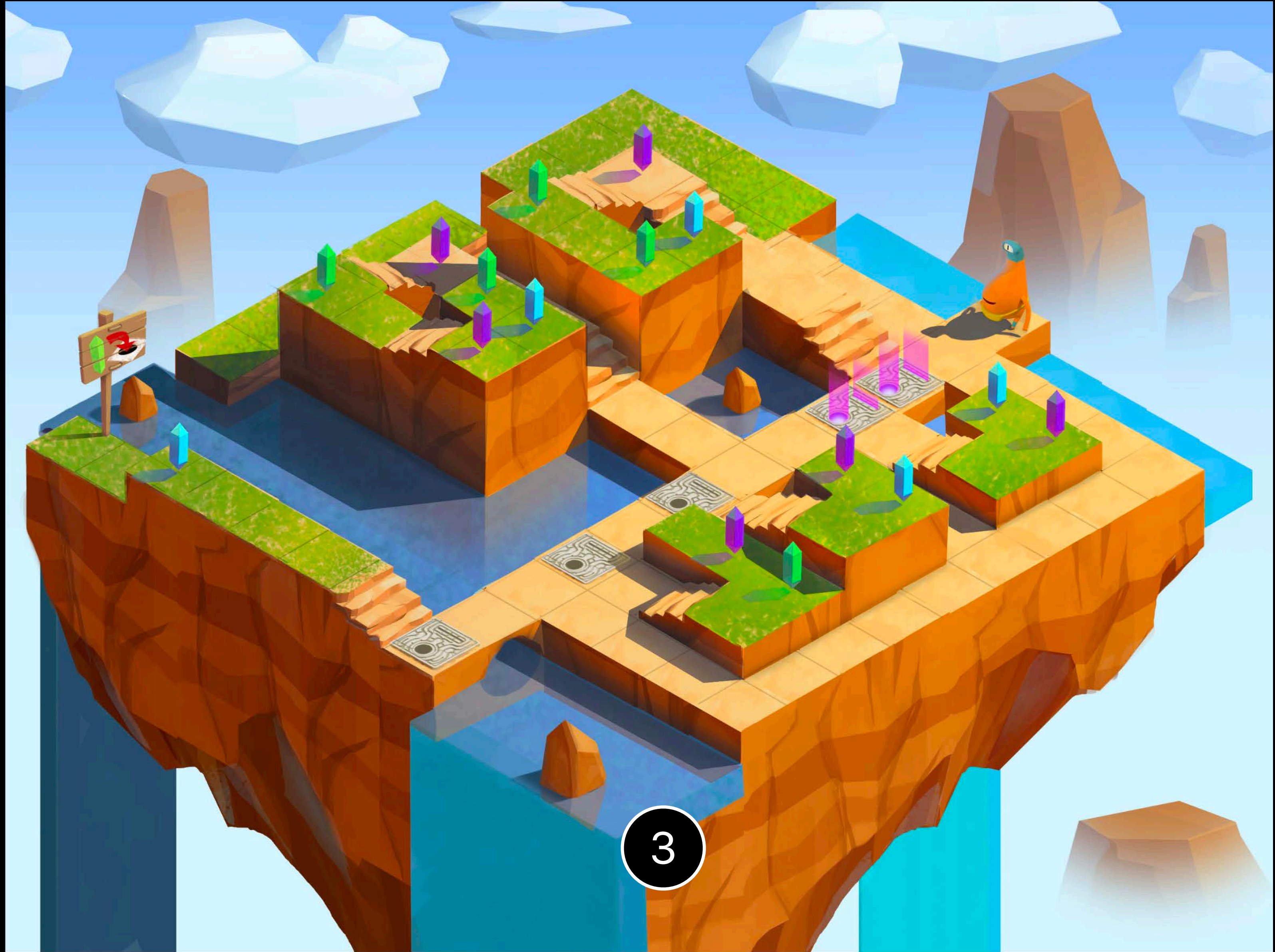
```
    node.removeAnimation(forKey: key)
```

```
}
```

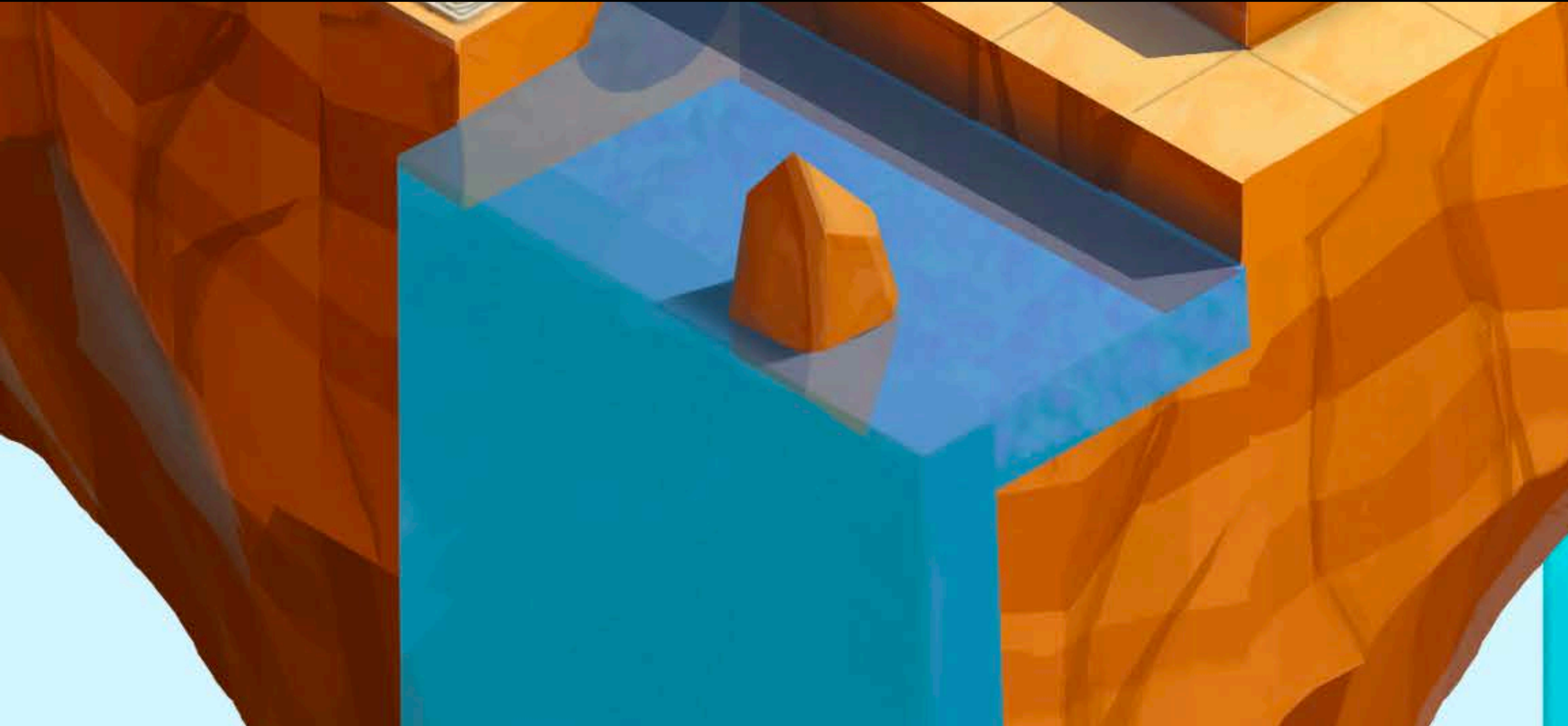
```
SCNTransaction.commit()
```



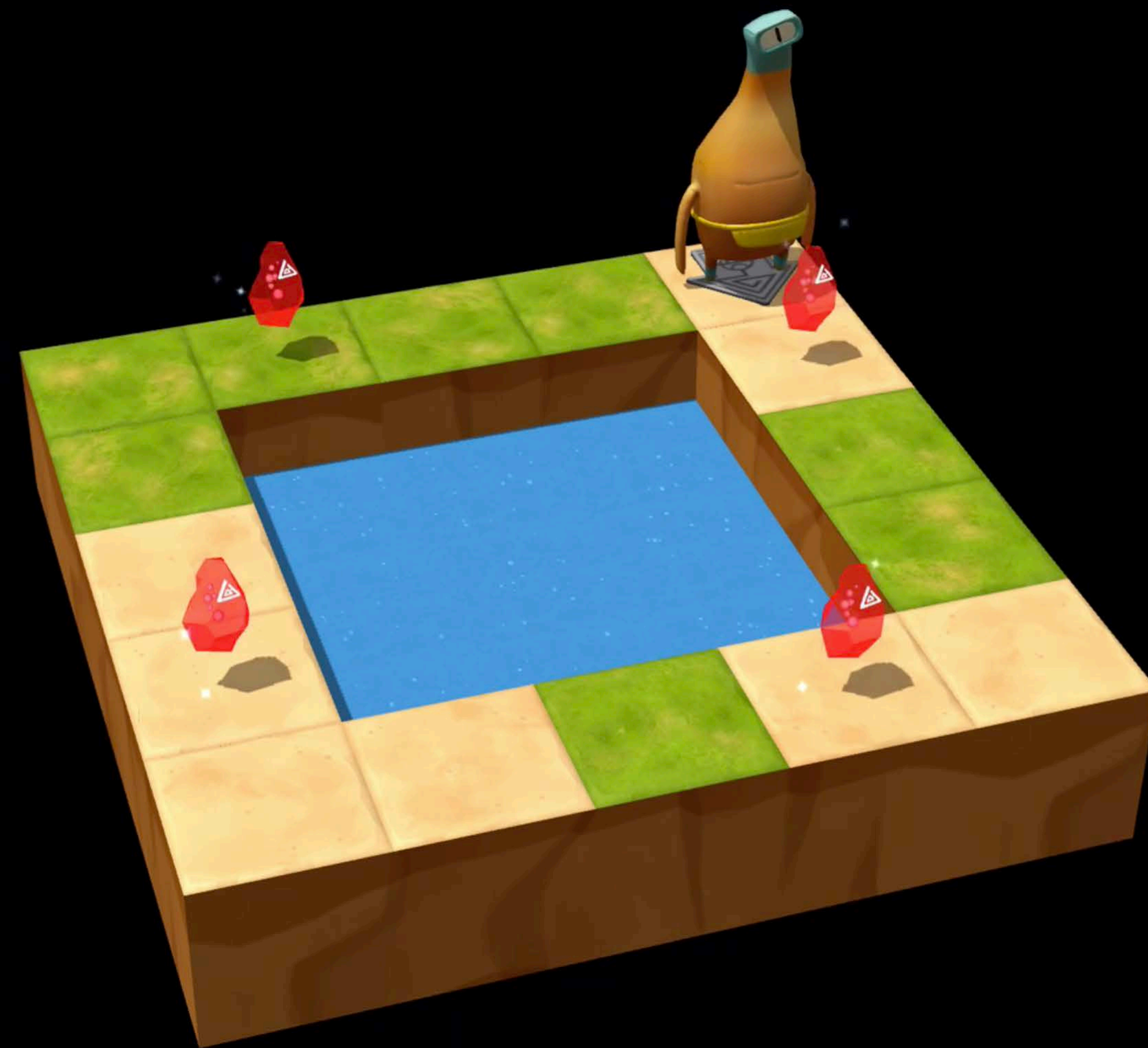




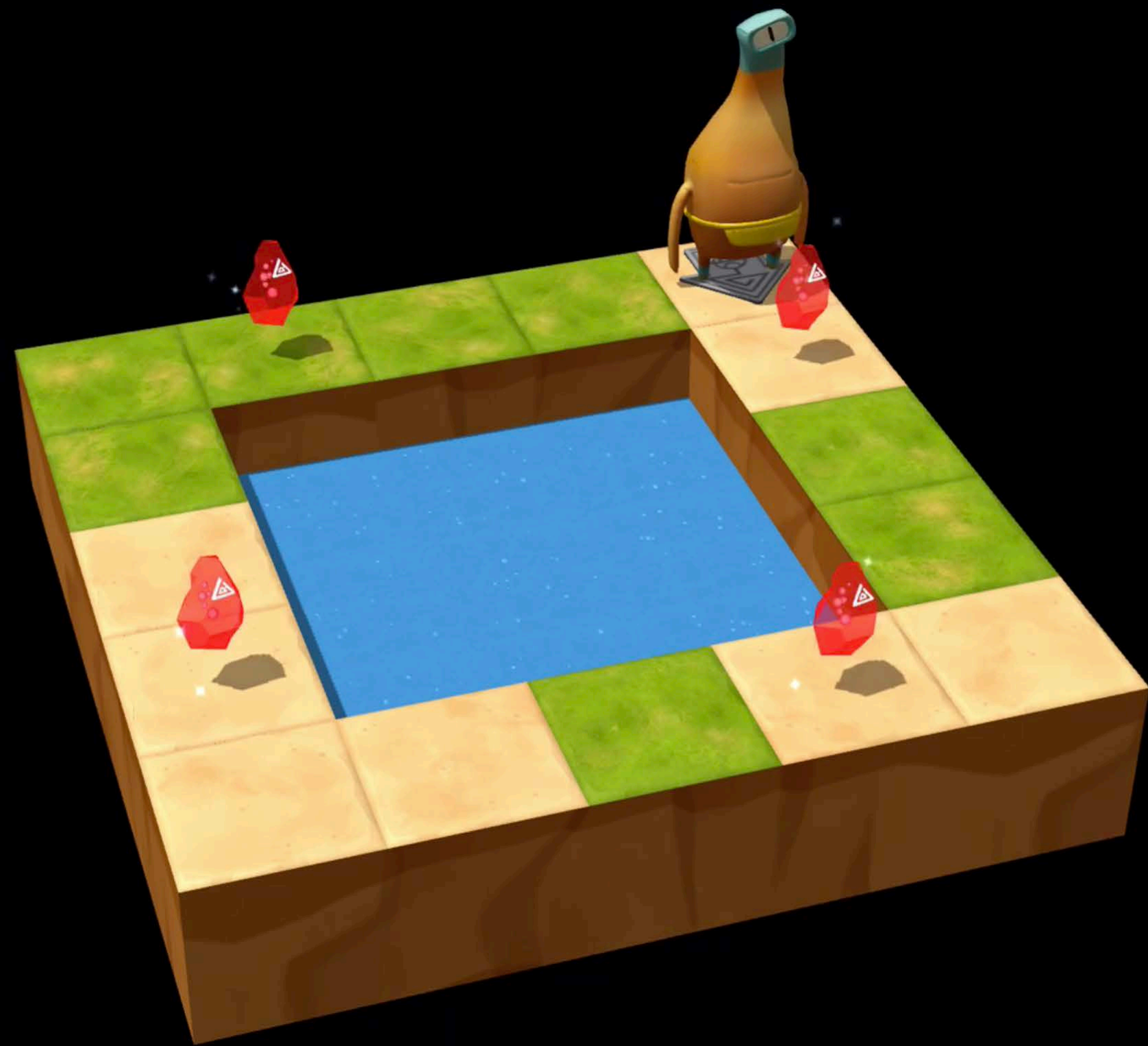
# Dynamic Scenery



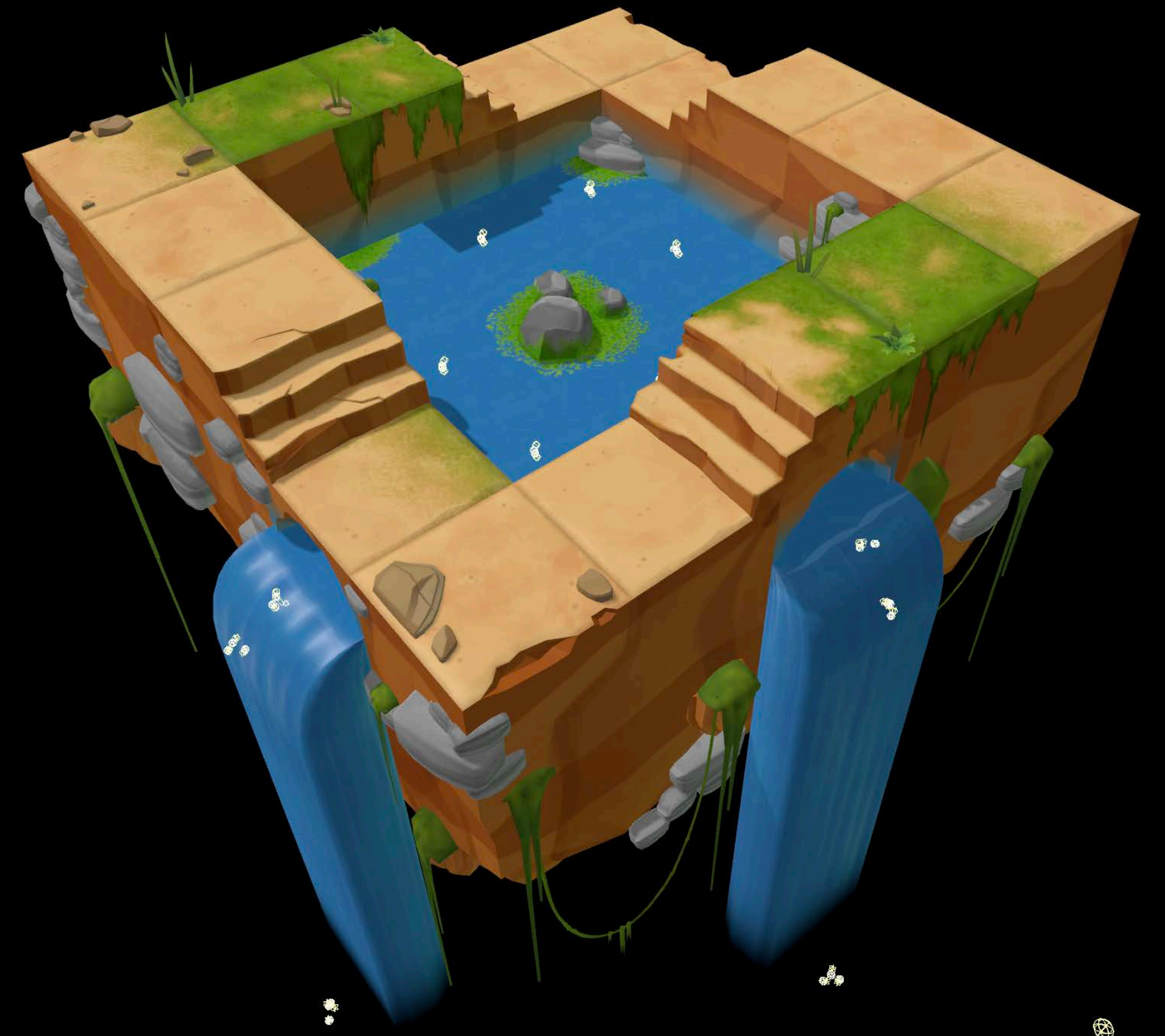
# Original Map



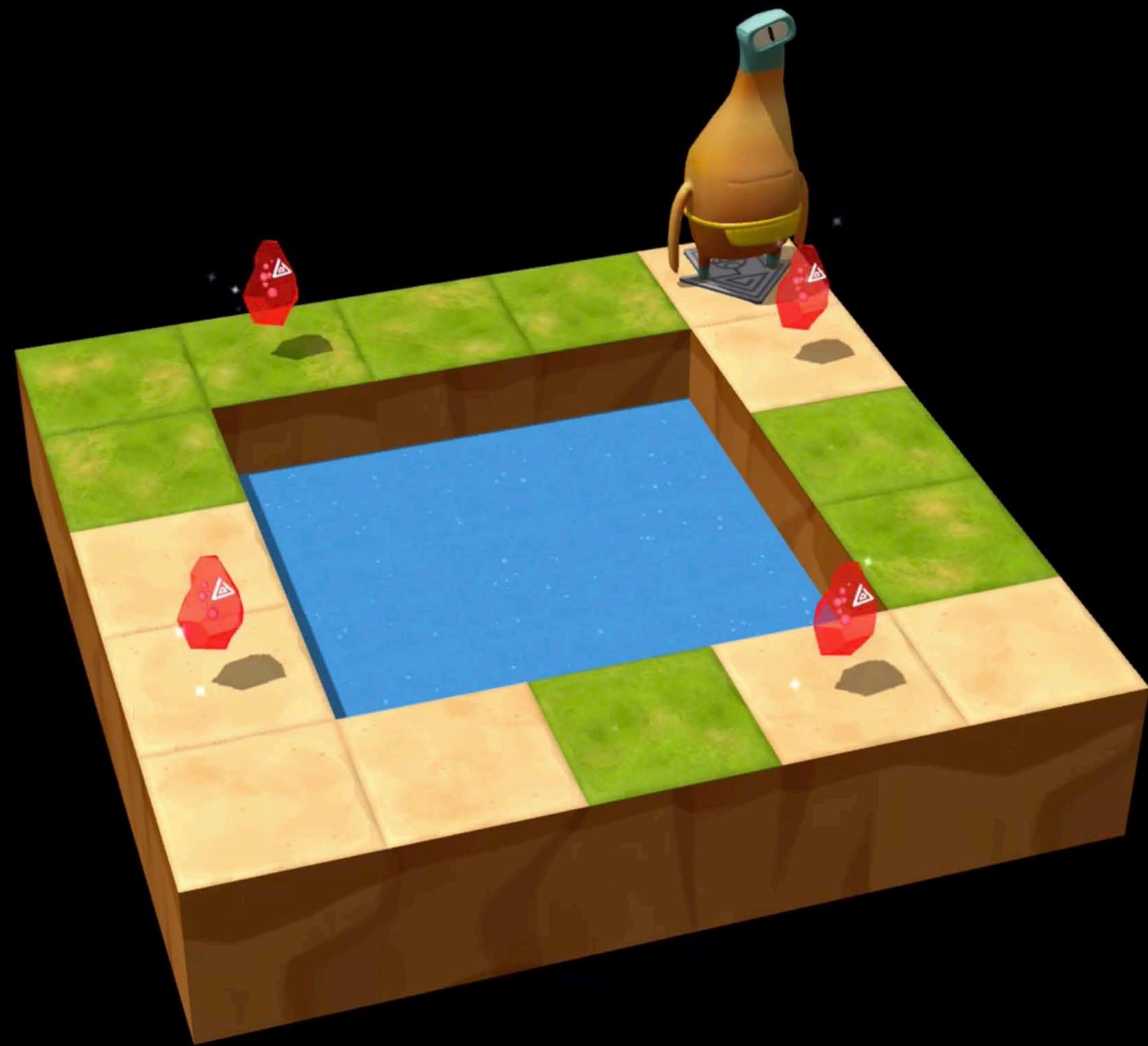
# Original Map



# Designed Map



# Original Map



Save



# Designed Map









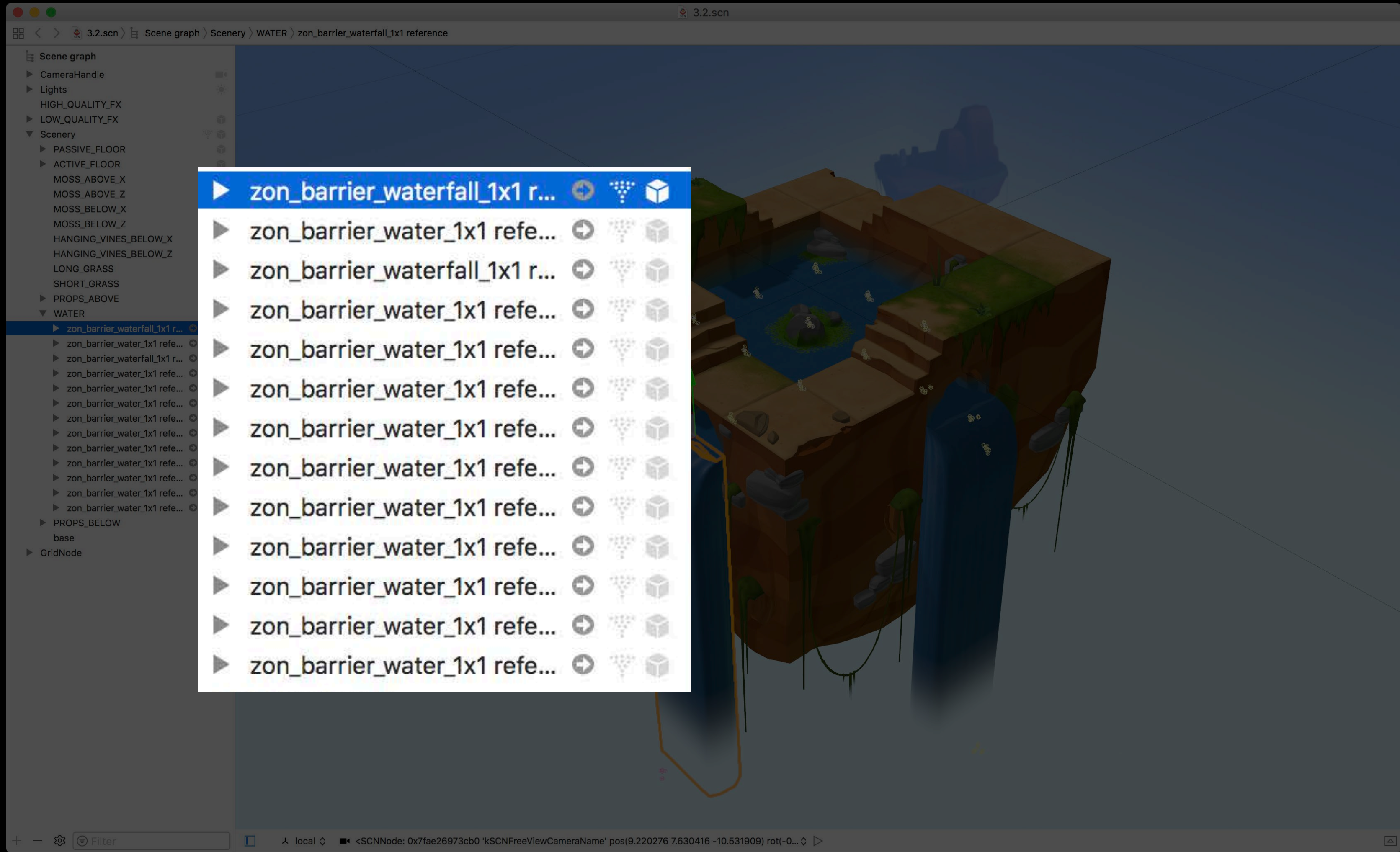
3.2.scn

Scene graph > Scenery > WATER > zon\_barrier\_waterfall\_1x1 reference

- Scene graph
  - CameraHandle
  - Lights
    - HIGH\_QUALITY\_FX
    - LOW\_QUALITY\_FX
  - Scenery
    - PASSIVE\_FLOOR
    - ACTIVE\_FLOOR
    - MOSS\_ABOVE\_X
    - MOSS\_ABOVE\_Z
    - MOSS\_BELOW\_X
    - MOSS\_BELOW\_Z
    - HANGING\_VINES\_BELOW\_X
    - HANGING\_VINES\_BELOW\_Z
    - LONG\_GRASS
    - SHORT\_GRASS
    - PROPS\_ABOVE
    - WATER
      - zon\_barrier\_waterfall\_1x1 r...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_waterfall\_1x1 r...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - zon\_barrier\_water\_1x1 refe...
      - PROPS\_BELOW
      - base
  - GridNode

Filter

local <> <SCNNode: 0x7fae26973cb0 'kSCNFreeViewCameraName' pos(9.220276 7.630416 -10.531909) rot(-0... >



▶ zon\_barrier\_waterfall\_1x1 r... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_waterfall\_1x1 r... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

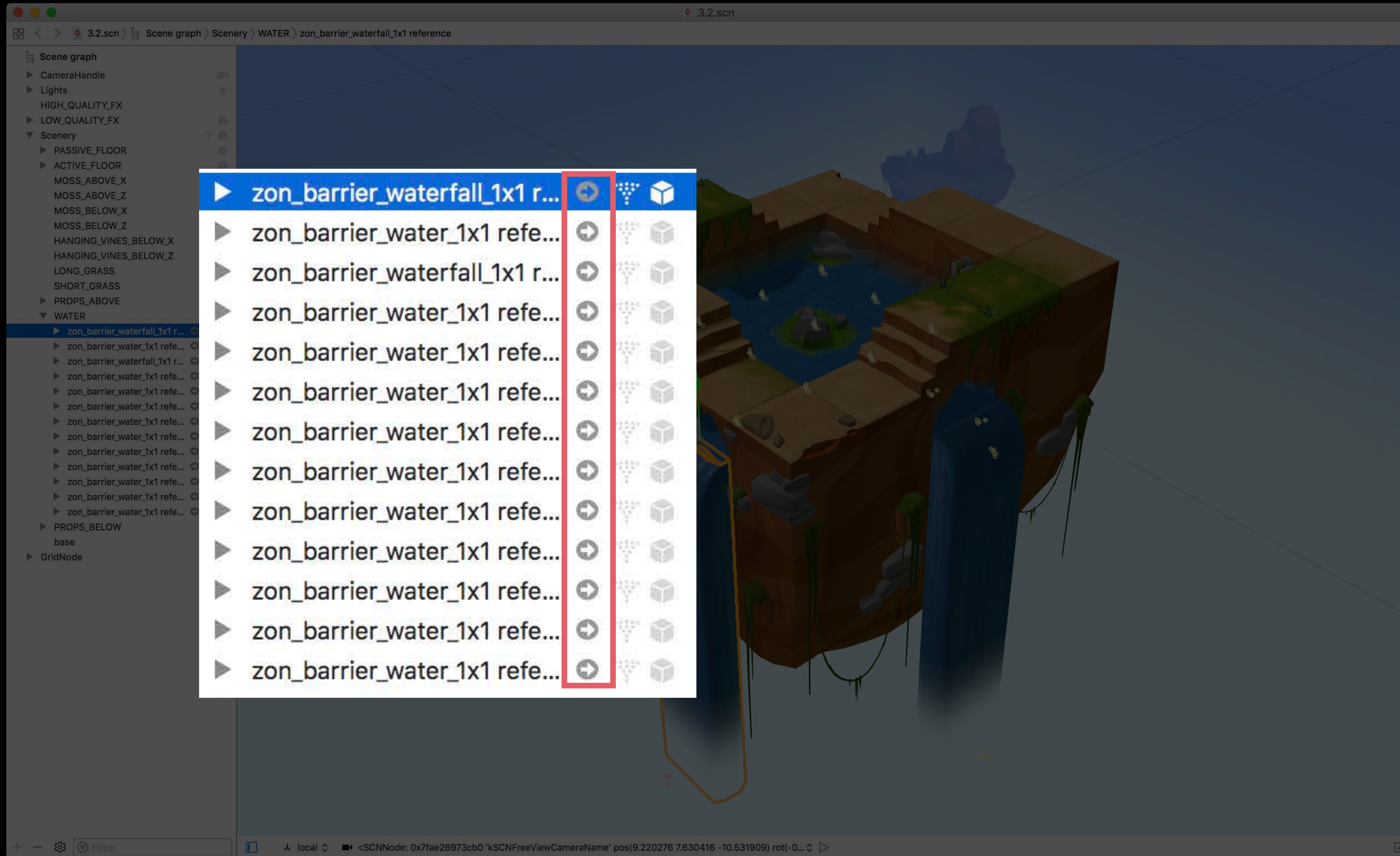
▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

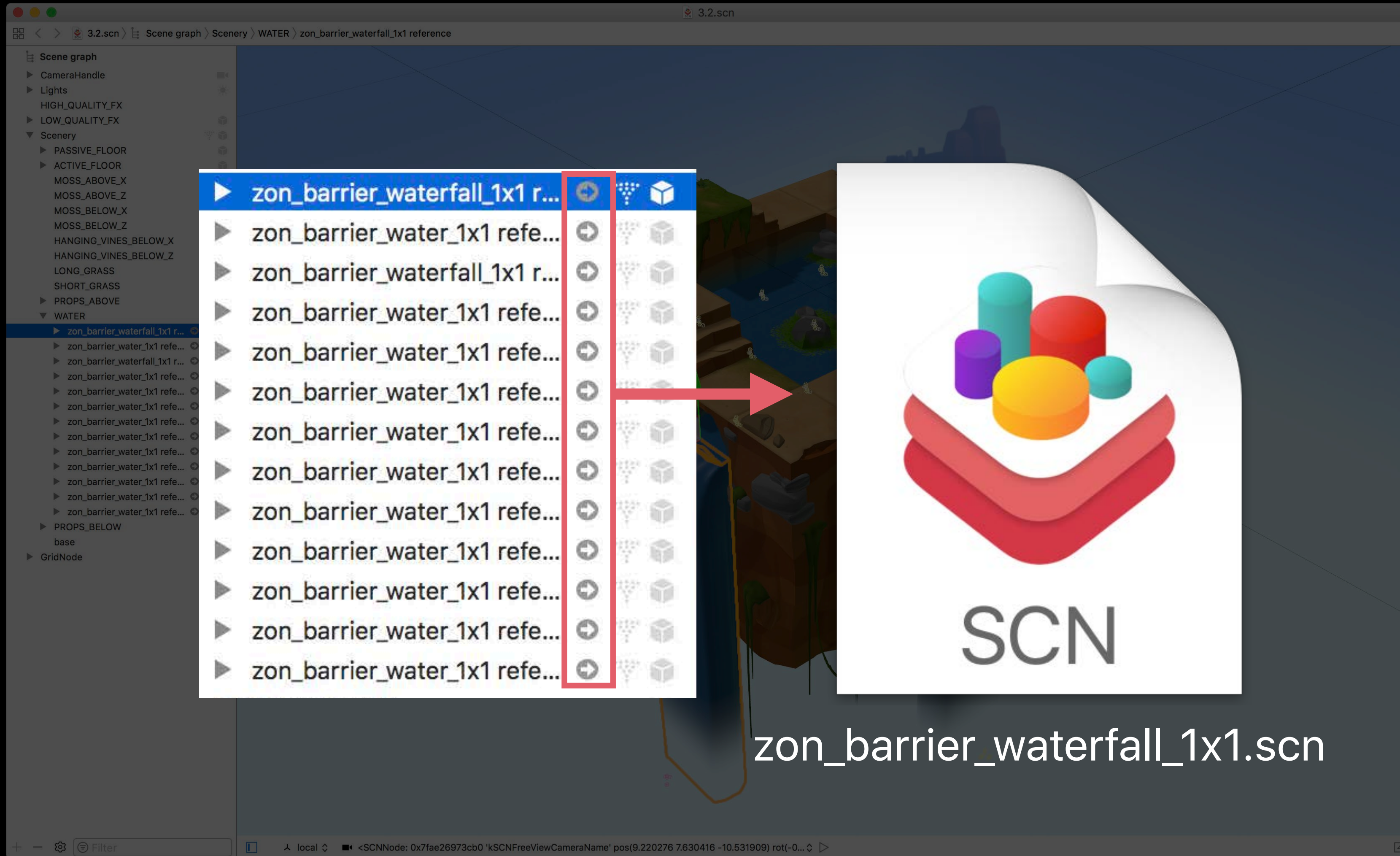
▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

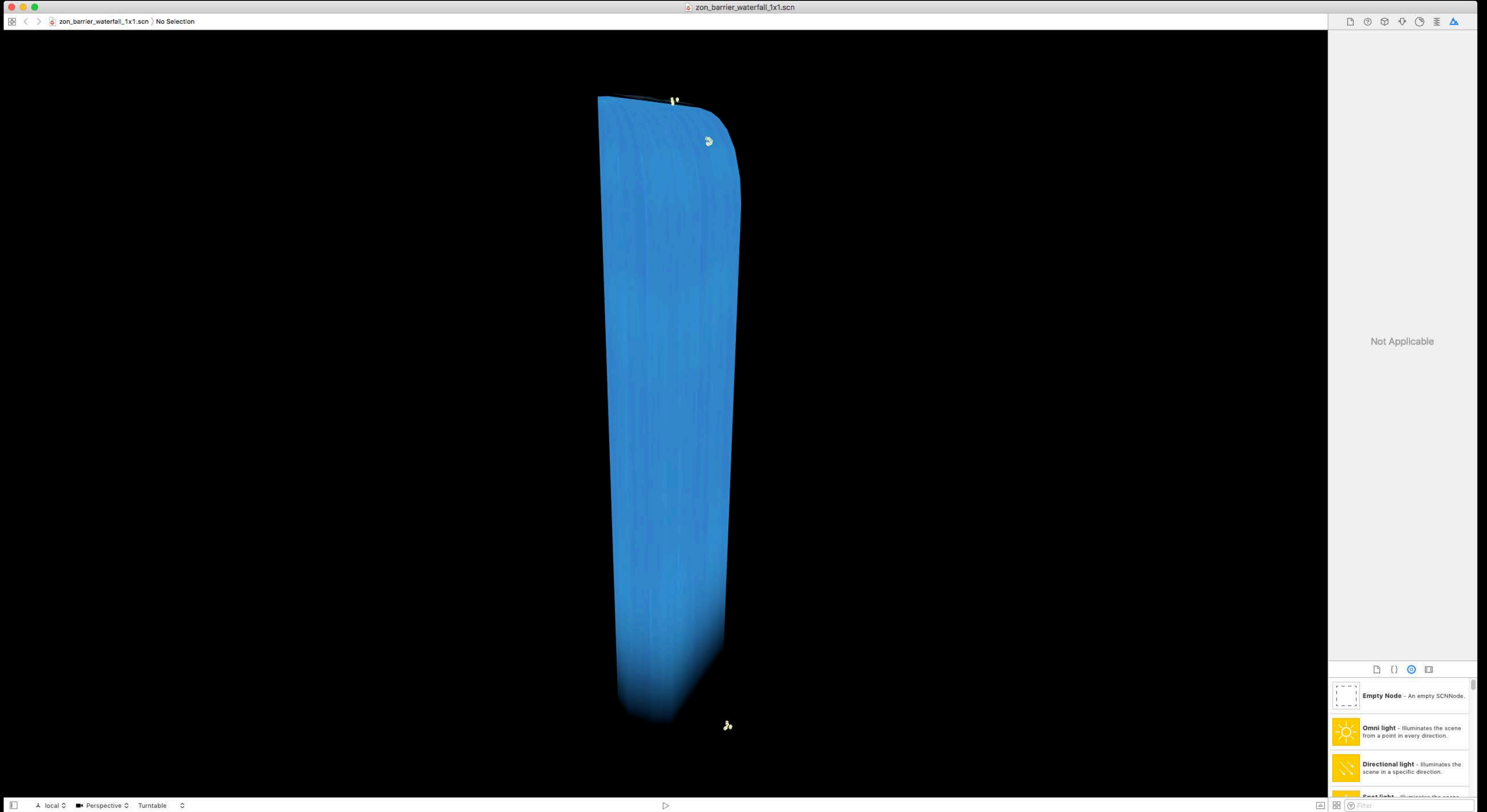
▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦

▶ zon\_barrier\_water\_1x1 refe... ↻ 🍷 📦







zon\_barrier\_waterfall\_1x1.scn > No Selection


zon\_barrier\_waterfall\_1x1.scn



Not Applicable



 **Empty Node** - An empty SCNNode.

 **Omni light** - Illuminates the scene from a point in every direction.

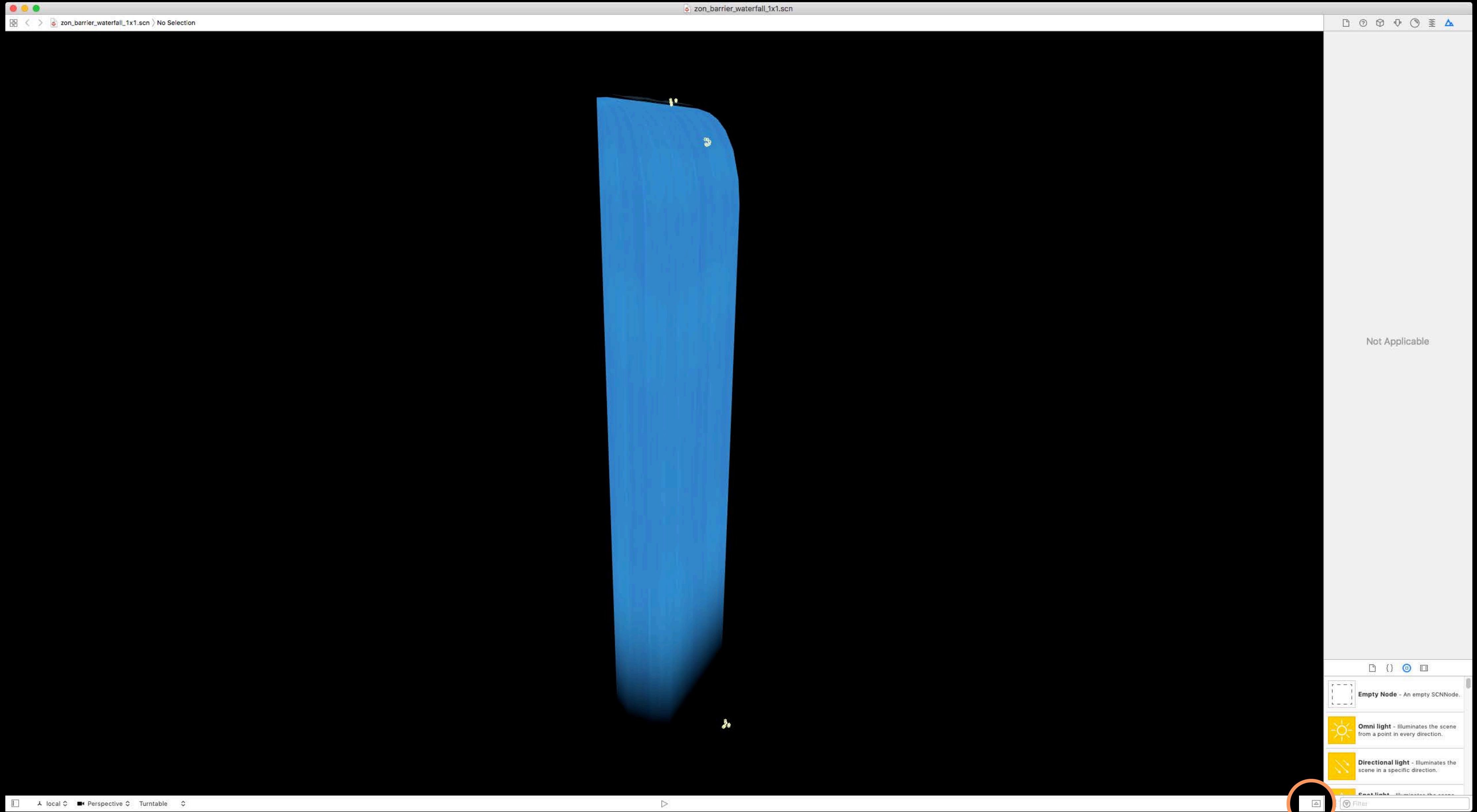
 **Directional light** - Illuminates the scene in a specific direction.

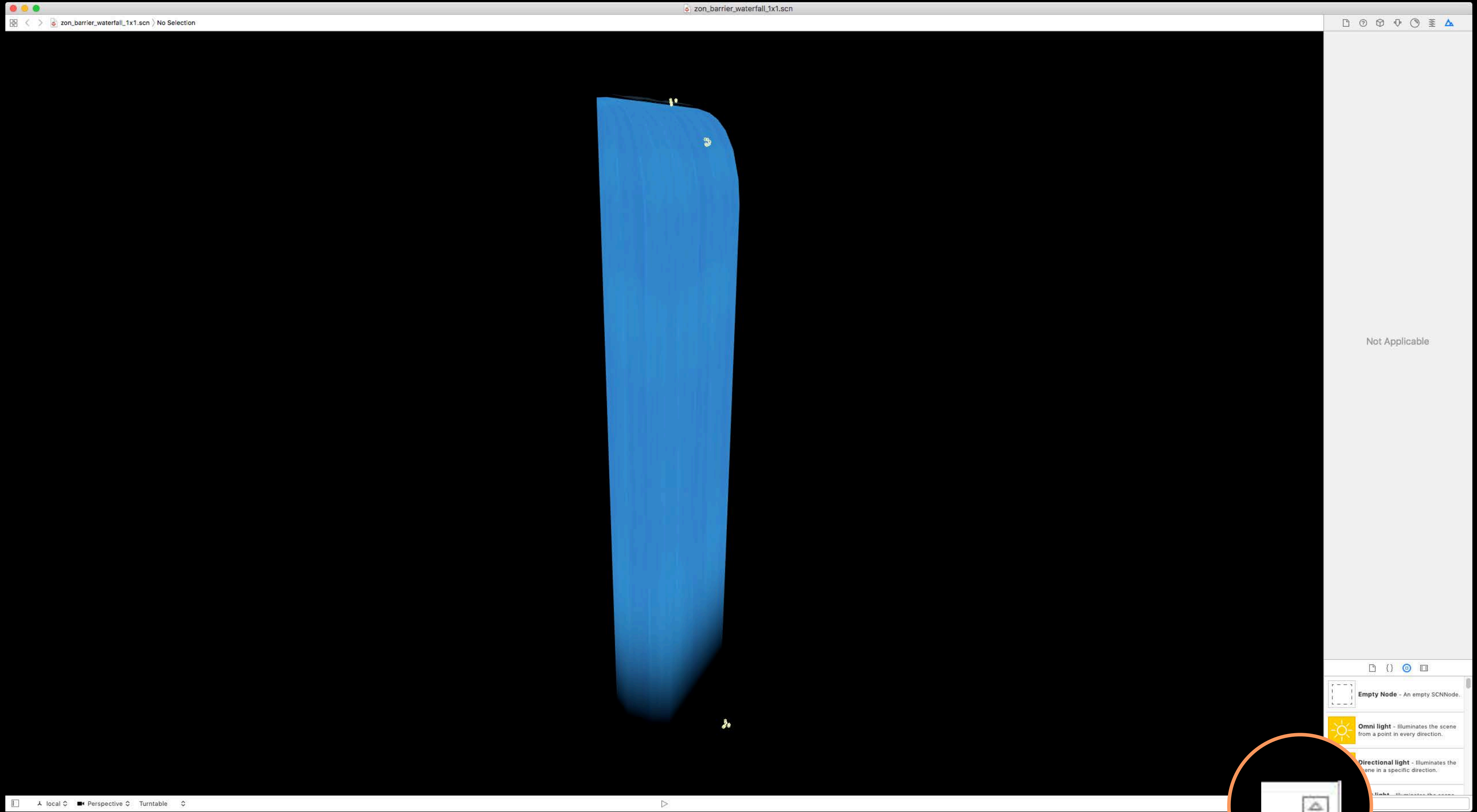
 **Spot light** - Illuminates the scene from a point in a specific direction.

local Perspective Turntable



Filter





Not Applicable

Empty Node - An empty SCNNode.

Omni light - Illuminates the scene from a point in every direction.


Directional light - Illuminates the scene in a specific direction.





zon\_barrier\_waterfall\_1x1.scn

Scene graph > zon\_barrier\_waterfall\_1x1\_GRP > zon\_barrier\_waterfall\_1x1\_Down\_GRP > zon\_barrier\_waterfallTop\_1x1\_GEO



Scene

Background

Lighting Environment

Light probes

Fog

Physics Settings

Timing

Geometry

Embedded Textures

local Perspective Turntable

Shaders Actions

zon\_barrier\_waterfallTop\_1x1\_GEOShape

- Geometry modifier
- Surface modifier
- Lighting model modifier
- Fragment modifier
- waterFall\_SHD
  - Geometry modifier
  - Surface modifier
  - Lighting model modifier
  - Fragment modifier

Geometry modifier

```
1 /*
2 Access: ReadWrite
3 Stages: Vertex shader only
4 */
5 struct SCNShaderGeometry {
6     vec4 position;
7     vec3 normal;
8     vec4 tangent;
9     vec4 color;
10    vec2 texcoords[kSCNTexCoordCount];
11 } _geometry;
12
13 // modifier
14 _geometry.texcoords[0].y -= u_time*0.5;
15
16
17
```

Empty Node - An empty SCNNode.

Omni light - Illuminates the scene from a point in every direction.

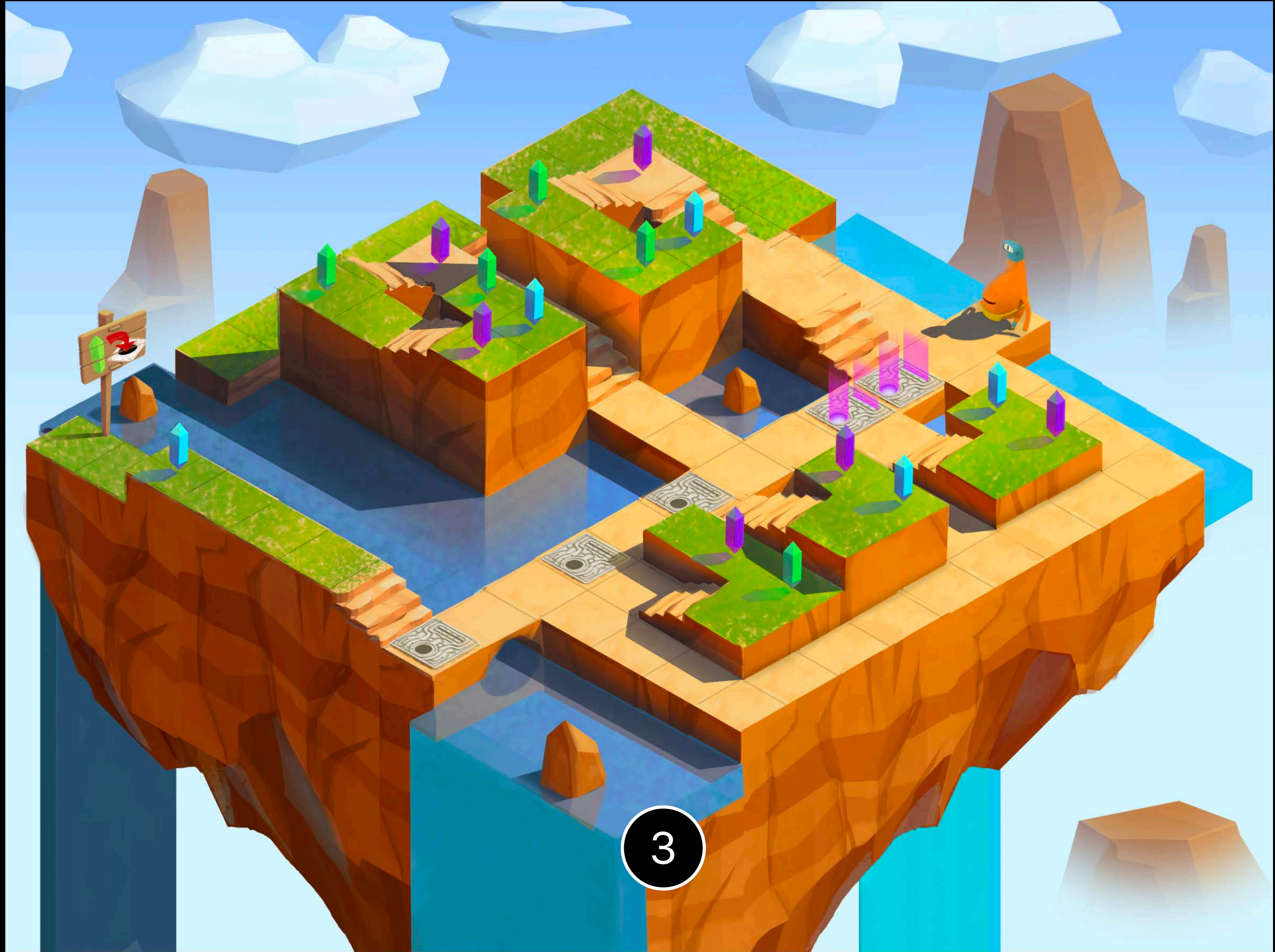
Directional light - Illuminates the scene in a specific direction.

Spot light - Illuminates the scene from a point in a specific direction.

Filters



MacBook Pro





# VoiceOver Support

# VoiceOver Support

Great non-visual experience

# VoiceOver Support

Great non-visual experience

Enable all interactions

Support custom actions

Describe important elements

# VoiceOver Support

Great non-visual experience

Enable all interactions

Support custom actions

Describe important elements





```
// Making accessible coordinates

class CoordinateAccessibilityElement: UIAccessibilityElement {
    var world: GridWorld
    var coordinate: Coordinate

    override var accessibilityLabel: String? {
        get {
            return world.speakableContents(of: coordinate)
        }
        ...
    }
}
```

```
// Setup accessibility elements.

// 1. Create accessibility element.
let accessibilityElement = CoordinateAccessibilityElement(coordinate: coordinate)

// 2. Find coordinates in view's coordinate space.
let upperLeft = scnView.projectPoint(SCNVector3(...))
accessibilityElement.frame = ...

// 3. Add element to the view.
view.accessibilityElements!.append(accessibilityElement)
```

```
// Setup accessibility elements.
```

```
// 1. Create accessibility element.
```

```
let accessibilityElement = CoordinateAccessibilityElement(coordinate: coordinate)
```

```
// 2. Find coordinates in view's coordinate space.
```

```
let upperLeft = scnView.projectPoint(SCNVector3(...))
```

```
accessibilityElement.frame = ...
```

```
// 3. Add element to the view.
```

```
view.accessibilityElements!.append(accessibilityElement)
```

```
// Setup accessibility elements.  
  
// 1. Create accessibility element.  
let accessibilityElement = CoordinateAccessibilityElement(coordinate: coordinate)  
  
// 2. Find coordinates in view's coordinate space.  
let upperLeft = scnView.projectPoint(SCNVector3(...))  
accessibilityElement.frame = ...  
  
// 3. Add element to the view.  
view.accessibilityElements!.append(accessibilityElement)
```

```
// Setup accessibility elements.

// 1. Create accessibility element.
let accessibilityElement = CoordinateAccessibilityElement(coordinate: coordinate)

// 2. Find coordinates in view's coordinate space.
let upperLeft = scnView.projectPoint(SCNVector3(...))
accessibilityElement.frame = ...

// 3. Add element to the view.
view.accessibilityElements!.append(accessibilityElement)
```

```
// Setup accessibility elements.

// 1. Create accessibility element.
let accessibilityElement = CoordinateAccessibilityElement(coordinate: coordinate)

// 2. Find coordinates in view's coordinate space.
let upperLeft = scnView.projectPoint(SCNVector3(...))
accessibilityElement.frame = ...

// 3. Add element to the view.
view.accessibilityElements!.append(accessibilityElement)
```

AccessibilityExtensions.swift

# Iterating





# Iterating

Separate data and visuals



# Iterating

Separate data and visuals

Value flexibility



# Iterating

Separate data and visuals

Value flexibility

Audit accessibility support



**Tuning**





< Which Way to Turn? >



0/1 0/6

1x



Mt 29fps

877 130K



Animations	0.9ms	Delegate	
Physics	0.0ms	Rendering	20.4ms
Constraints	0.3ms	WaitDrawable	0.2ms
Particles	1.8ms	2D	

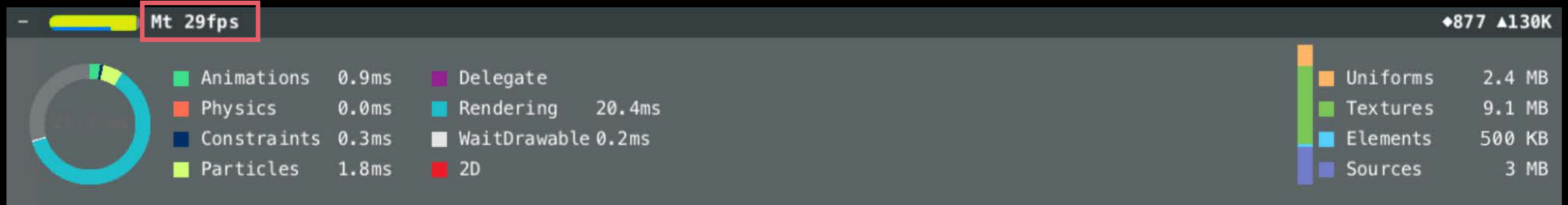
Uniforms	2.4 MB
Textures	9.1 MB
Elements	500 KB
Sources	3 MB

# Rendering Statistics



```
gameView.showsStatistics = true
```

# Rendering Statistics



```
gameView.showsStatistics = true
```

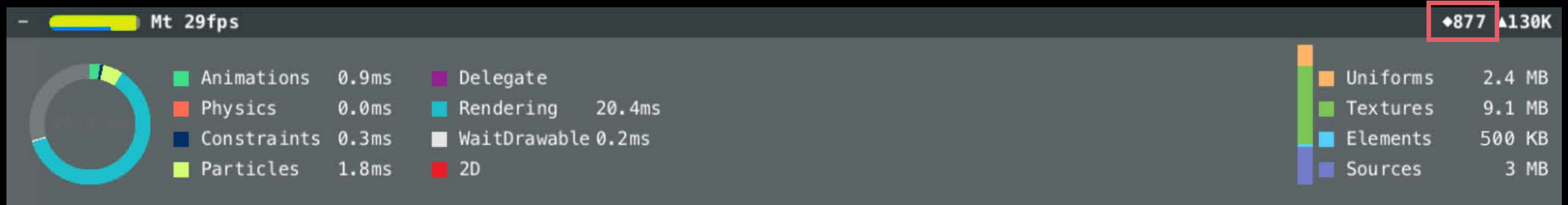


# Rendering Statistics



```
gameView.showsStatistics = true
```

# Rendering Statistics



```
gameView.showsStatistics = true
```

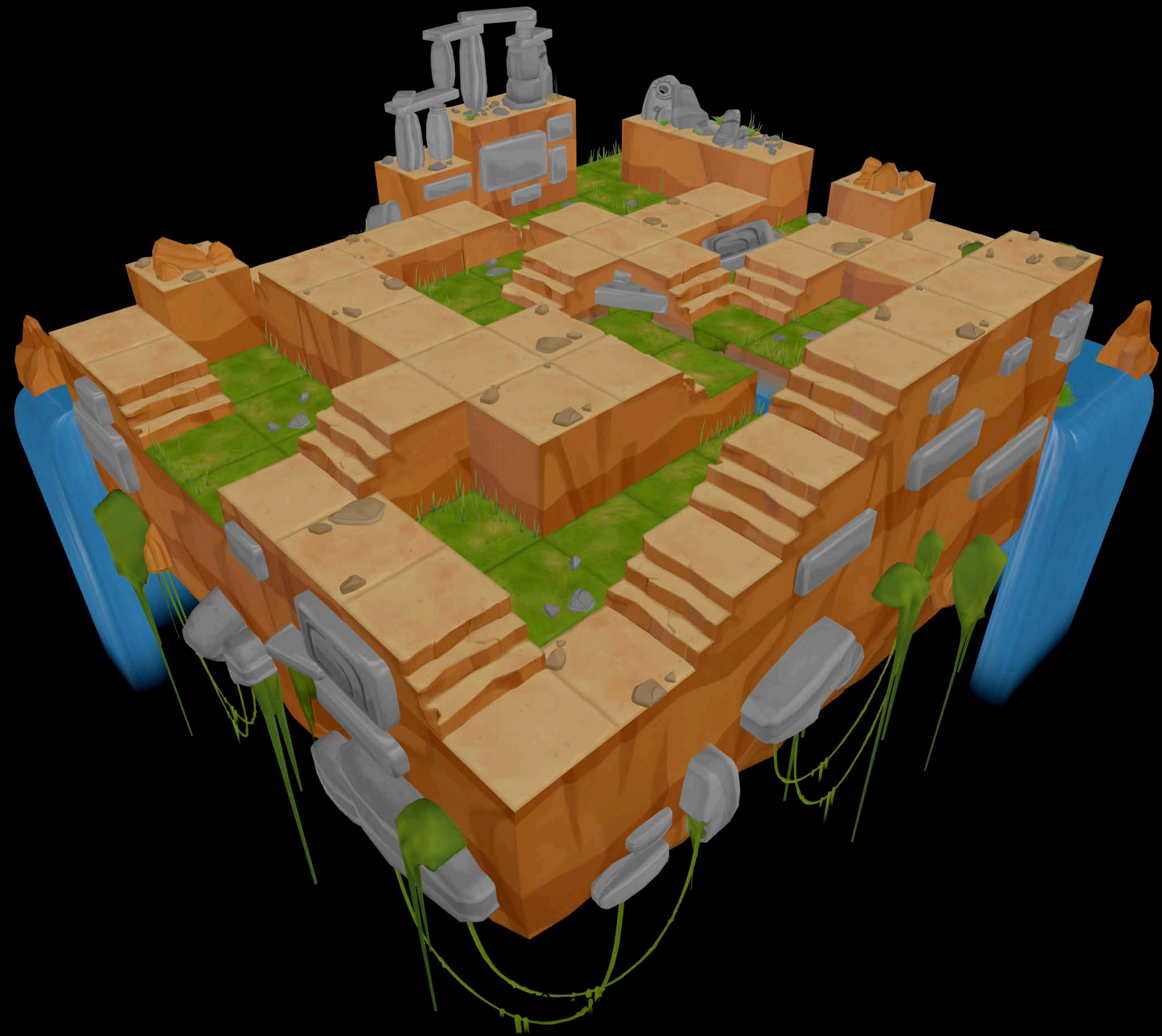
**Reduce Draw Call Count**

# Reducing Draw Calls

Geometry

Materials

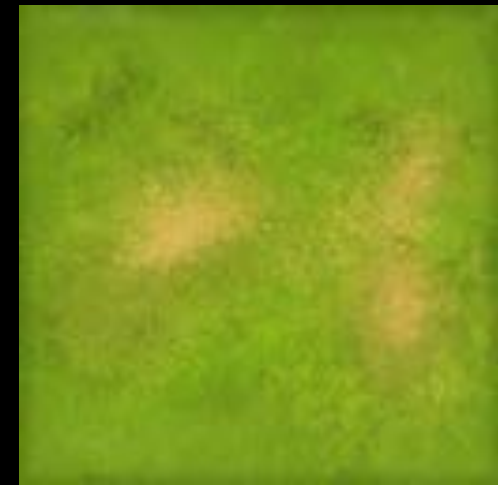
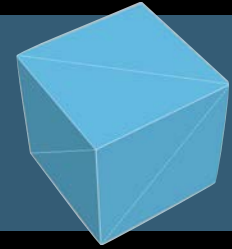
Lighting



1 draw call per mesh

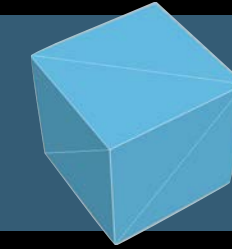
# Flattening Geometry

Geometry A



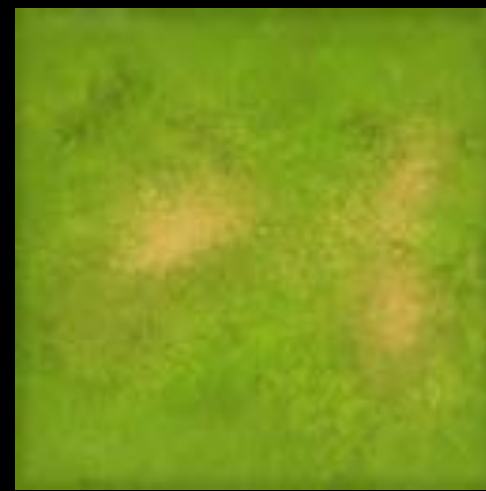
Grass Material

Geometry B



Grass Material

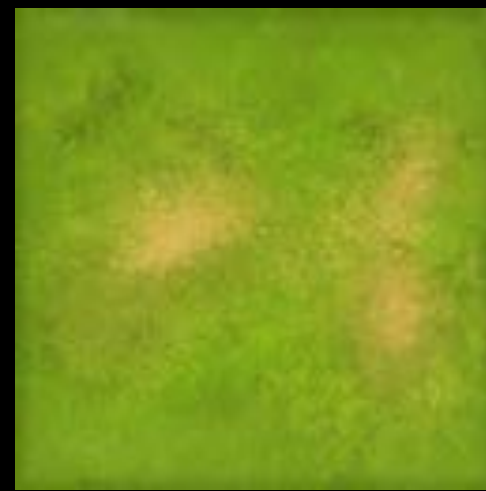
# Flattening Geometry



Grass Material



# Flattening Geometry



Grass Material

```
grassNode.flattenedClone()
```



# Flattening Geometry

## Tips

Store nodes to be flattened in a parent node

Save flattened scene

# Caveat

Large environments

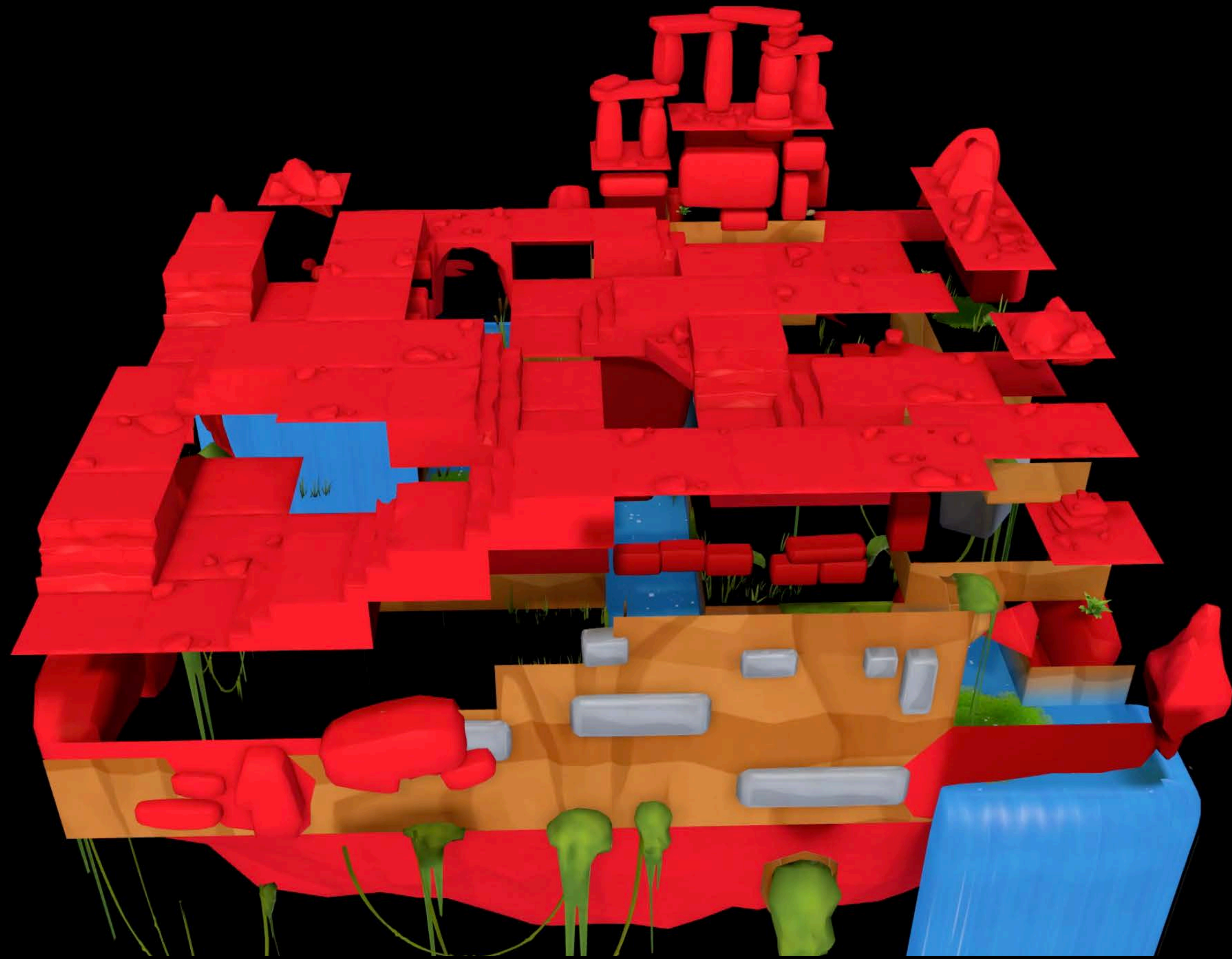


# Reducing Draw Calls

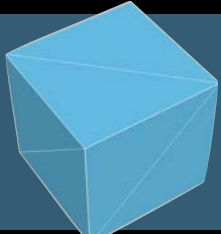
Geometry

Materials

Lighting



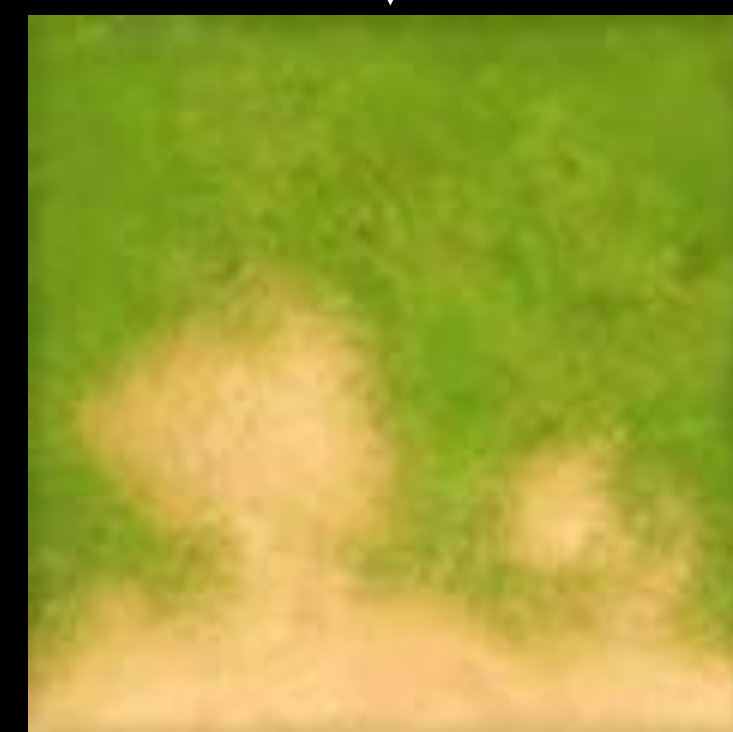
# Reducing Materials

Geometry A 



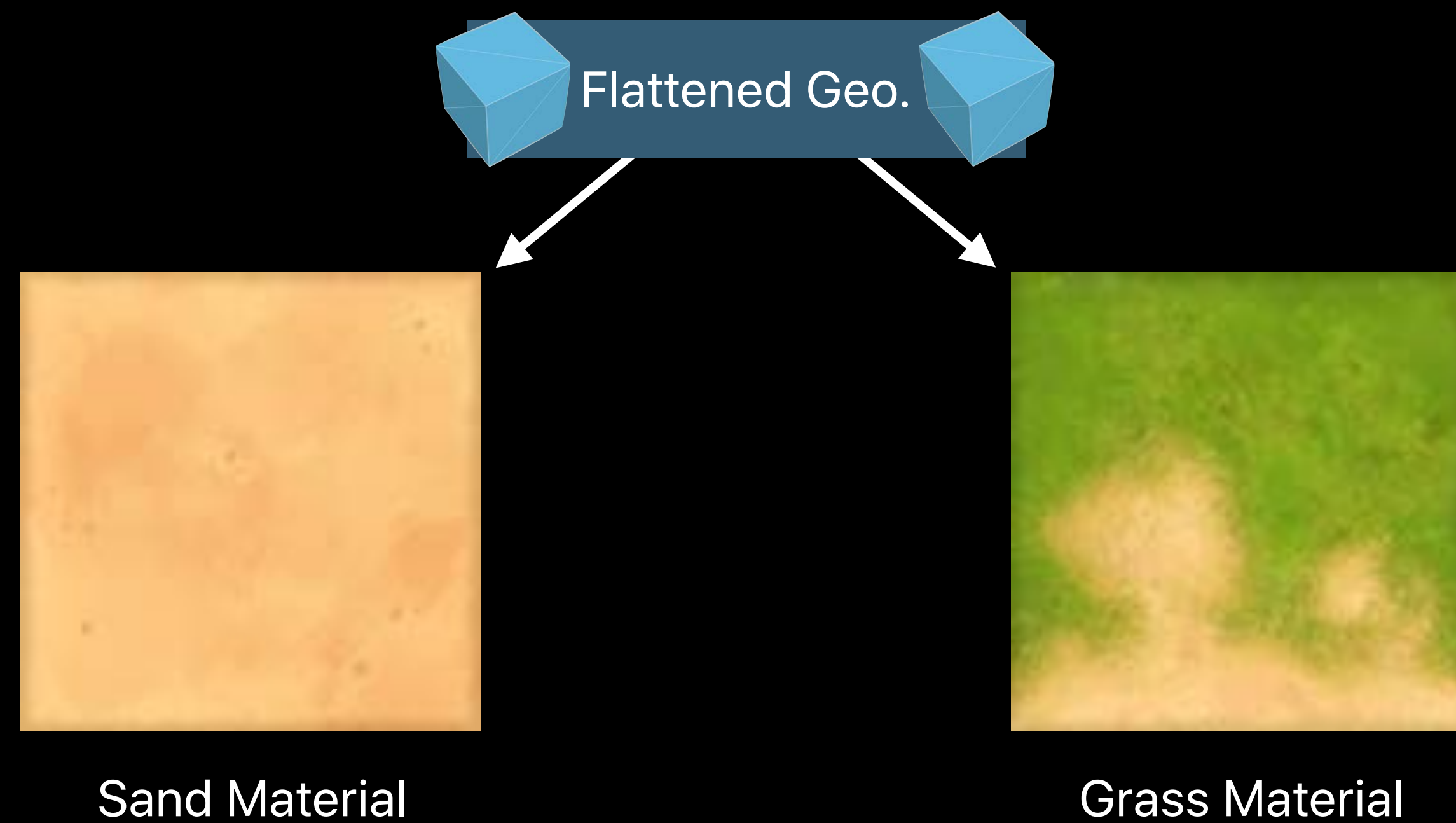
Sand Material

 Geometry B



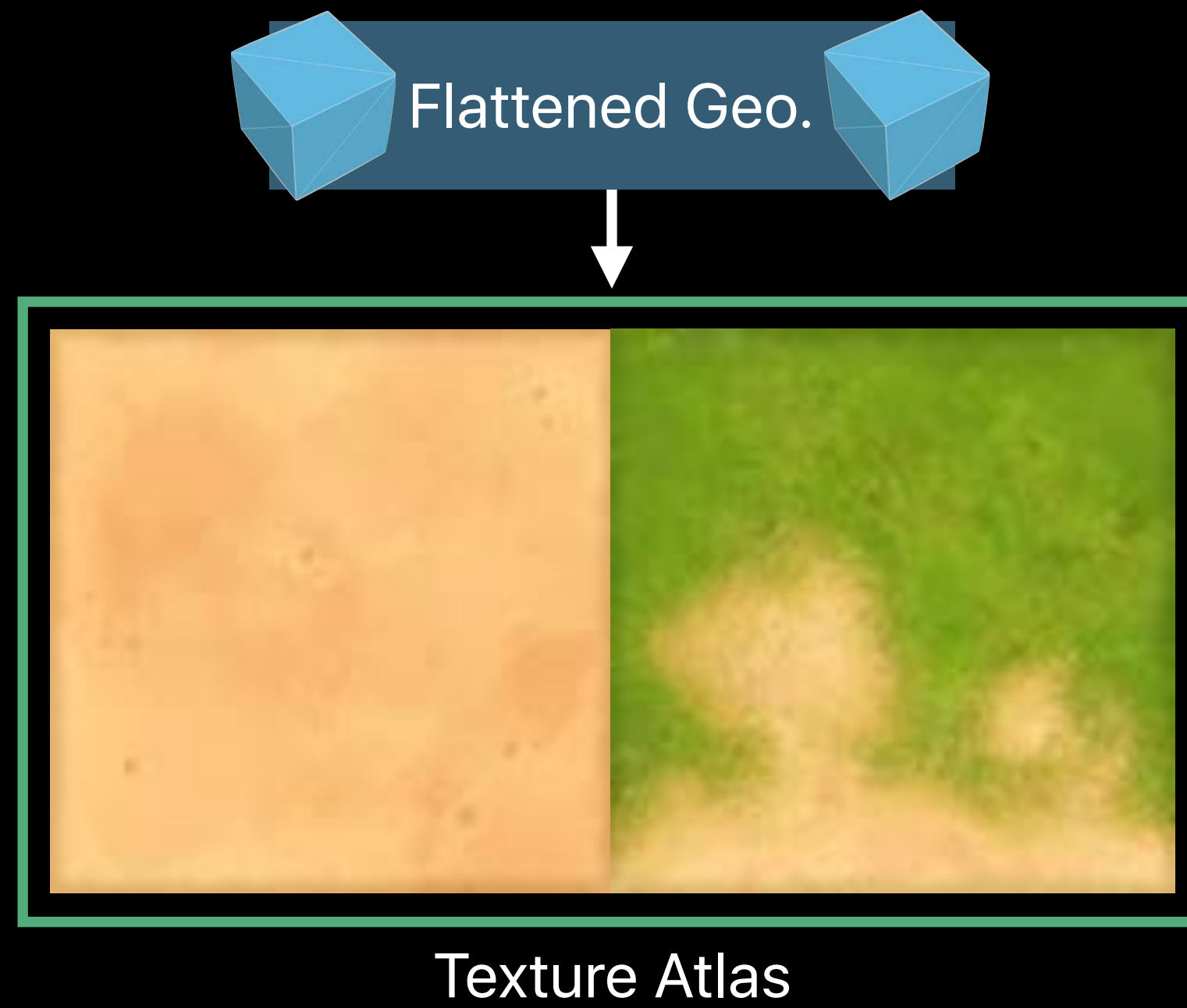
Grass Material

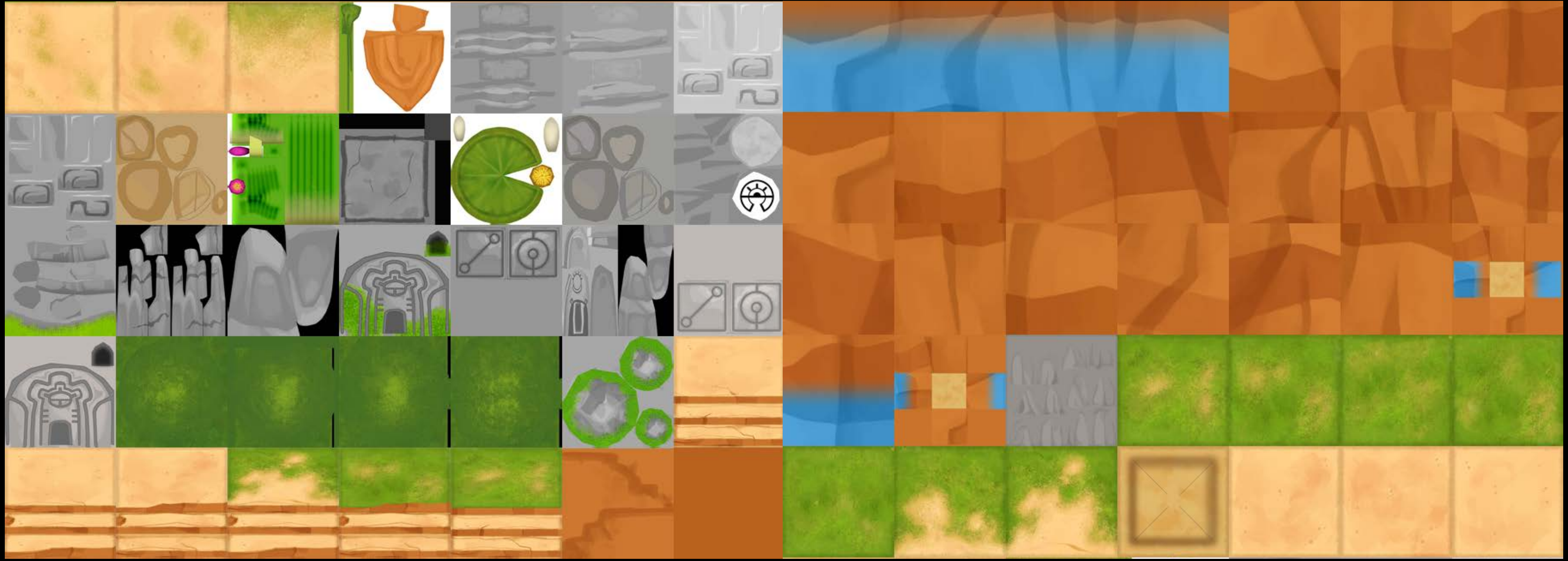
# Reducing Materials



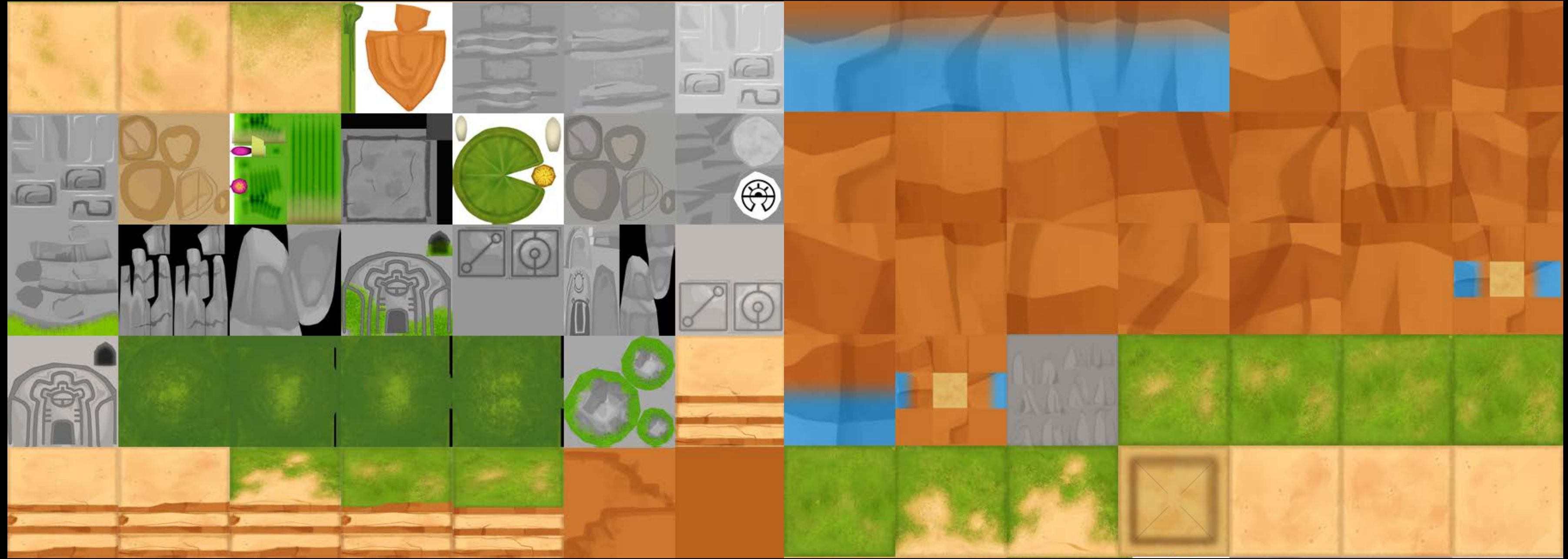


# Reducing Materials





5



14

# 1 Material

5

14



# Reducing Draw Calls

Geometry

Materials

Lighting

# Lights

---

Spotlight

1

---

Omni-Directional

4

---

Ambient

1

---



# Lights

## Baked lighting

---

Spotlight

1

---

Omni-Directional

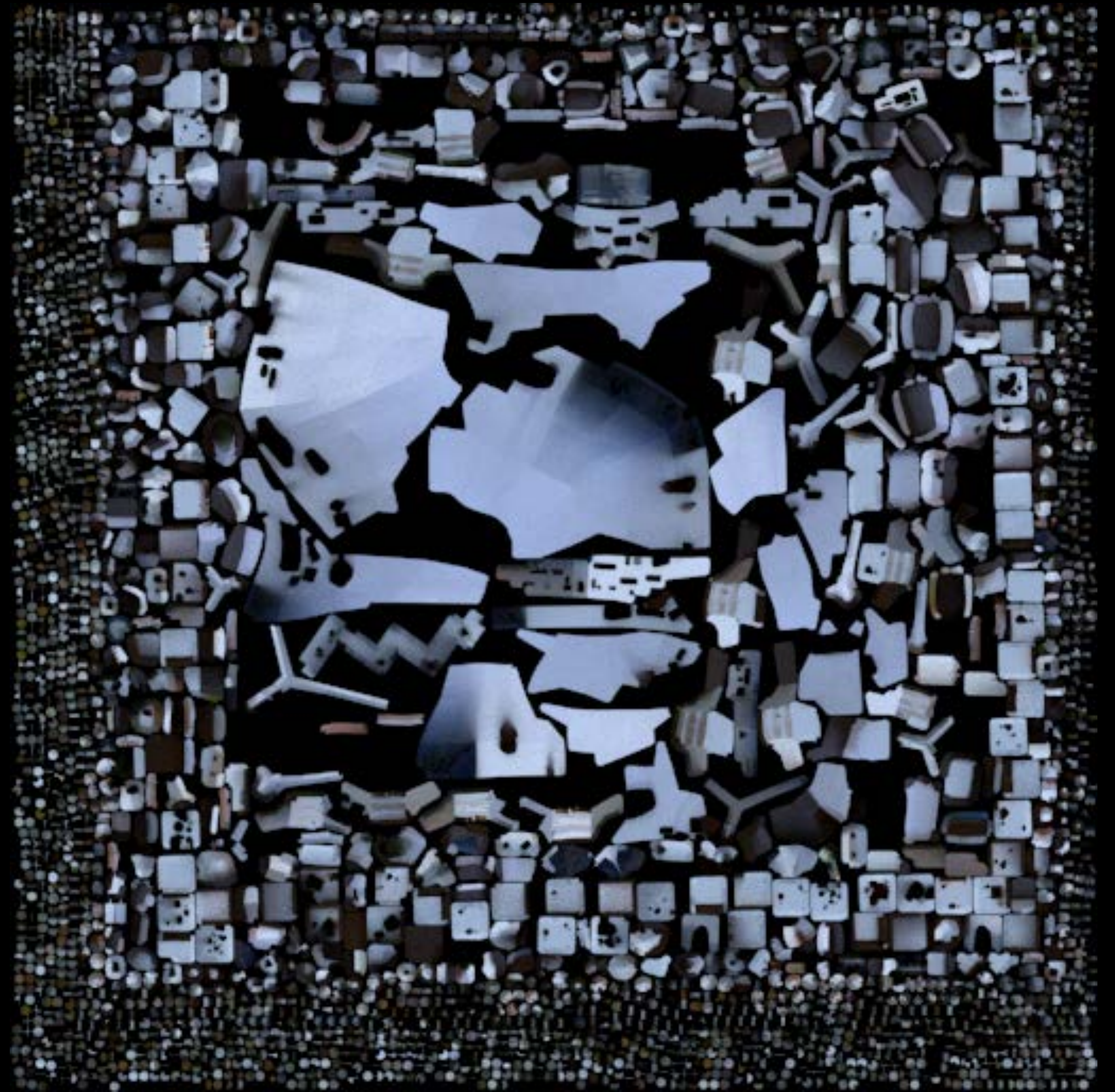
4

---

Ambient

1

---



# Lights

## Baked lighting

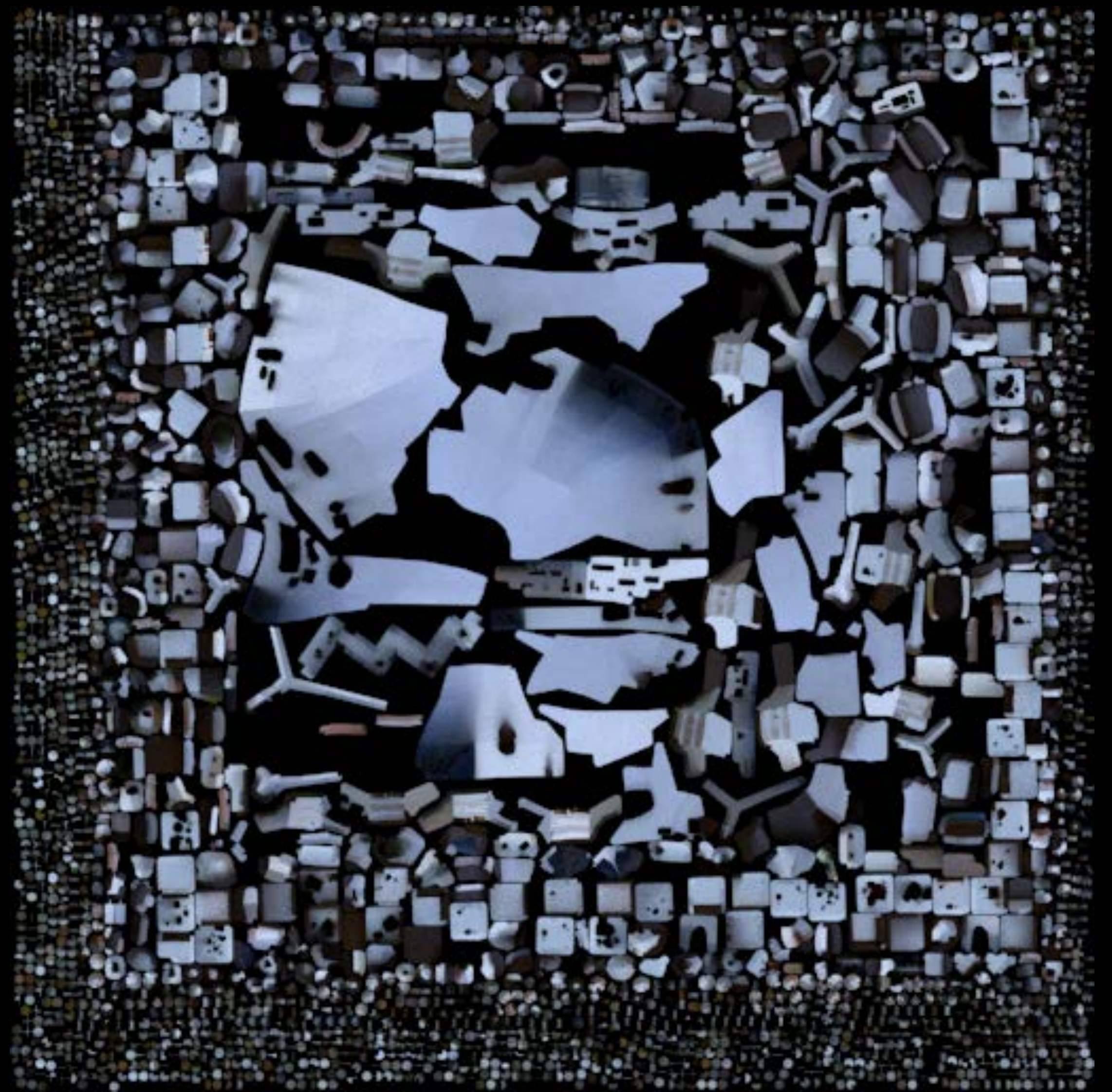
---

Spotlight

1

---

---







# < Which Way to Turn? >



0/1 
 0/6



Mt 60fps

73 167K



Animations	0.1ms	Delegate	
Physics	0.0ms	Rendering	2.3ms
Constraints	0.0ms	WaitDrawable	13.9ms
Particles	0.4ms	2D	

Run My Code

Uniforms	2.4 MB
Textures	14.3 MB
Element	KB
Sources	MB

Hint

-  Mt 60fps

◆73 ▲167K



Animations	0.1ms	Delegate	
Physics	0.0ms	Rendering	2.3ms
Constraints	0.0ms	WaitDrawable	13.9ms
Particles	0.4ms	2D	

 Run My Code

Uniforms	2.4 MB
Textures	14.3 MB
Elements	1.2 KB
Sources	1.2 MB

 Hint


— **Mt 60fps** ◆73 ▲167K









Category	Value
Animations	0.1ms
Physics	0.0ms
Constraints	0.0ms
Particles	0.4ms
Delegate	
Rendering	2.3ms
WaitDrawable	13.9ms
2D	


Asset Type	Memory Usage
Uniforms	2.4 MB
Textures	14.3 MB
Element	KB
Sources	MB


Run My Code





Hint

—  Mt 60fps ◆73 ▲167K

 Animations	0.1ms	 Delegate	
 Physics	0.0ms	 Rendering	2.3ms
 Constraints	0.0ms	 WaitDrawable	13.9
 Particles	0.4ms	 2D	

 **Run My Code**



 Uniforms	2.4 MB
 Textures	14.3 MB
 Elements	KB
 Sources	MB

**Hint**

-  Mt 60fps

◆73 ▲167K



Animations	0.1ms	Delegate	
Physics	0.0ms	Rendering	2.3ms
Constraints	0.0ms	WaitDrawable	13.9
Particles	0.4ms	2D	

 Run My Code

Uniforms	2.4 MB
Textures	14.3 MB
Element	KB
Sources	MB

Hint

# Optimizing Performance

Flatten geometry

Use a texture atlas

Use light maps

# Related Sessions

---

Enhancements in SceneKit

WWDC 2015

---

Advanced in SceneKit Rendering

WWDC 2015

---

SceneKit: What's New

WWDC 2017

---

# Labs

---

Creating Content for Swift Playgrounds Lab

Technology Lab E

Thur 12:00PM–3:10PM

---



