

Building a Game with SceneKit

Session 610

Amaury Balliet

Software Engineer







Score

0

1:58

Time



Related Sessions

SceneKit basics

WWDC 2013

- "What's new in SceneKit"

WWDC 2014

- "What's new in SceneKit"
- "What's new in SpriteKit"

Outline

Where to start

How to give life to the scene

How we made the Bananas demo

How to get great performance

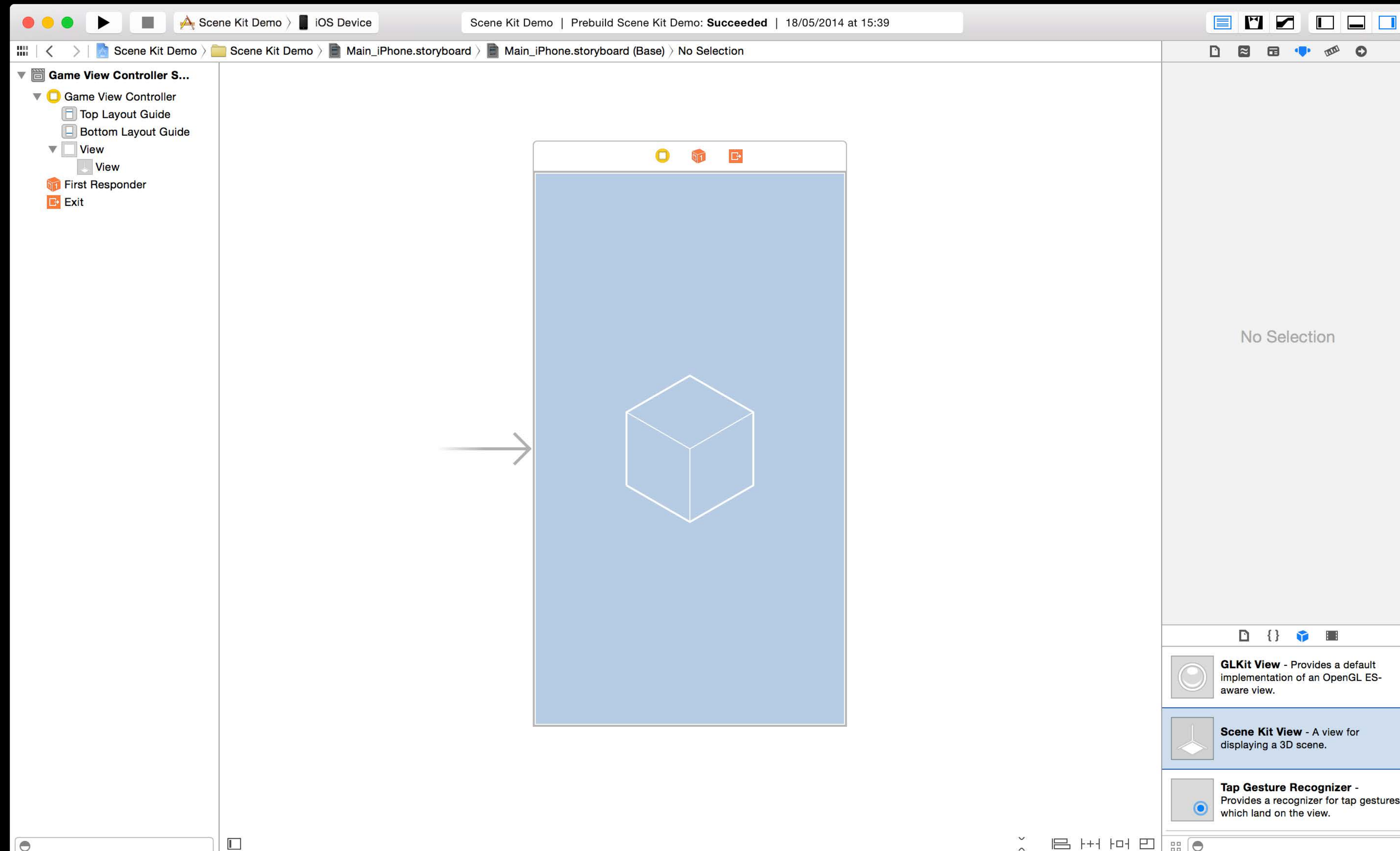
How you can build your own tools

Starting a Game

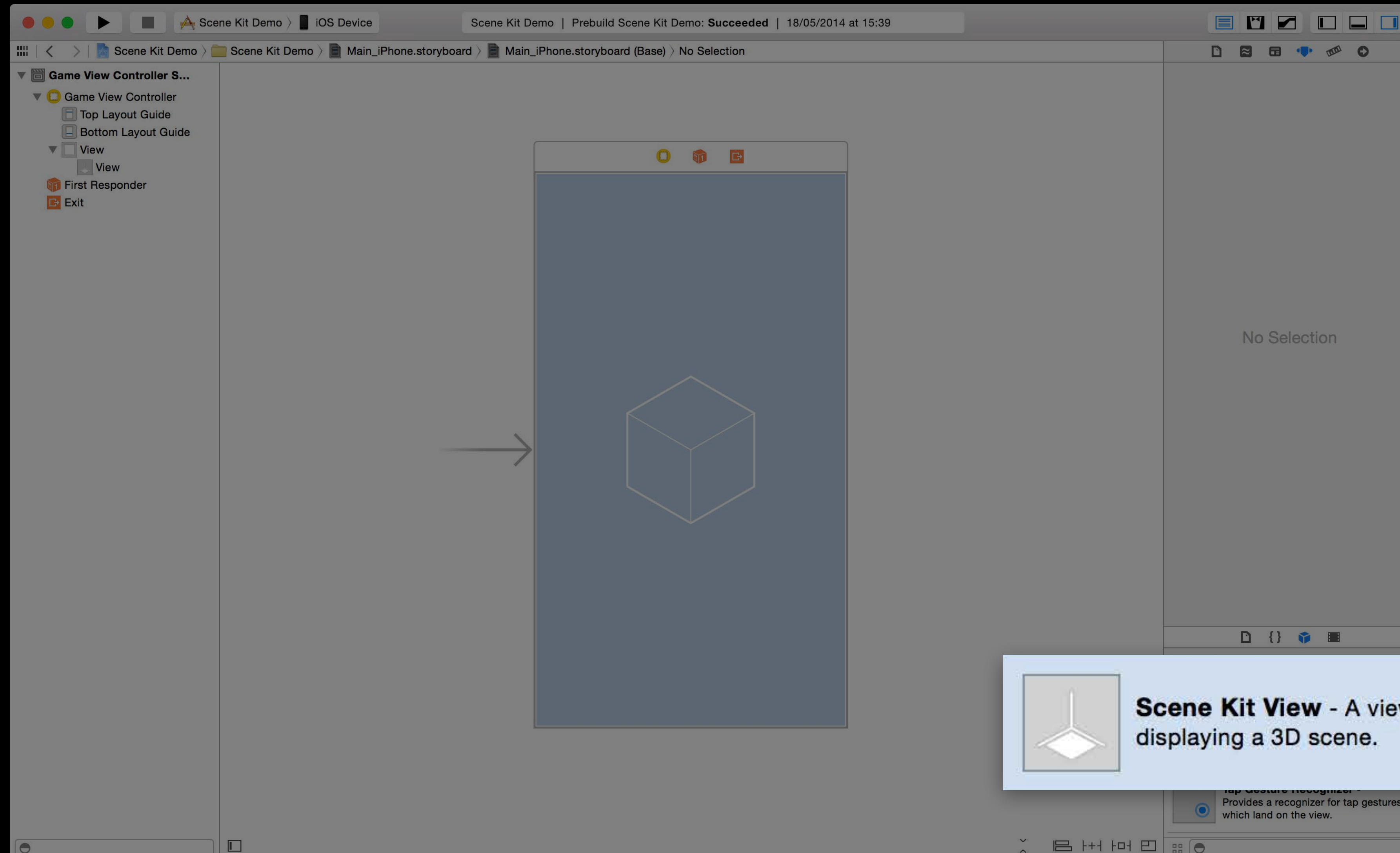
Displaying a 3D scene

Displaying a 3D Scene

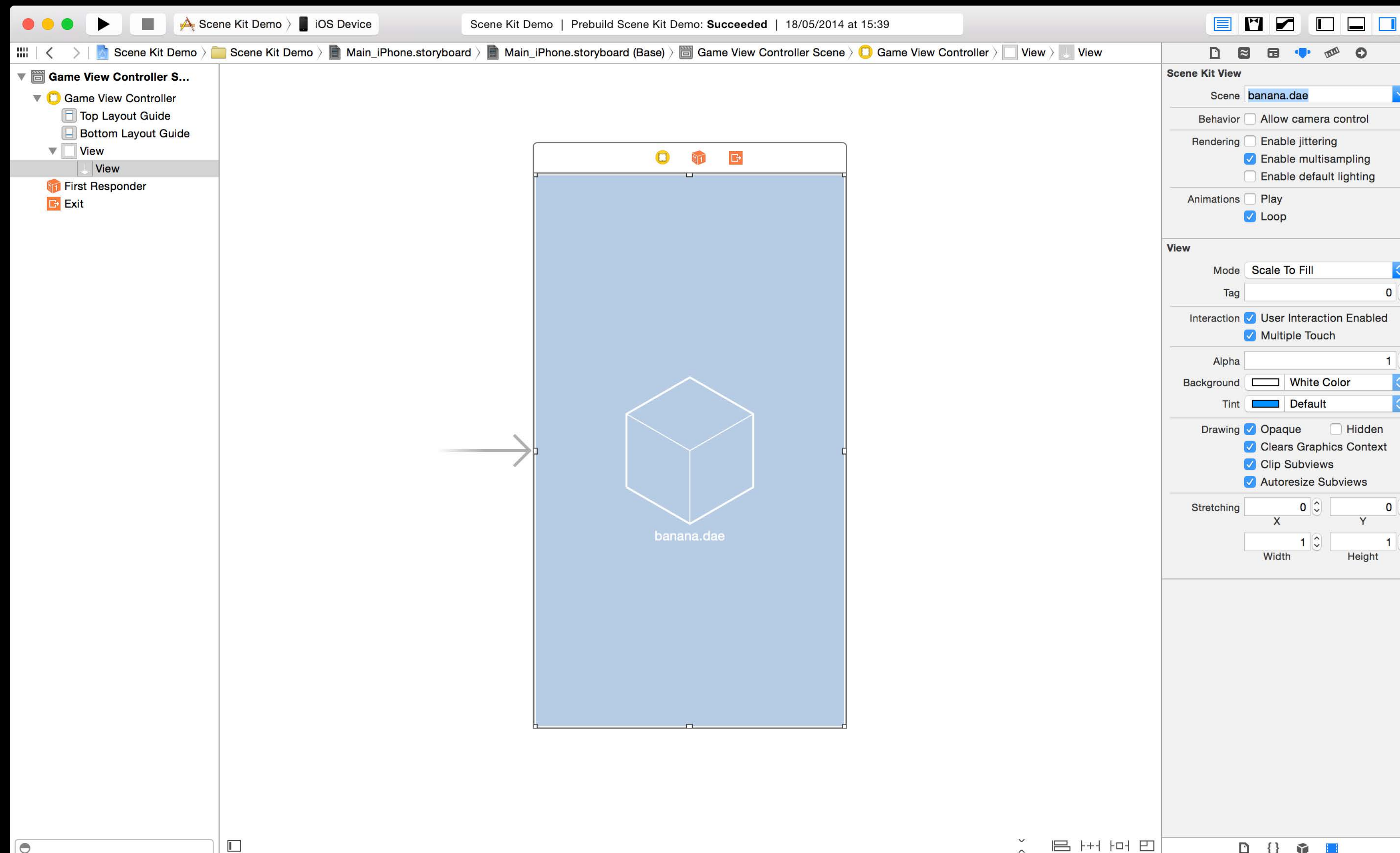
Interface Builder



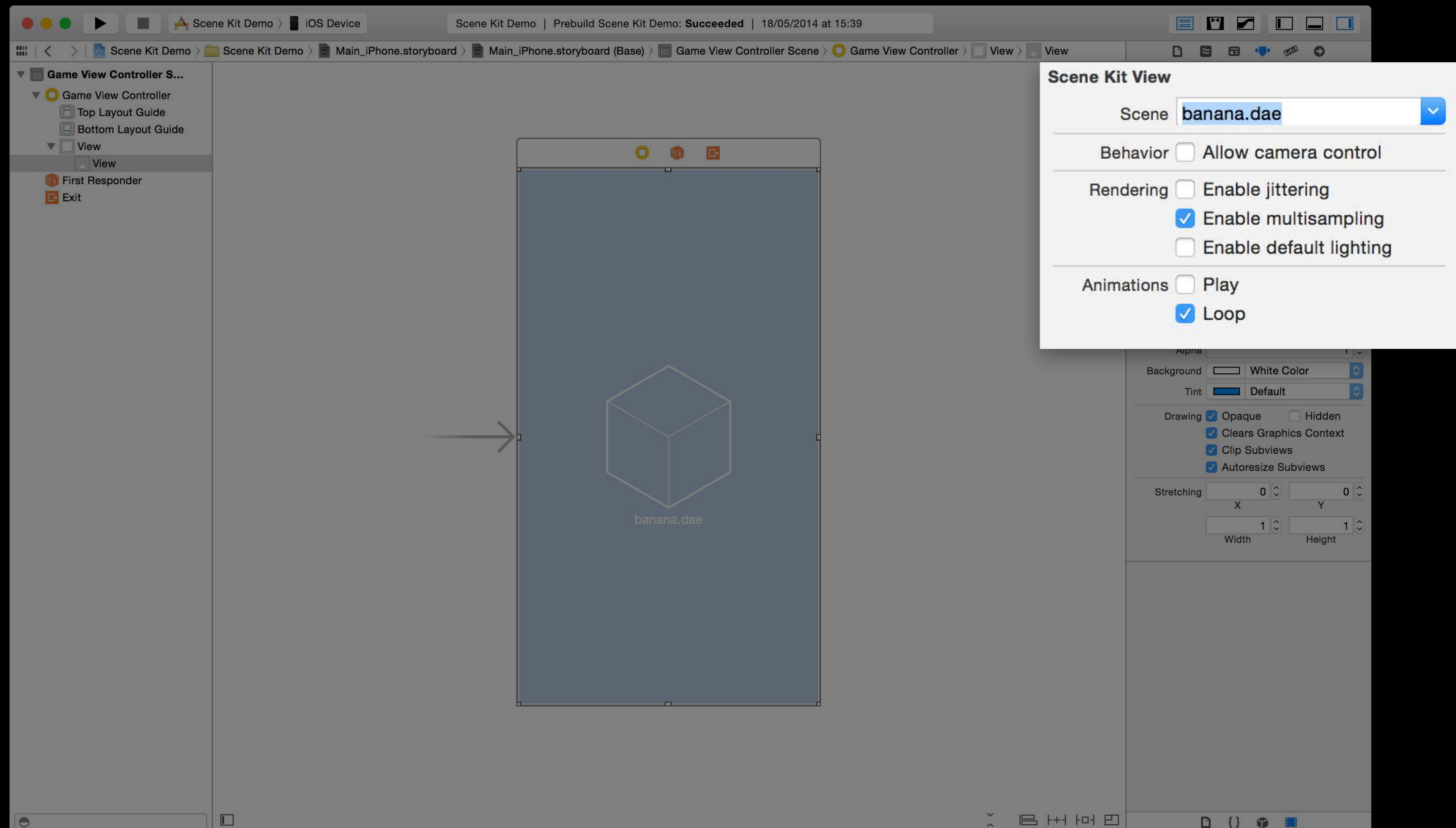
Displaying a 3D Scene Interface Builder



Displaying a 3D Scene Interface Builder



Displaying a 3D Scene Interface Builder



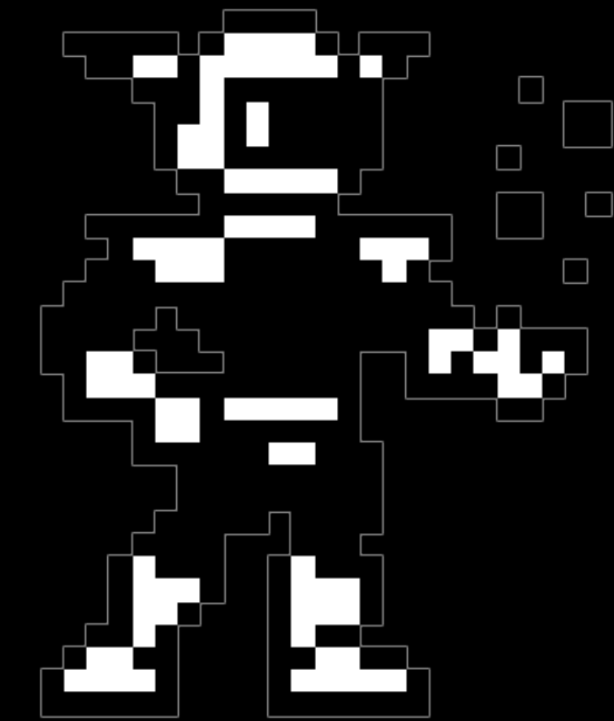
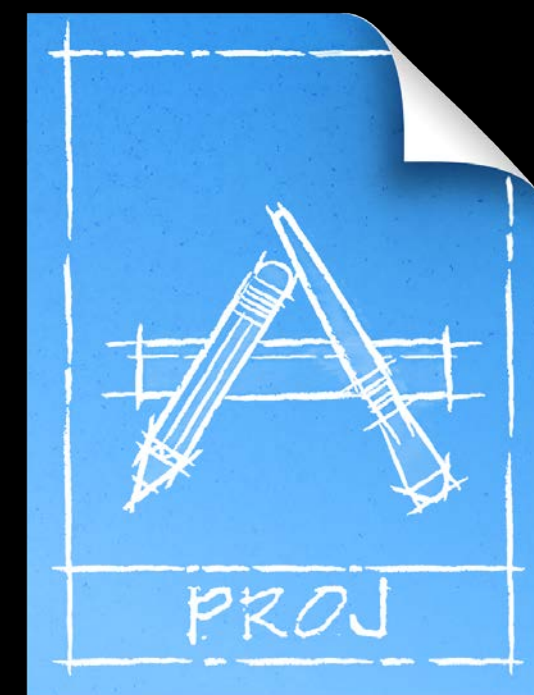
Displaying a 3D Scene

Interface Builder



Xcode Template

Unified template for games

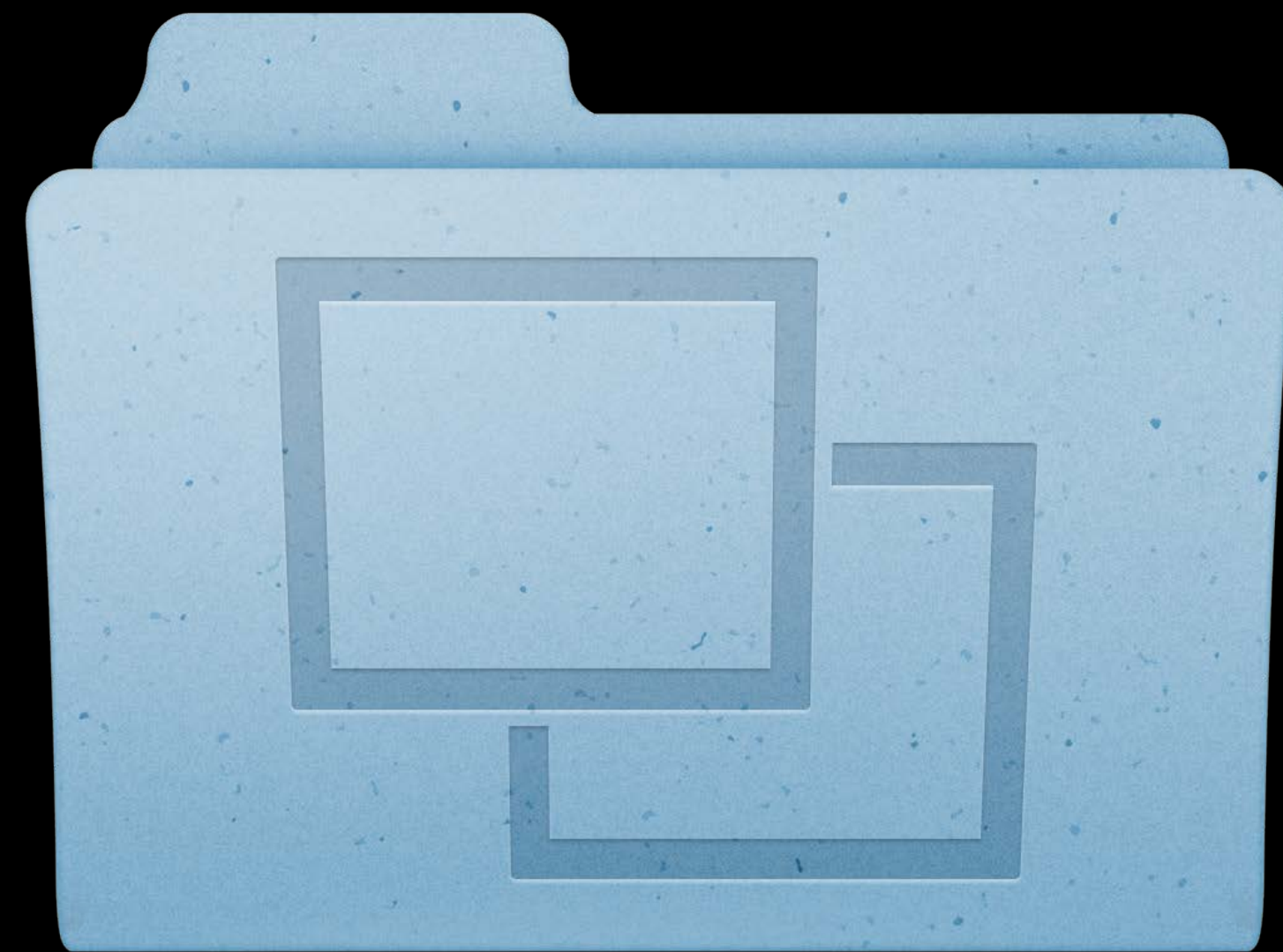


SceneKit
SpriteKit
GLES

Managing Assets

SceneKit asset catalog

NEW



game.scnassets

Managing Assets

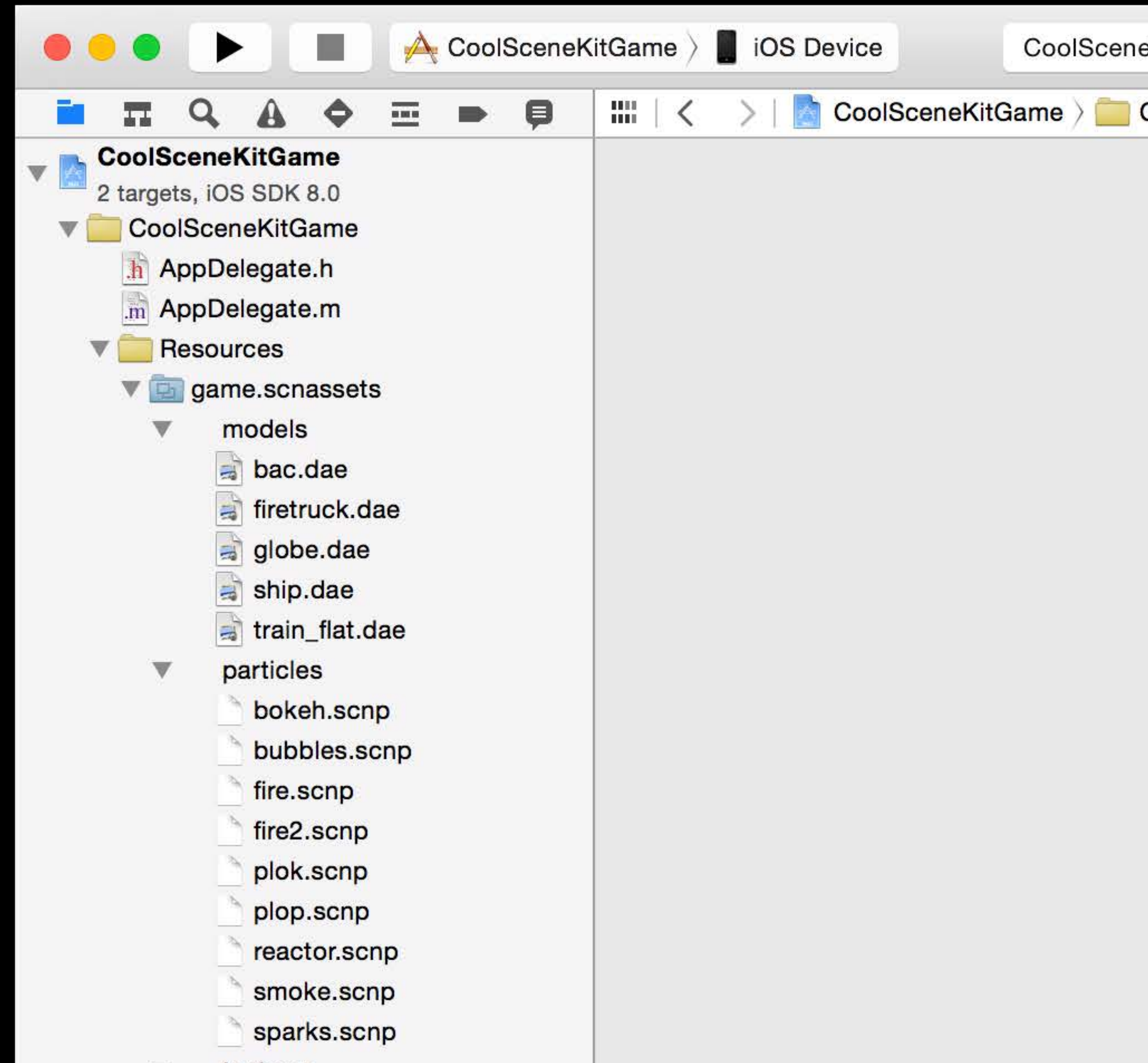
SceneKit asset catalog



Preserved file hierarchy

Organize models, textures, particles, etc

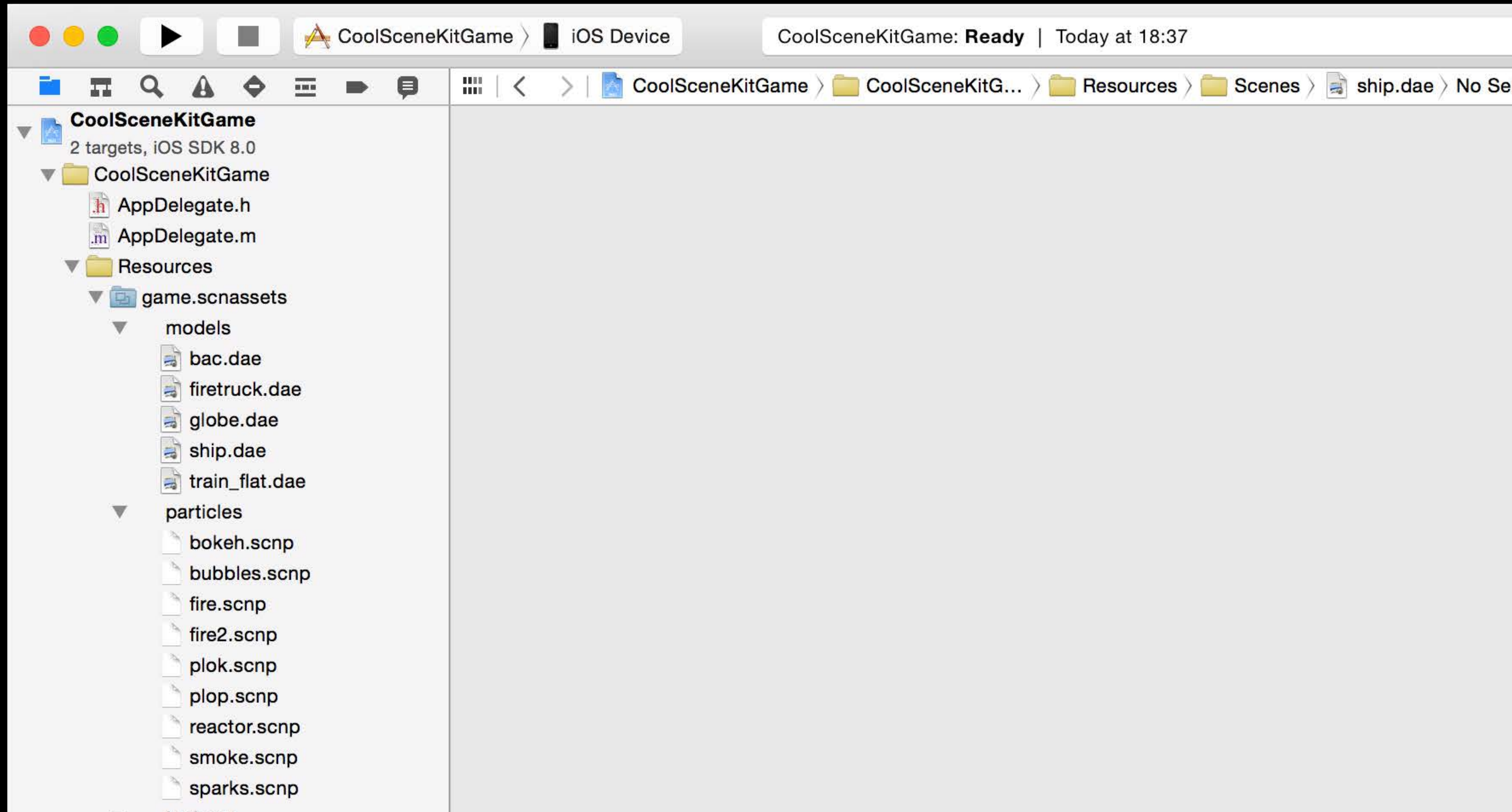
Automatic optimizations



Managing Assets

SceneKit asset catalog

NEW



Managing Assets

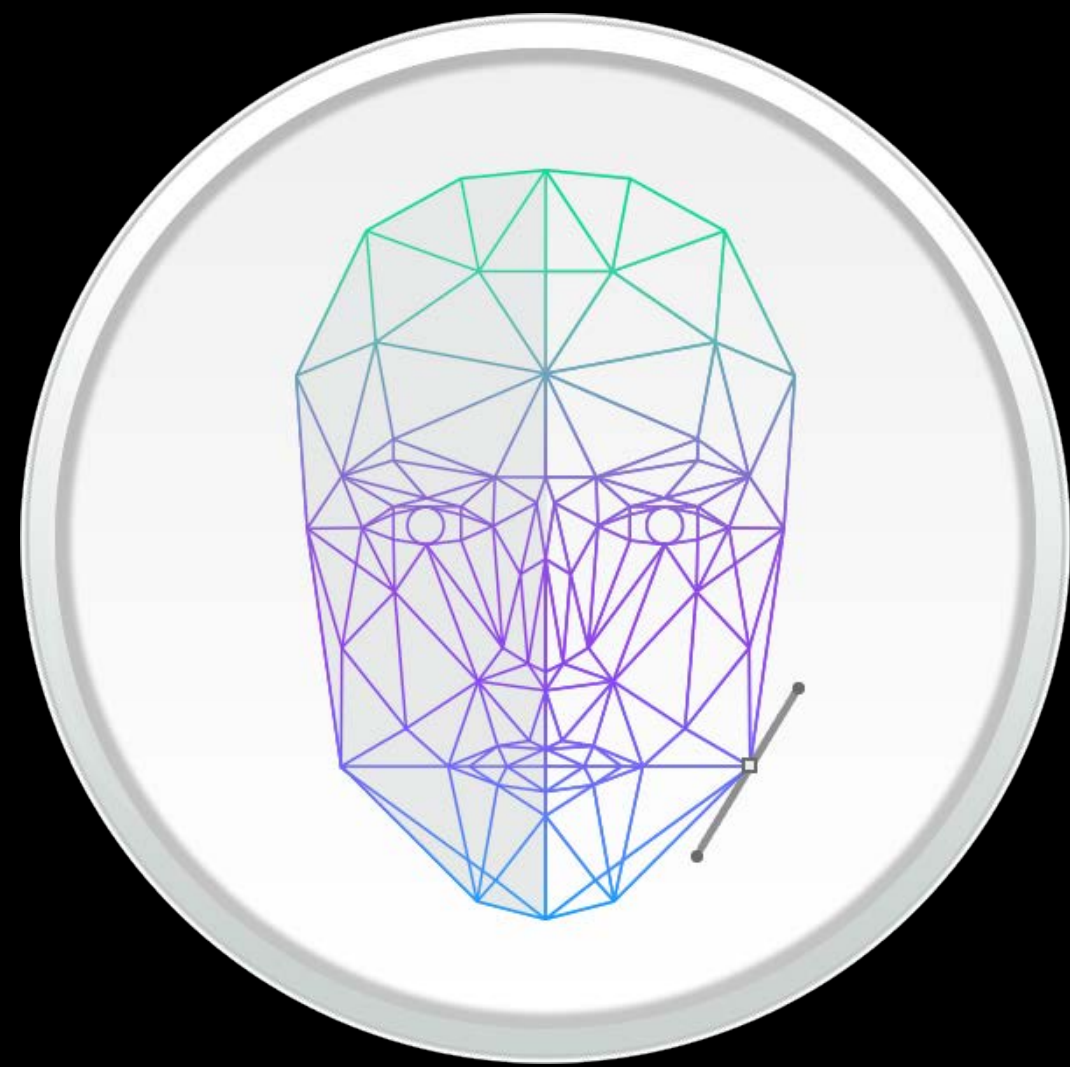
SceneKit asset catalog



	iOS	OS X
Up axis conversion	✓	✓
Geometry interleaving	✓	✓
PVRTC textures	✓	

Workflow

Working with artists



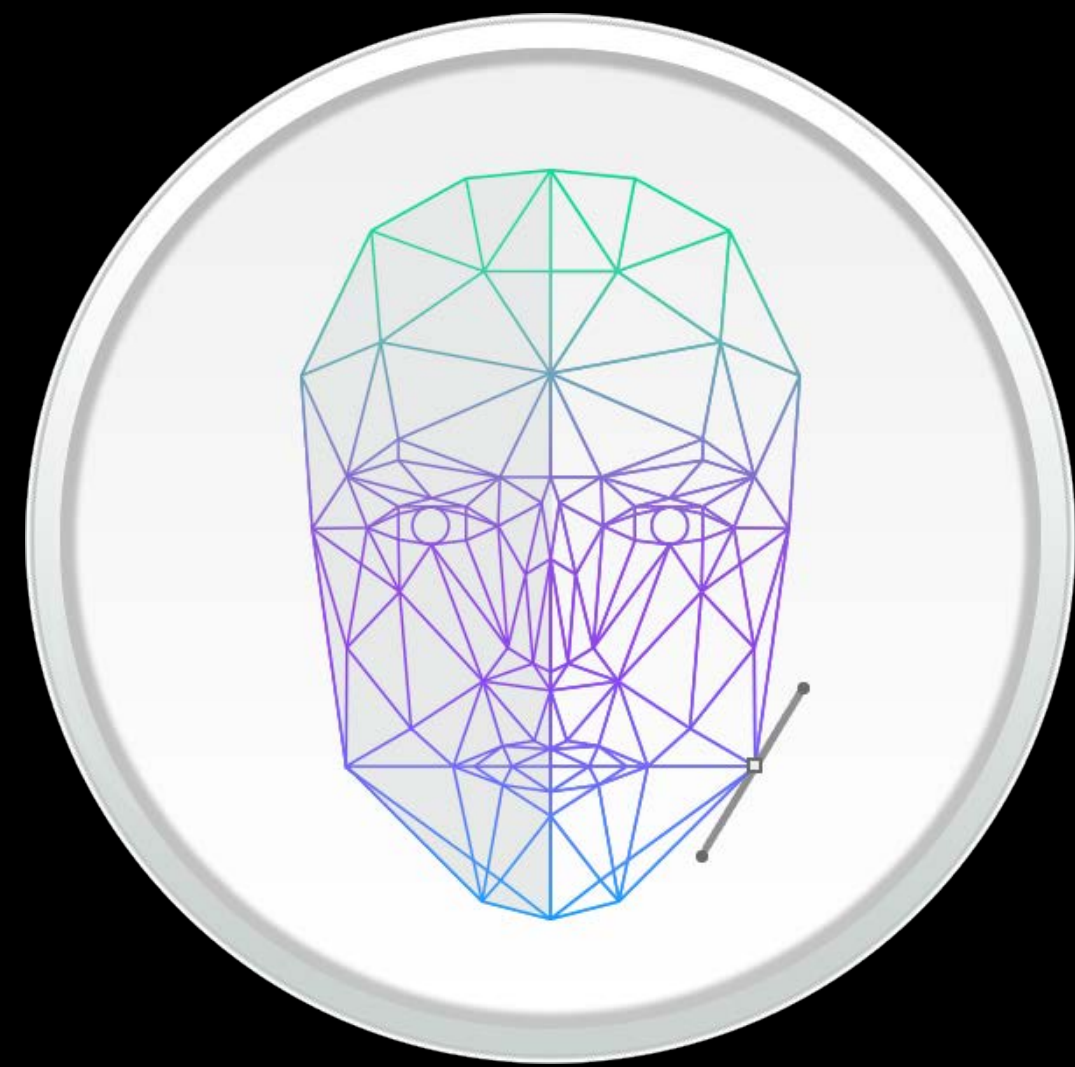
Artists



Developers

Workflow

Working with artists



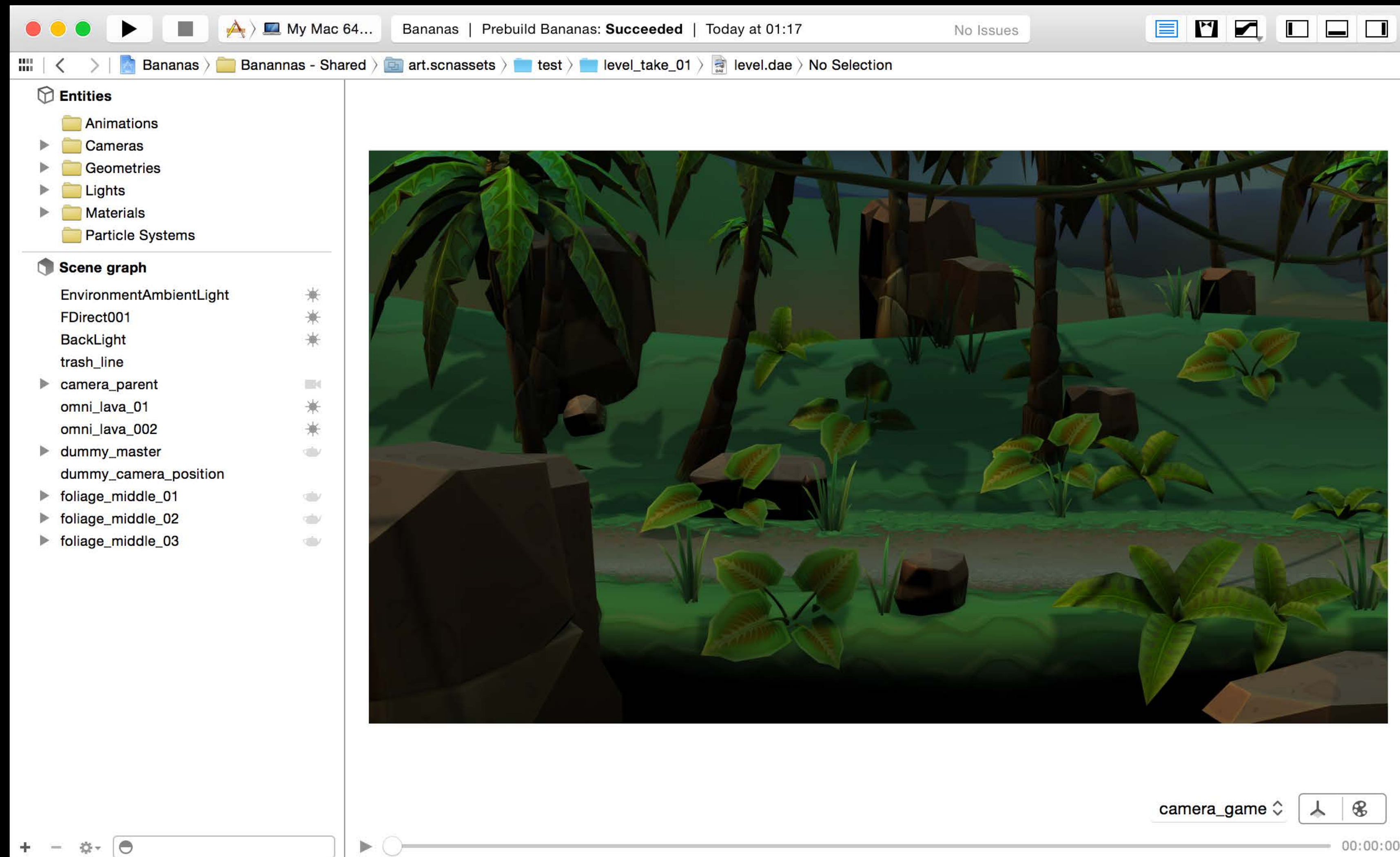
Artists



Developers

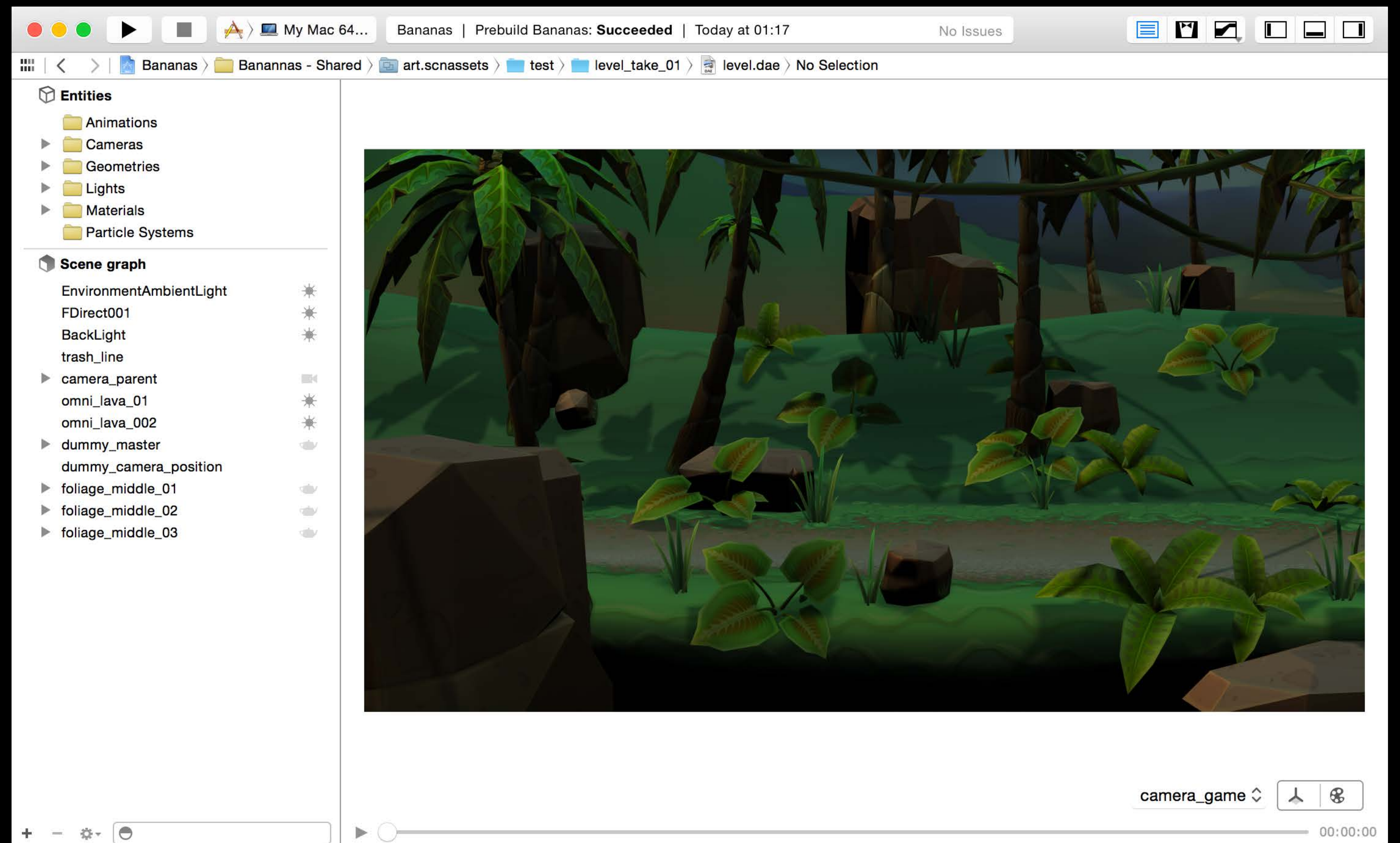
Editing Assets

Scene editor



Editing Assets

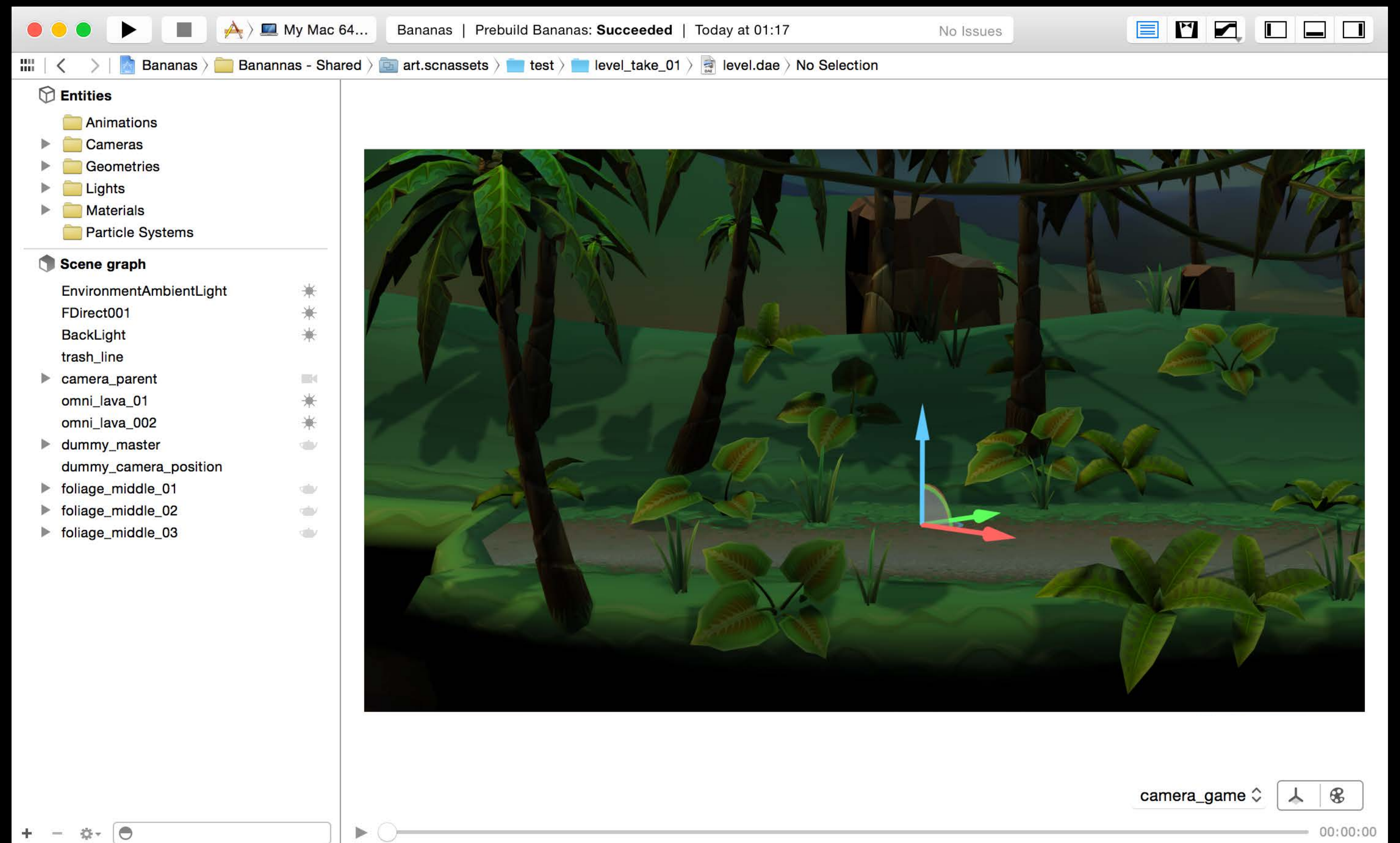
Scene editor



Editing Assets

Scene editor

Direct manipulation

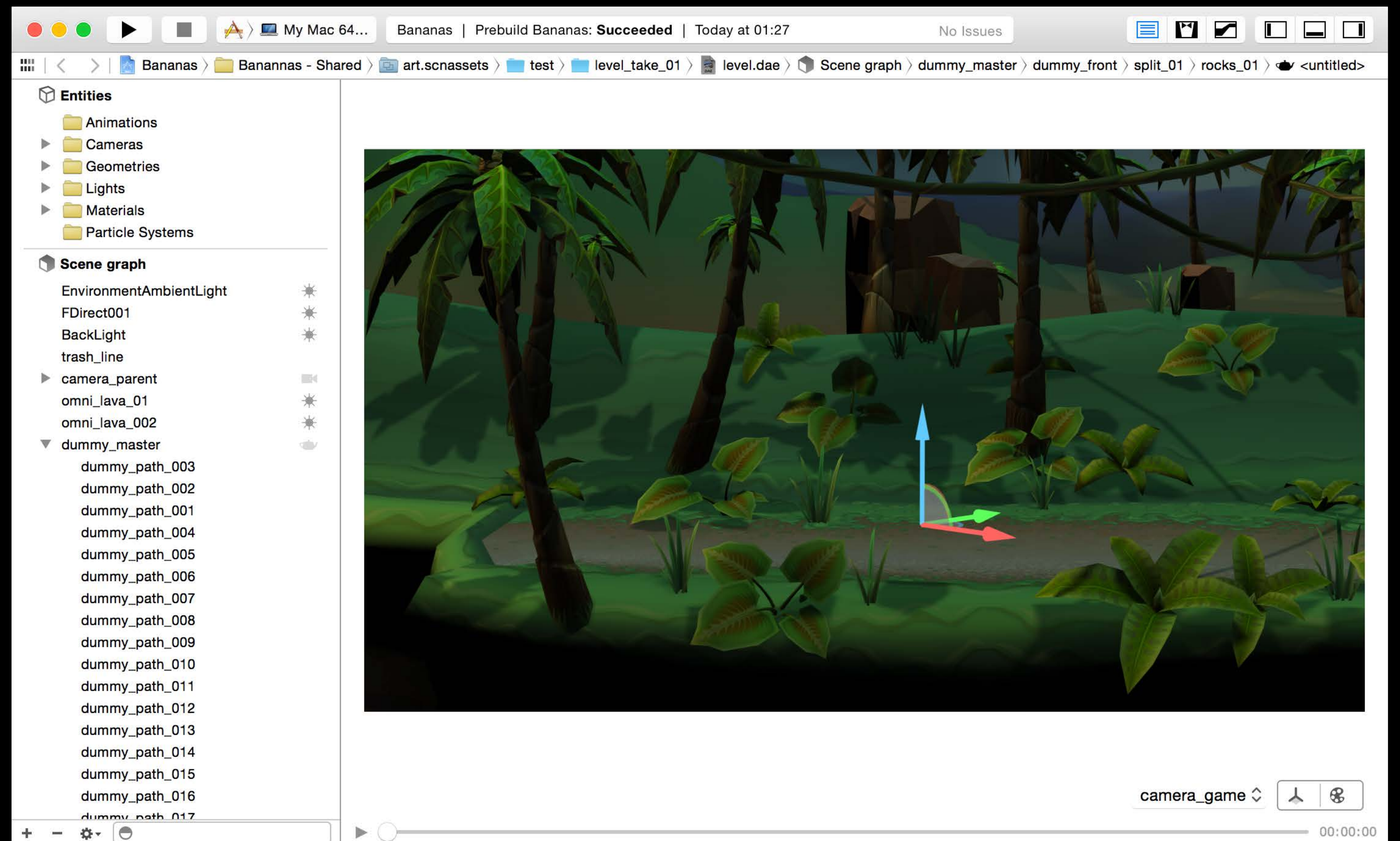


Editing Assets

Scene editor

Direct manipulation

Node hierarchy

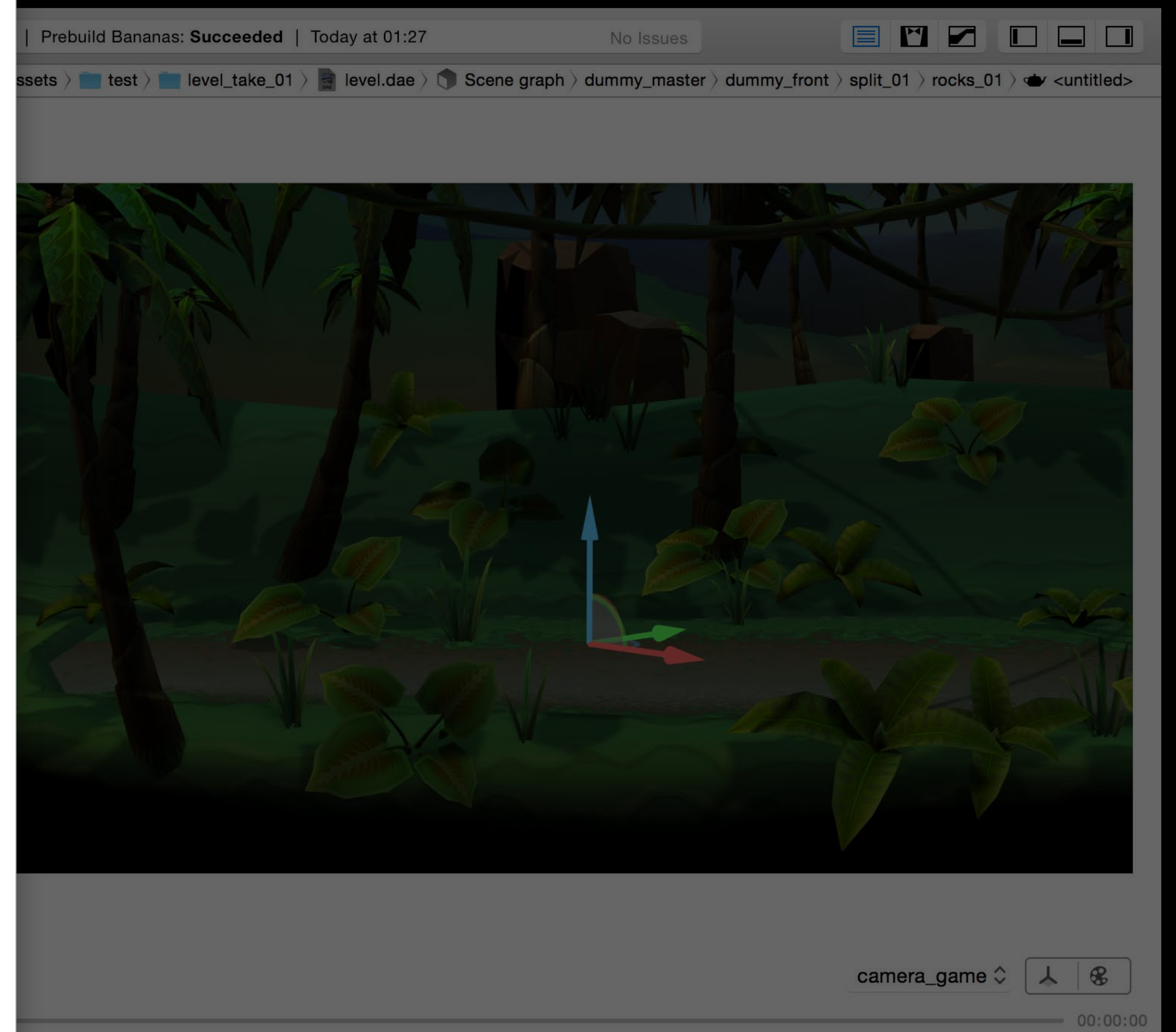
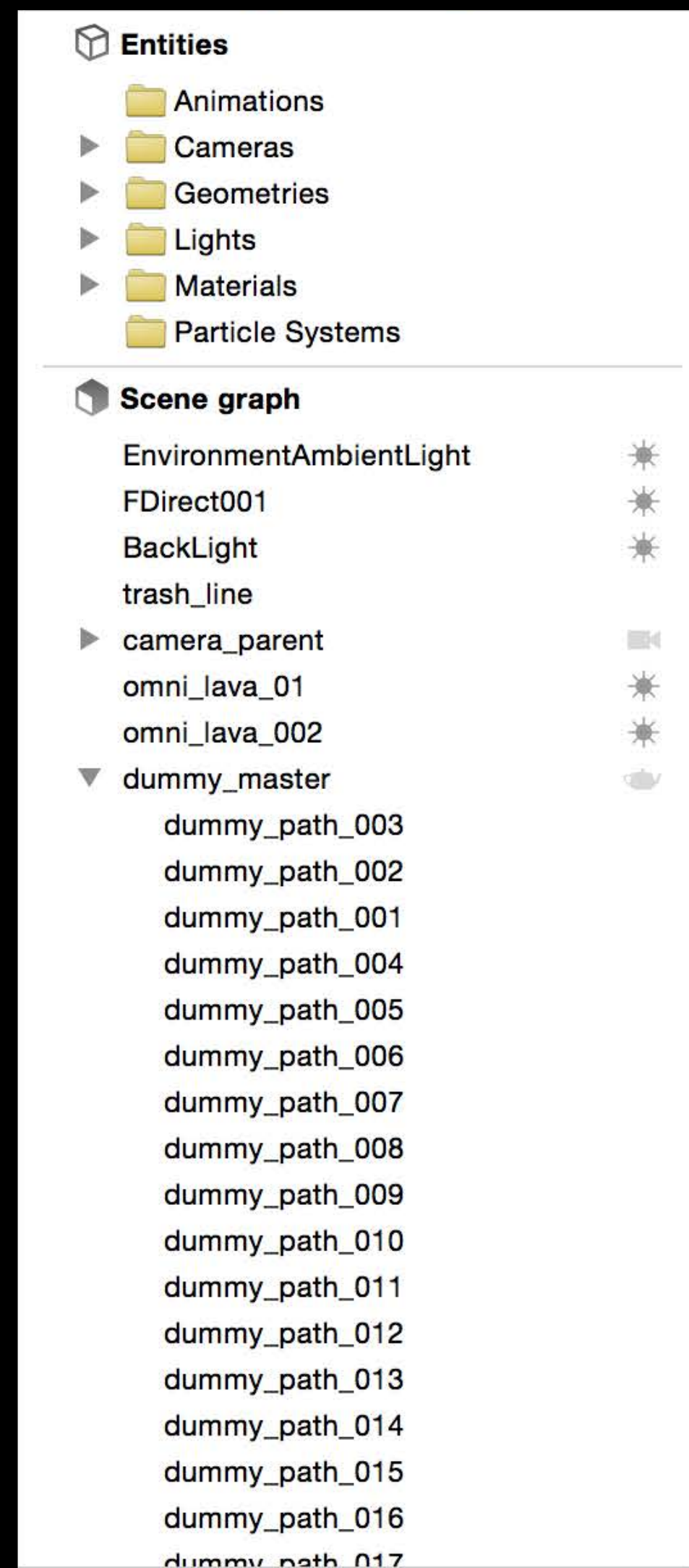


Editing Assets

Scene editor

Direct manipulation

Node hierarchy

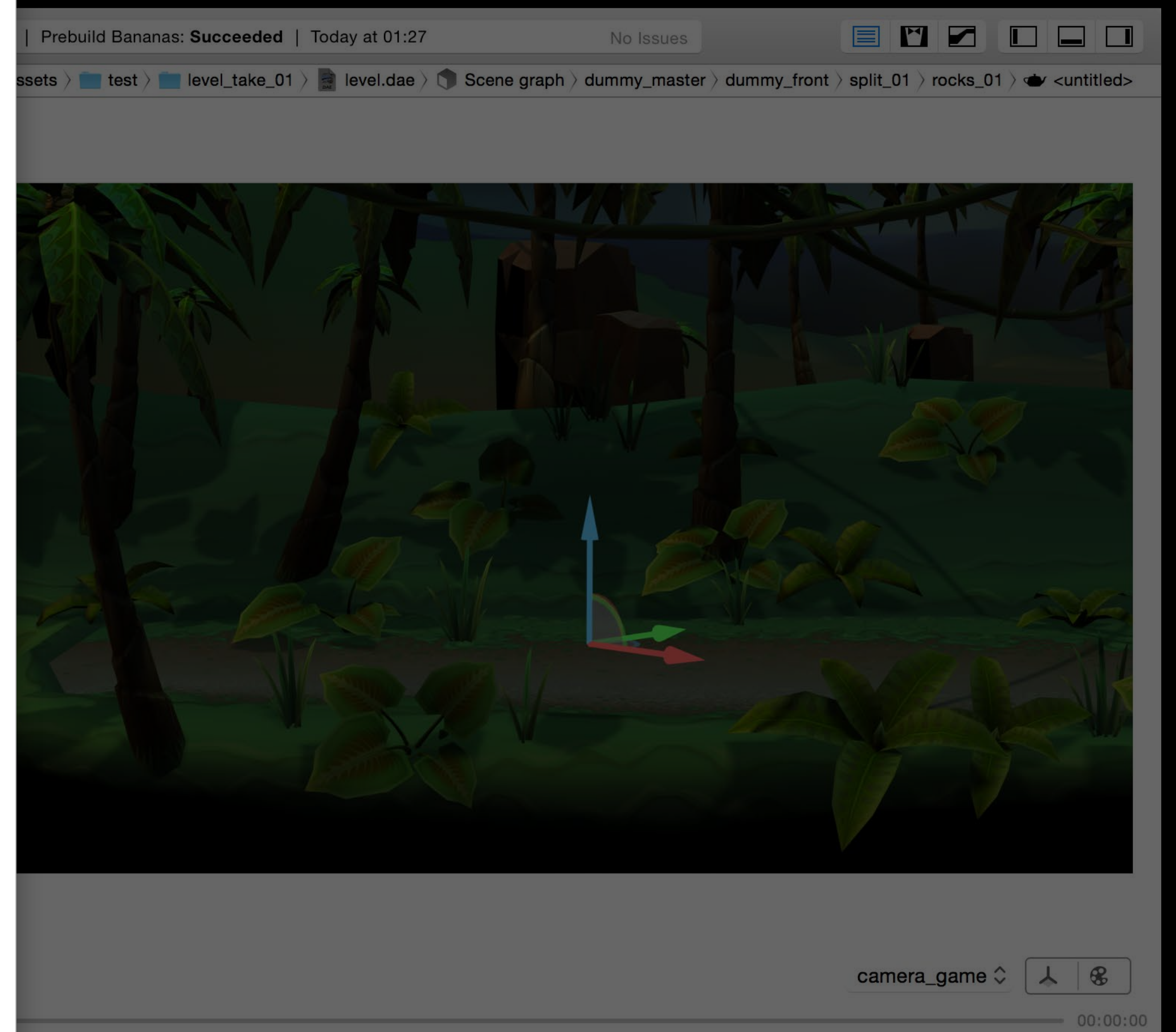
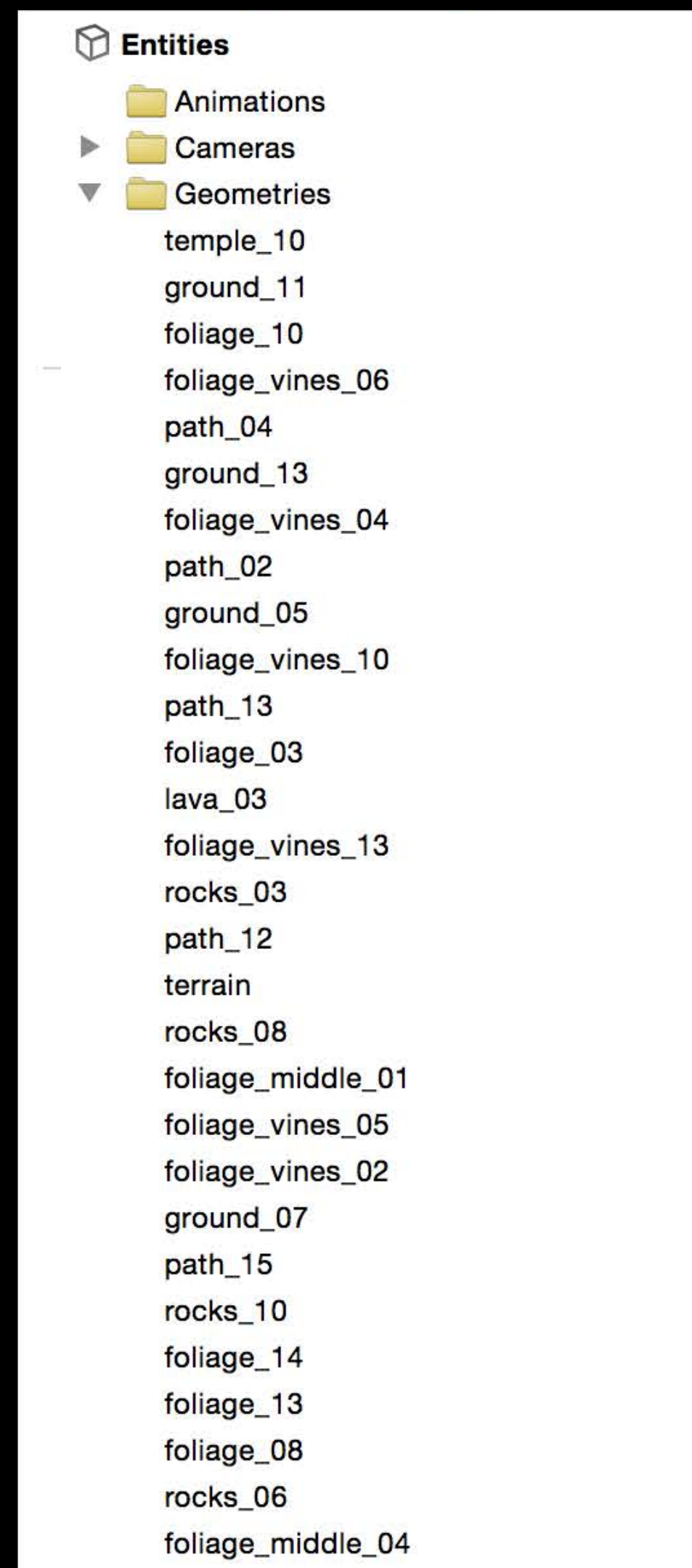


Editing Assets

Scene editor

Direct manipulation

Node hierarchy and entities



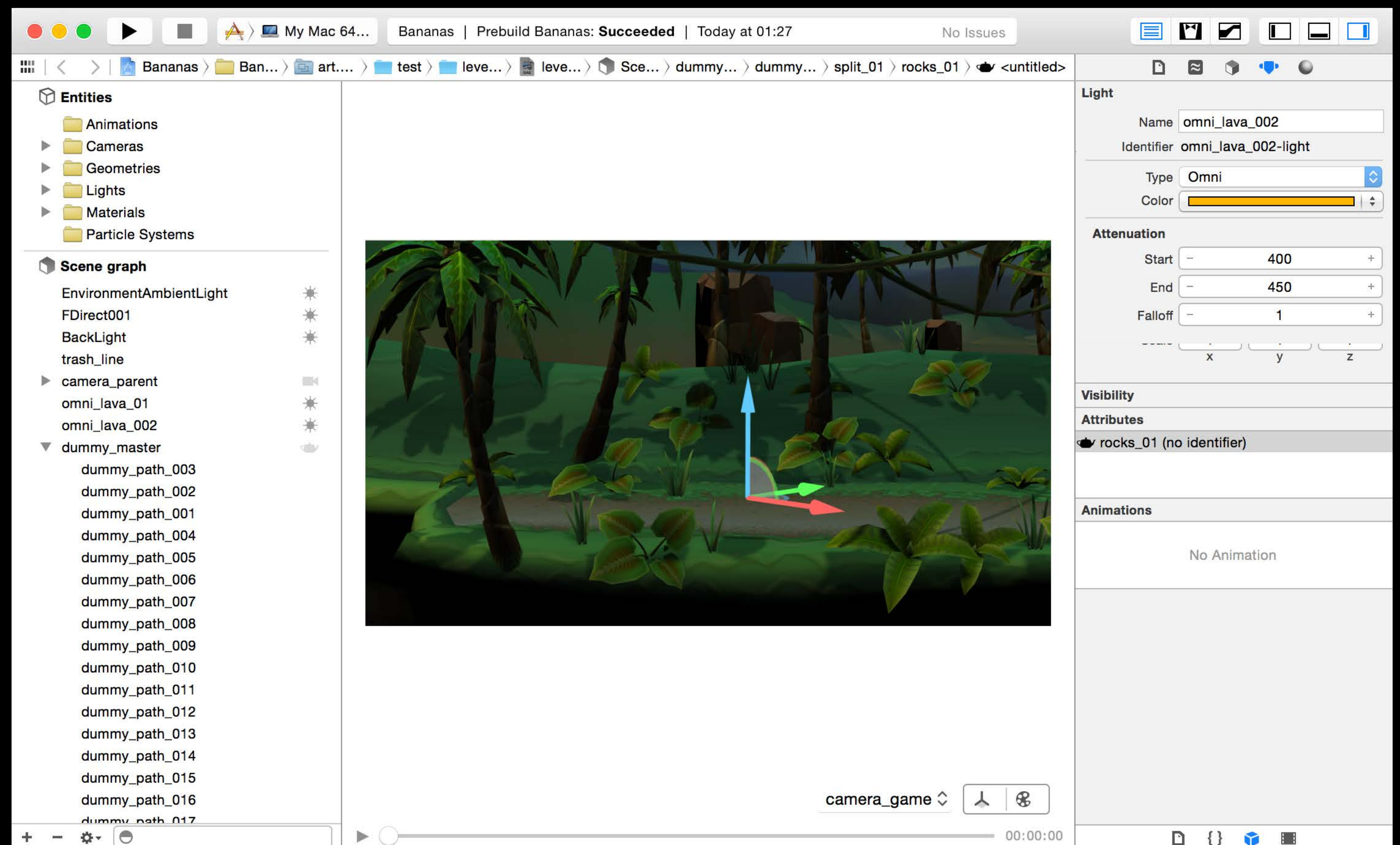
Editing Assets

Scene editor

Direct manipulation

Node hierarchy and entities

Node properties



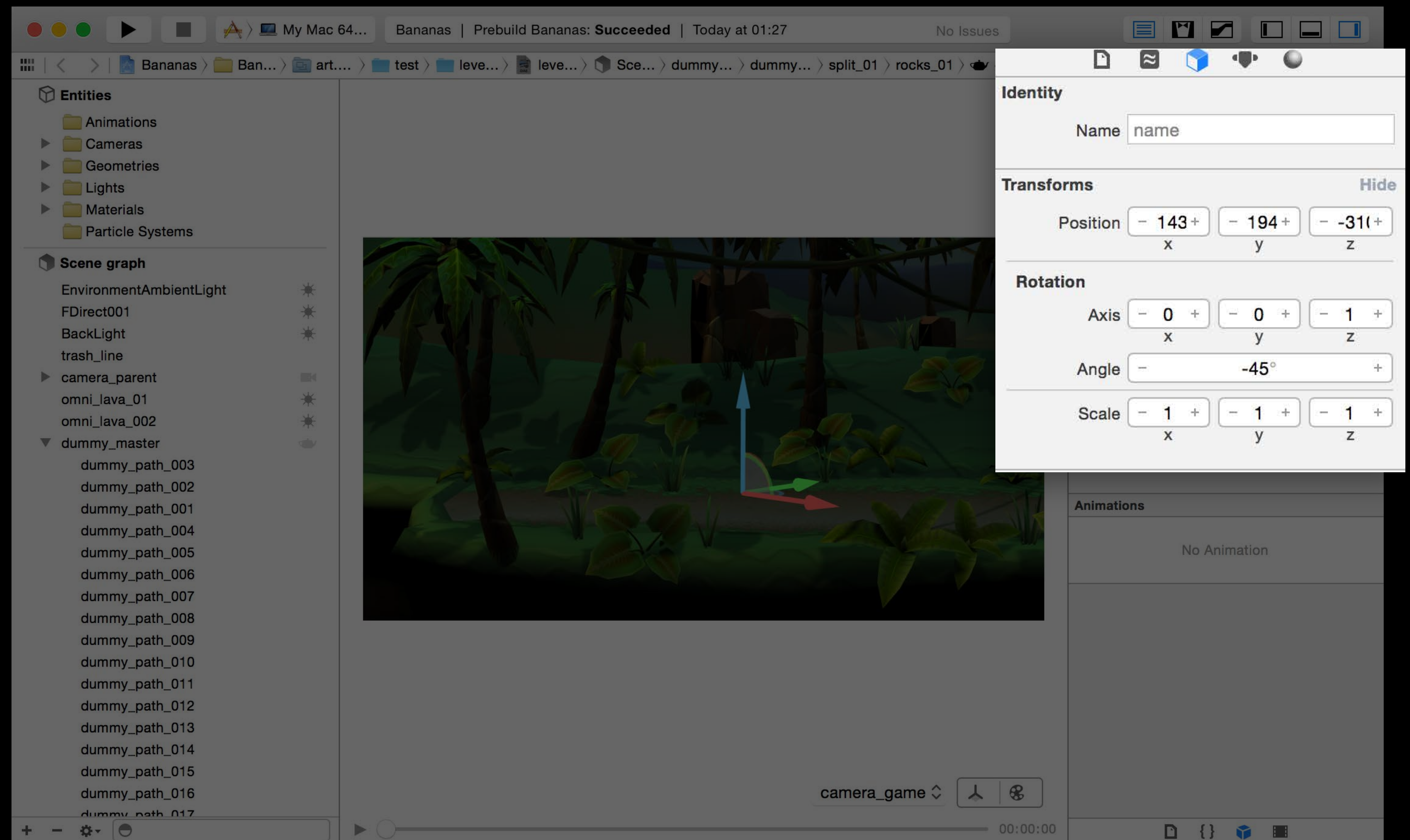
Editing Assets

Scene editor

Direct manipulation

Node hierarchy and entities

Node properties



Editing Assets

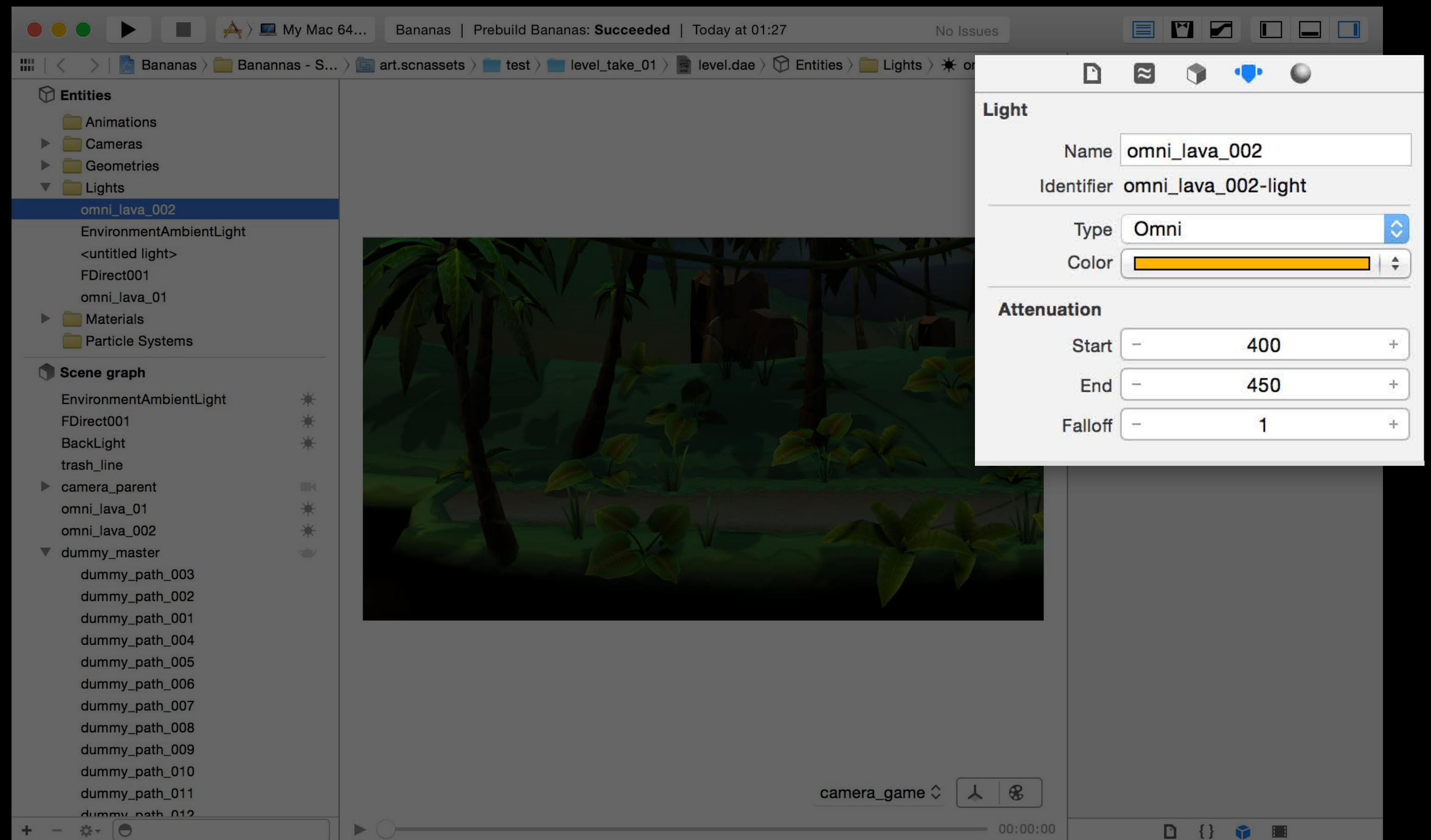
Scene editor

Direct manipulation

Node hierarchy and entities

Node properties

Node attributes



Editing Assets

Scene editor

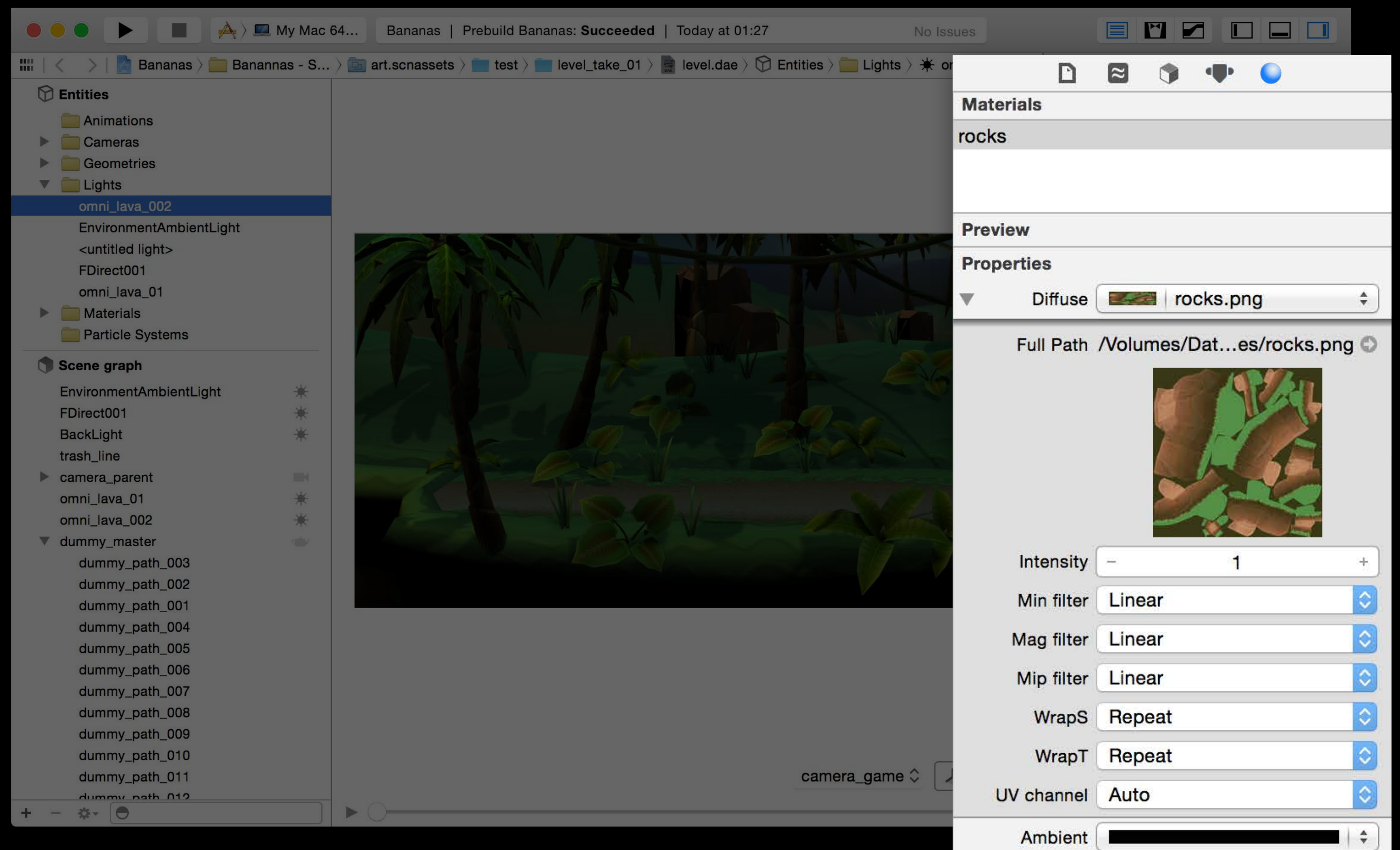
Direct manipulation

Node hierarchy and entities

Node properties

Node attributes

Materials

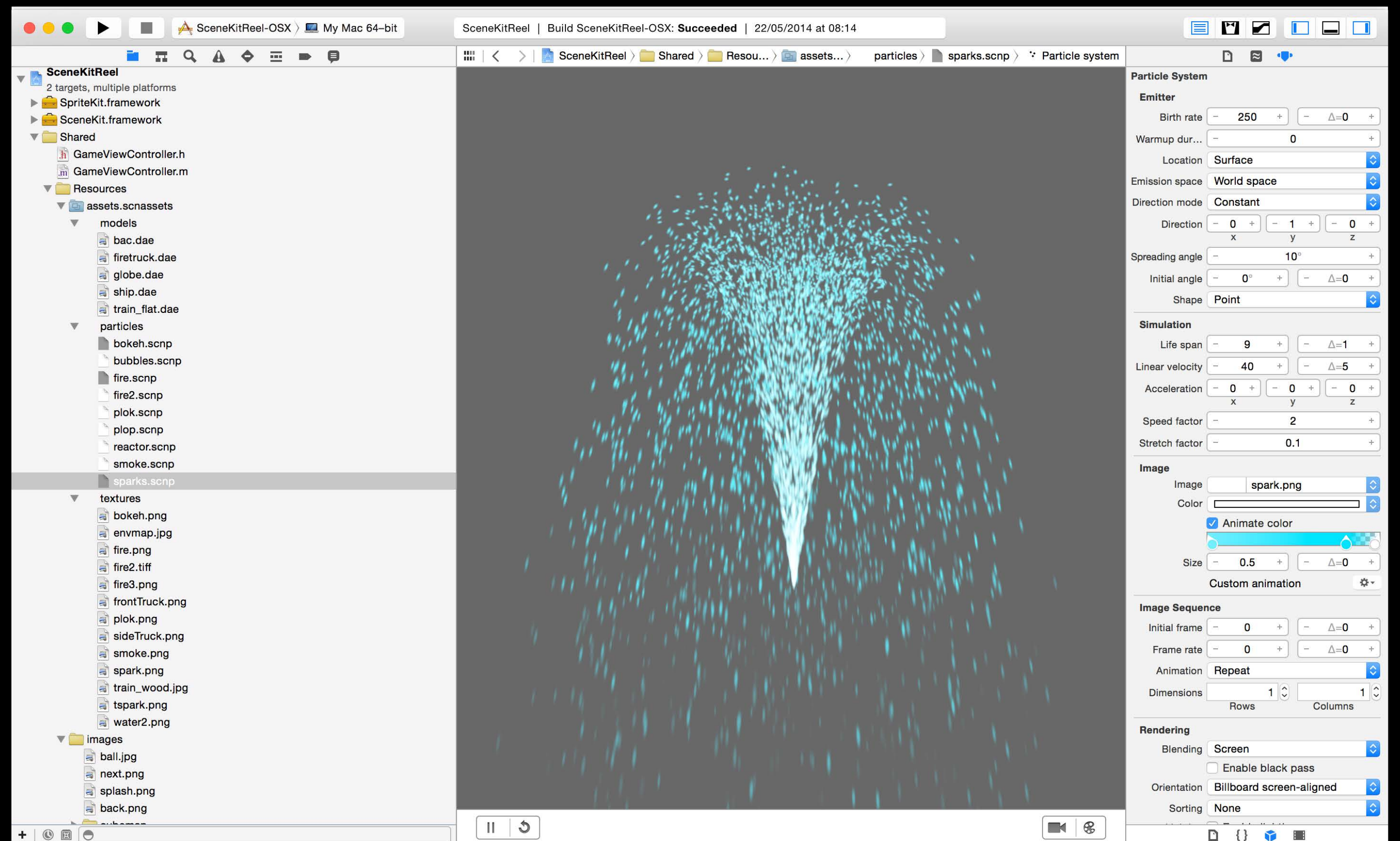


Editing Assets

Particle system editor

Preview and edit

Particle system properties

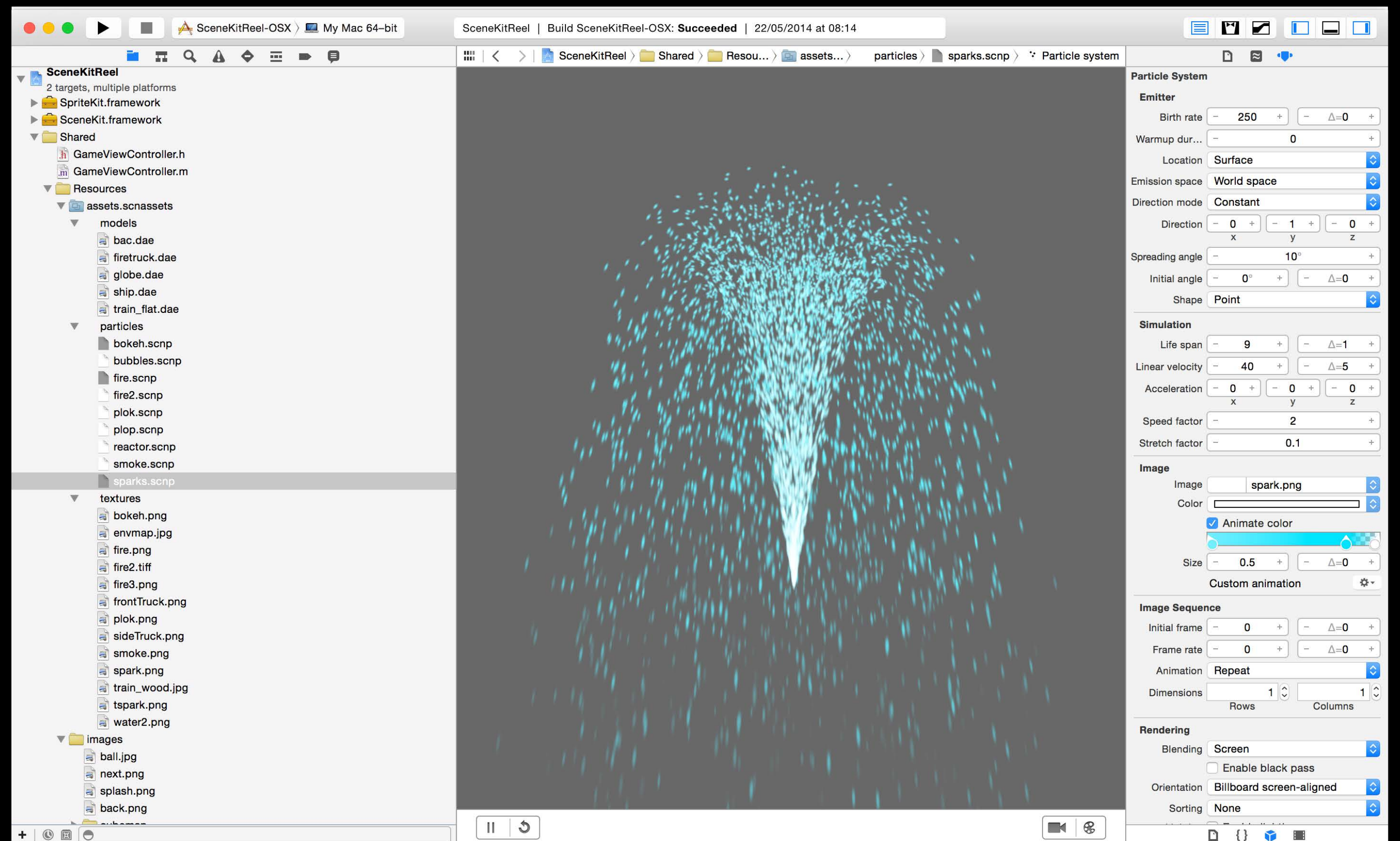


Editing Assets

Particle system editor

Preview and edit

Particle system properties

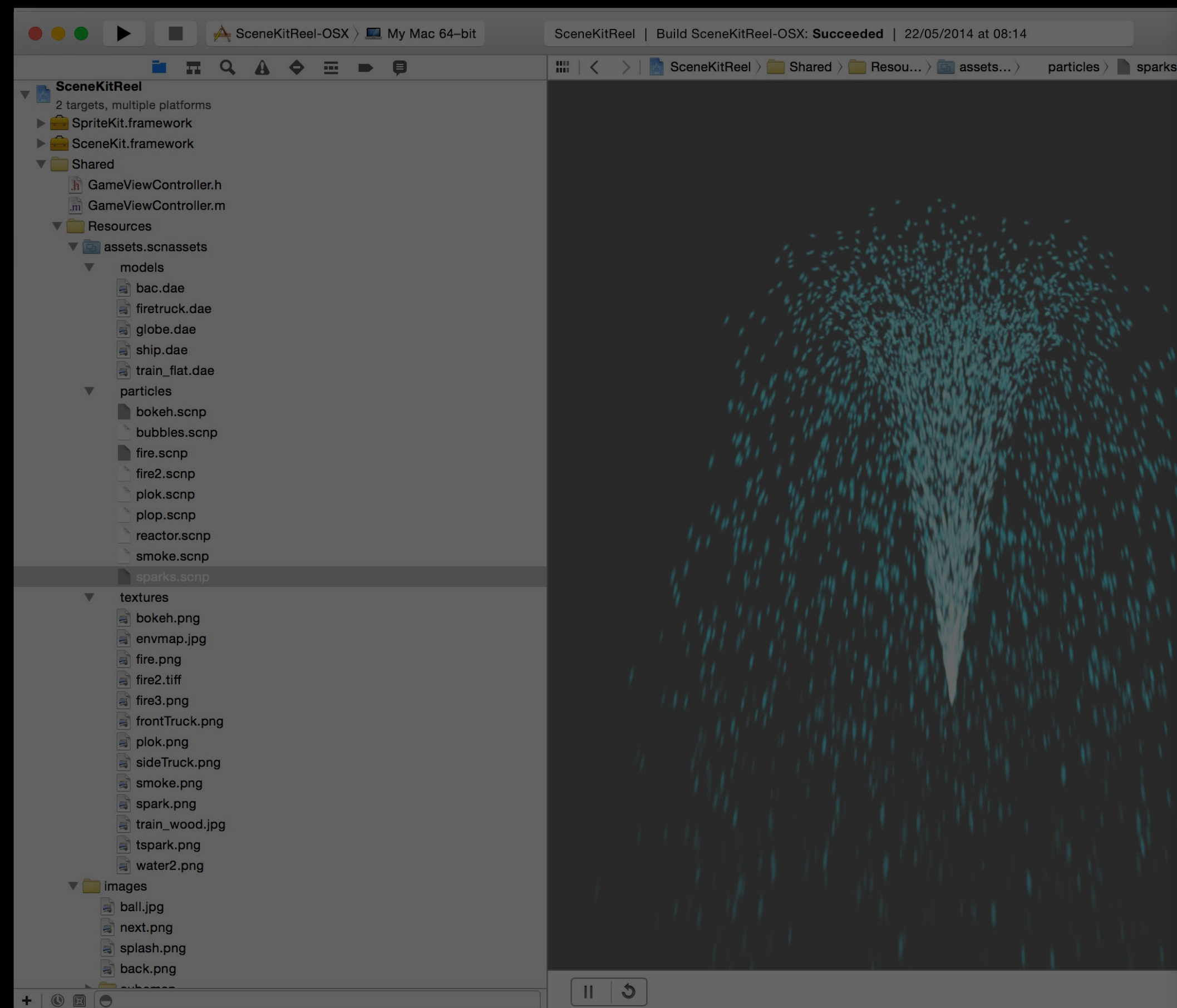


Editing Assets

Particle system editor

Preview and edit

Particle system properties



Particle System

Emitter

Birth rate

Warmup dur...

Location

Emission space

Direction mode

Direction
x y z

Spreading angle

Initial angle

Shape

Simulation

Life span

Linear velocity

Acceleration
x y z

Speed factor

Stretch factor

Image

Image

Color

Animate color

Size

Custom animation

Image Sequence

Initial frame

Frame rate

Animation

Dimensions
Rows Columns

Rendering

Blending

Enable black pass

Orientation

Sorting

Case Study

Bananas

Bananas

Sample code

Illustrates

- Animation
- Lighting and shadows
- Physics
- Particles
- Advanced rendering

Demo

Bananas

Bananas

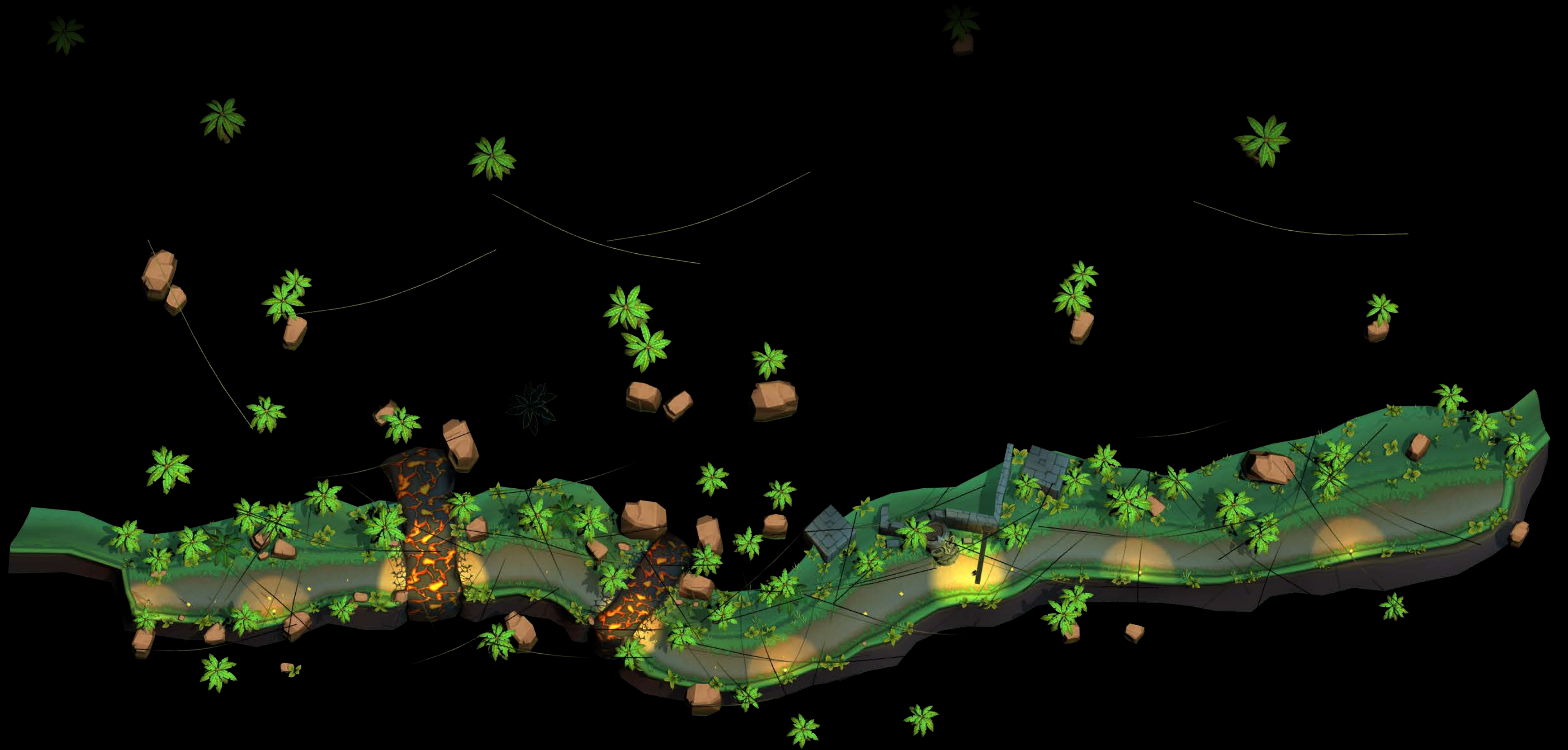
Code base

All Objective-C code

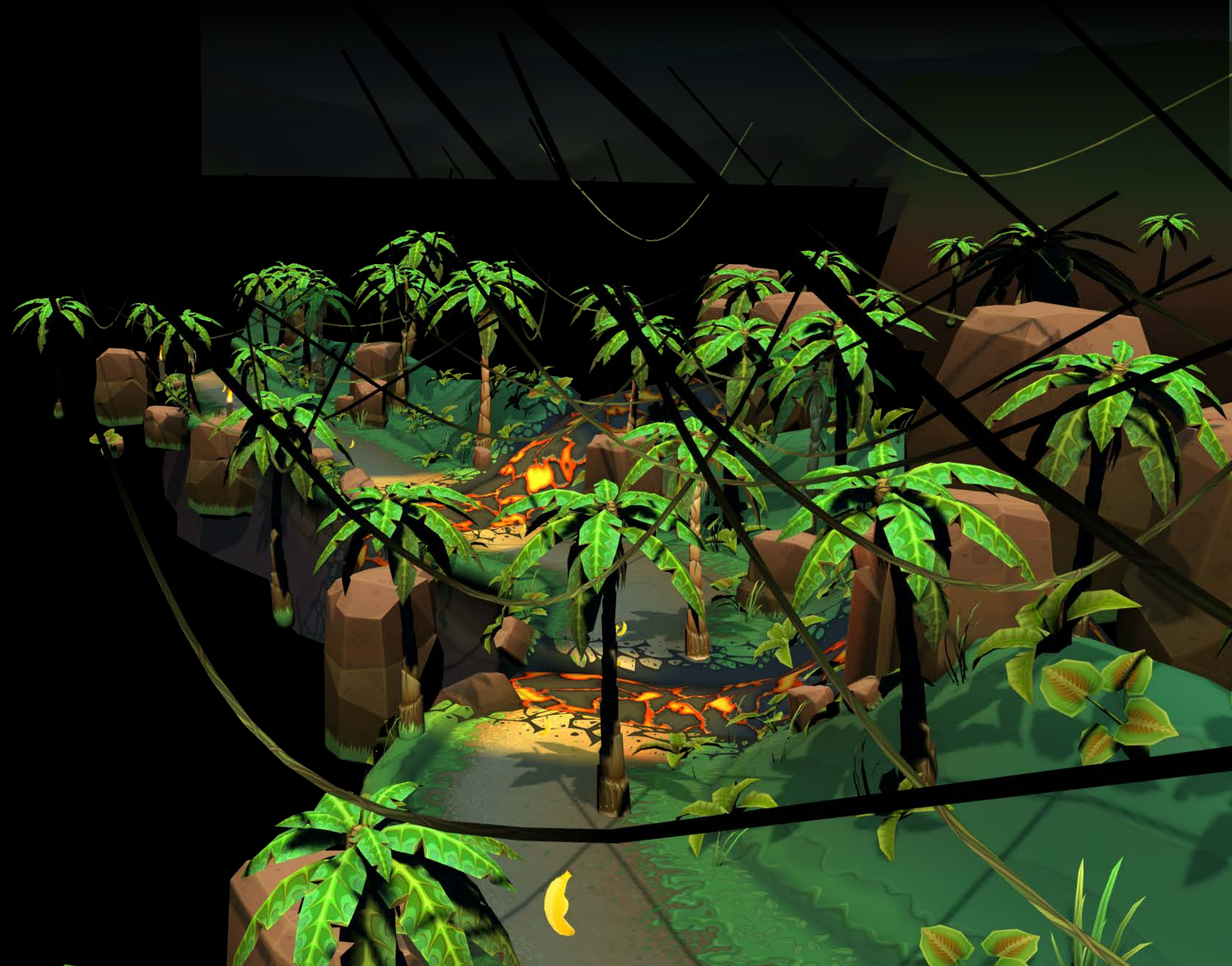
Runs on iOS and OS X

~2700 lines of code

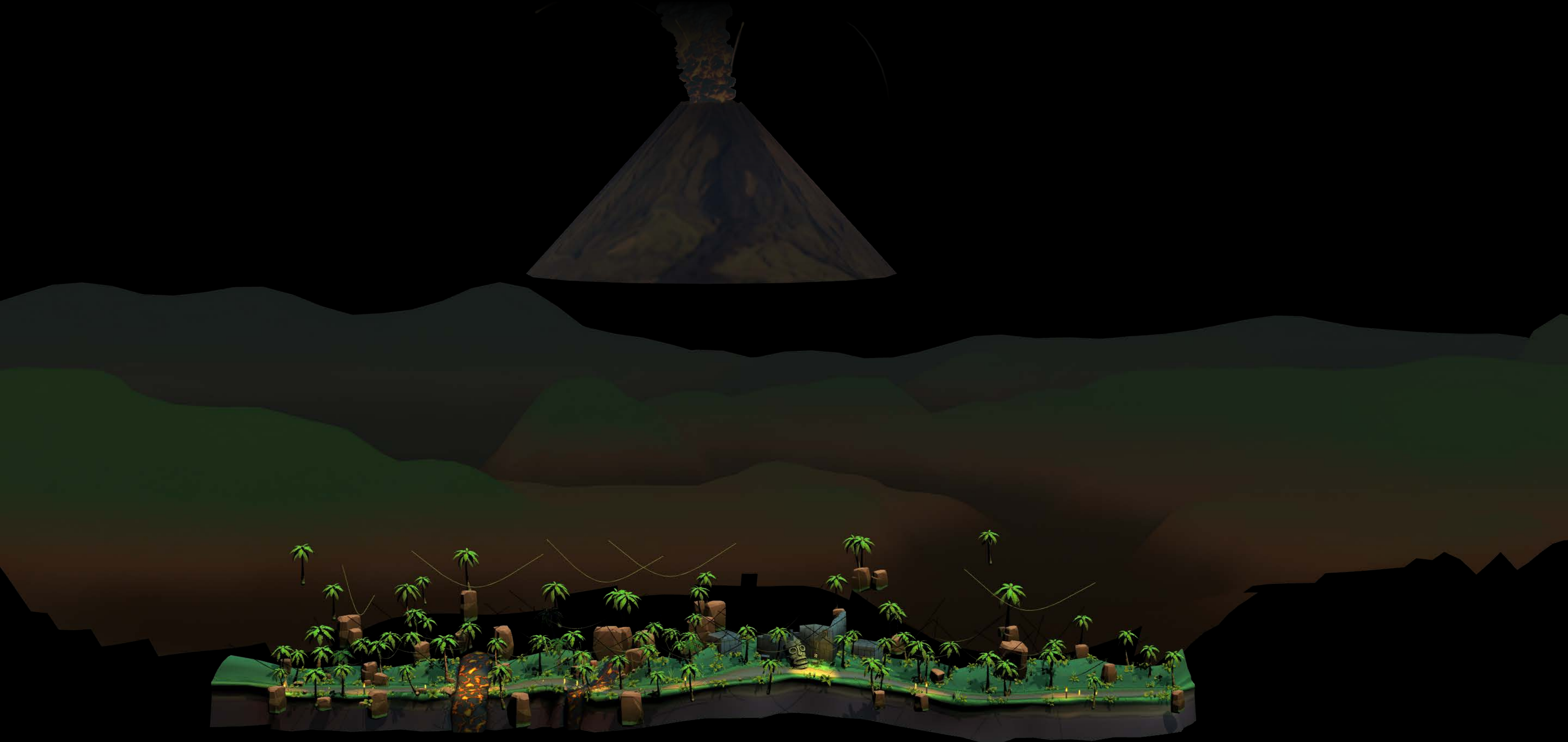
Level Design



Level Design



Level Design



Level Design

Statistics

Features

Details

Lights

10 lights in the world

Lights per object

3 lights (1 omni and 2 directional)

Polygons

200k polygons in the world

Polygons per frame

80k at most

Draw calls

50 at most

Textures

68M

Assets

Hierarchy

Asset catalogs

~25 3D documents

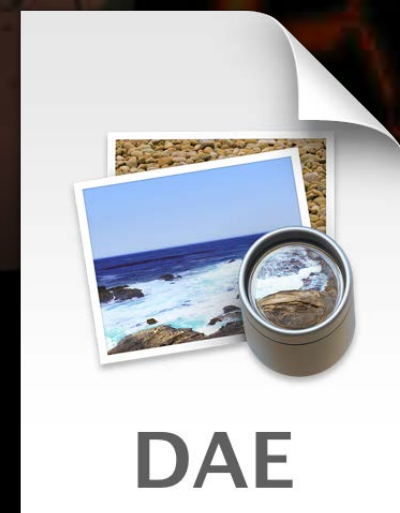
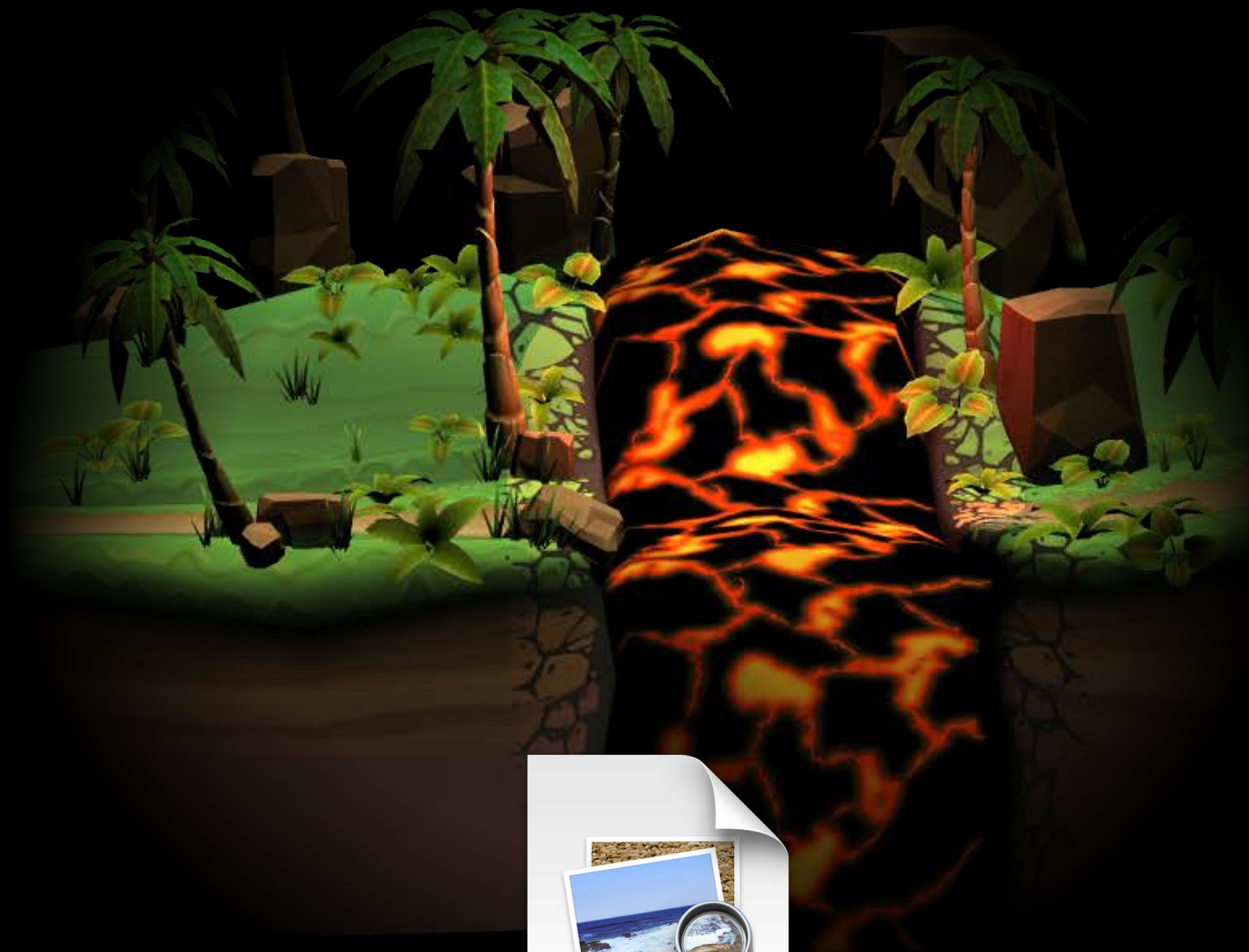
- Animations
- Models
- Textures
- Particle systems



game.scnassets

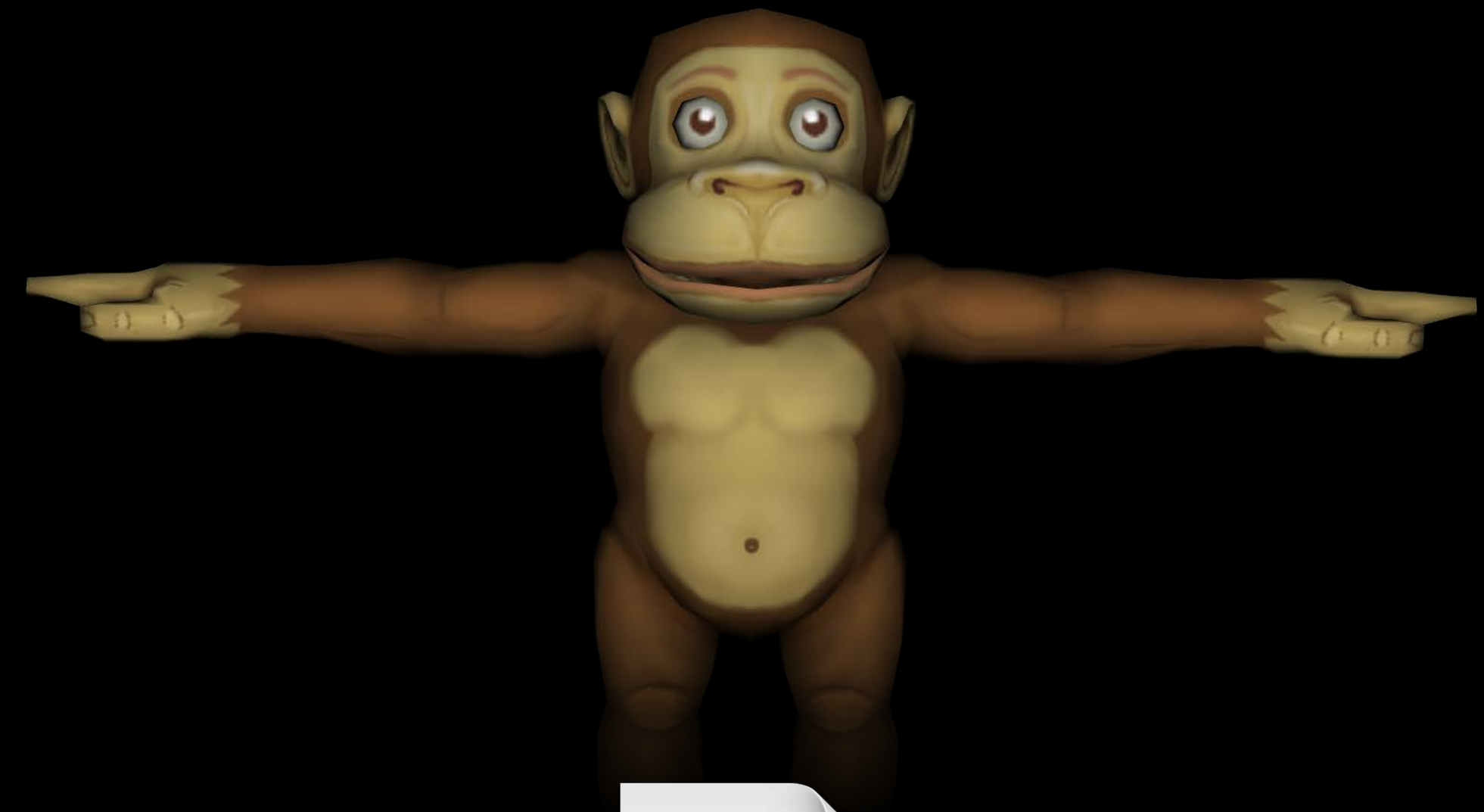
Managing Assets

Combining assets



Scene A

+



Scene B

Managing Assets

Combining assets

```
// Load two scenes  
SCNScene *mainScene = [SCNScene sceneNamed:@"level.dae"];  
SCNScene *characterScene = [SCNScene sceneNamed:@"monkey.dae"];
```

Managing Assets

Combining assets



```
// Load two scenes
SCNScene *mainScene = [SCNScene sceneNamed:@"level.dae"];
SCNScene *characterScene = [SCNScene sceneNamed:@"monkey.dae"];

// Merge them
[mainScene.rootNode addChildNode:characterScene.rootNode];
```


Managing Assets

Combining assets



```
// Load two scenes
SCNScene *mainScene = [SCNScene sceneNamed:@"level.dae"];
SCNScene *characterScene = [SCNScene sceneNamed:@"monkey.dae"];

// Not allowed
[mainScene.rootNode addChildNode:characterScene.rootNode];
```

Managing Assets

Combining assets



```
// Load two scenes
SCNScene *mainScene = [SCNScene sceneNamed:@"level.dae"];
SCNScene *characterScene = [SCNScene sceneNamed:@"monkey.dae"];

// Get the monkey
SCNNode *monkey = [characterScene.rootNode childNodeWithName:@"monkey"
                                                    recursively:YES];

// Add a monkey to the level
[mainScene.rootNode addChildNode:monkey];
```


Managing Assets

Combining assets



```
// Load two scenes
SCNScene *mainScene = [SCNScene sceneNamed:@"level.dae"];
SCNScene *characterScene = [SCNScene sceneNamed:@"monkey.dae"];

// Get the monkey
SCNNode *monkey = [characterScene.rootNode childNodeWithName:@"monkey"
                                                    recursively:YES];

// Add a monkey to the level
[mainScene.rootNode addChildNode:monkey];

// Add another monkey to the level
[mainScene.rootNode addChildNode:[monkey clone]];
```

Controls

iOS

Support for game controllers

Custom “D-pad” gesture recognizer

- Three buttons
 - Jump
 - Left, right (with long press state)

Controls

OS X

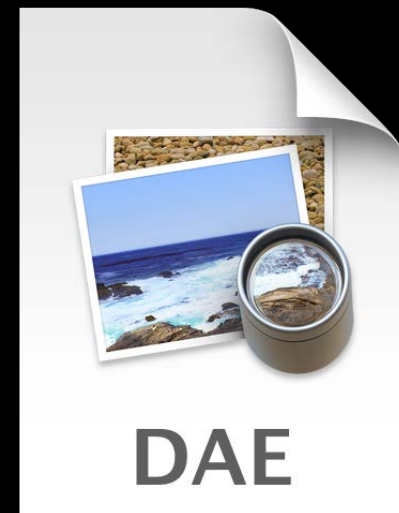
Support for game controllers

Keyboard events

- Jump
- Left, right (with long press state)

Behaviors

Animating characters

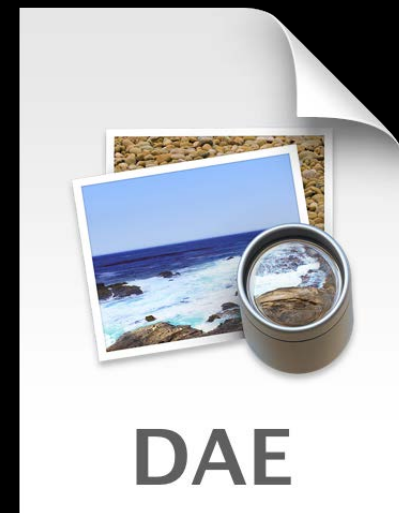


skinned-character.dae



Behaviors

Animating characters



skinned-character.dae



run.dae



jump.dae



idle.dae



Behaviors

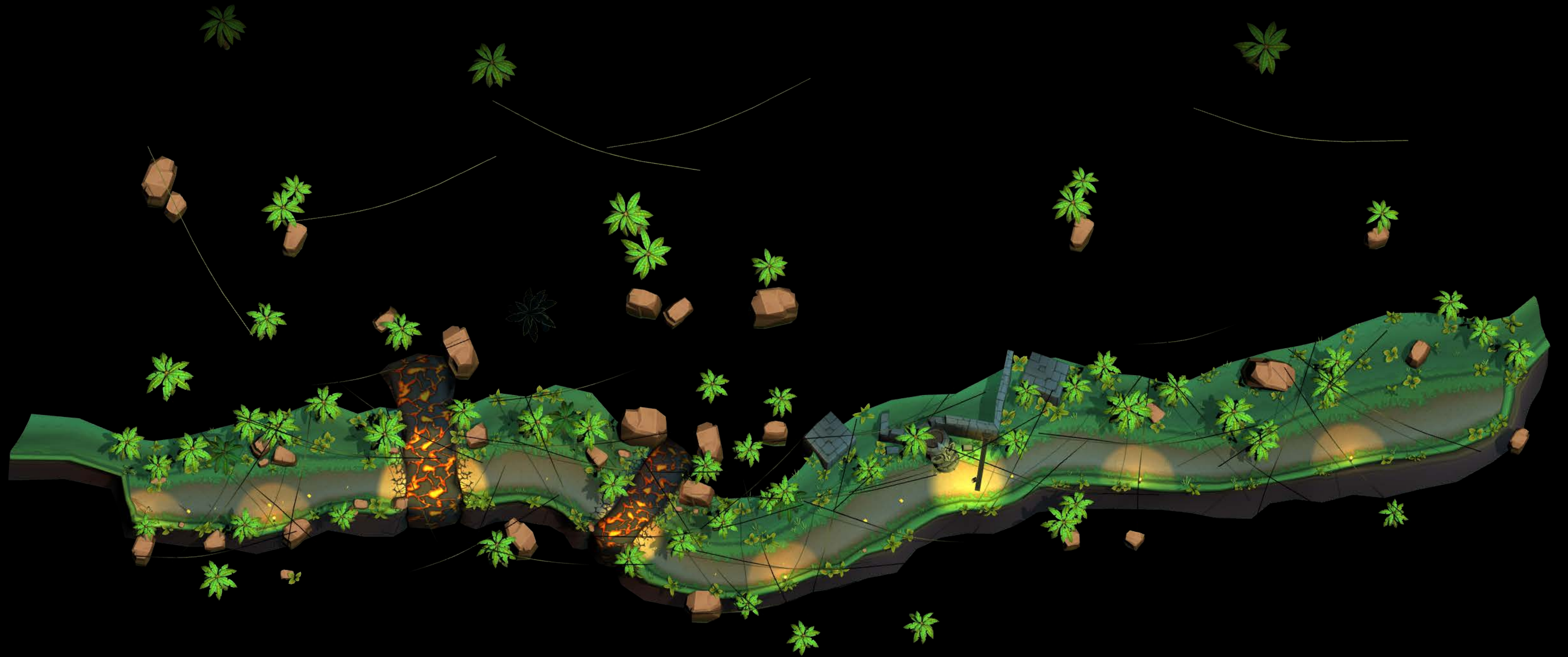
Animating characters

```
// Load an animation
CAAnimation *anim = [sceneSource entryWithIdentifier:animationName
                    withClass:[CAAnimation class]];

// Play it
[character addAnimation:anim
         forKey:@"run"];
```

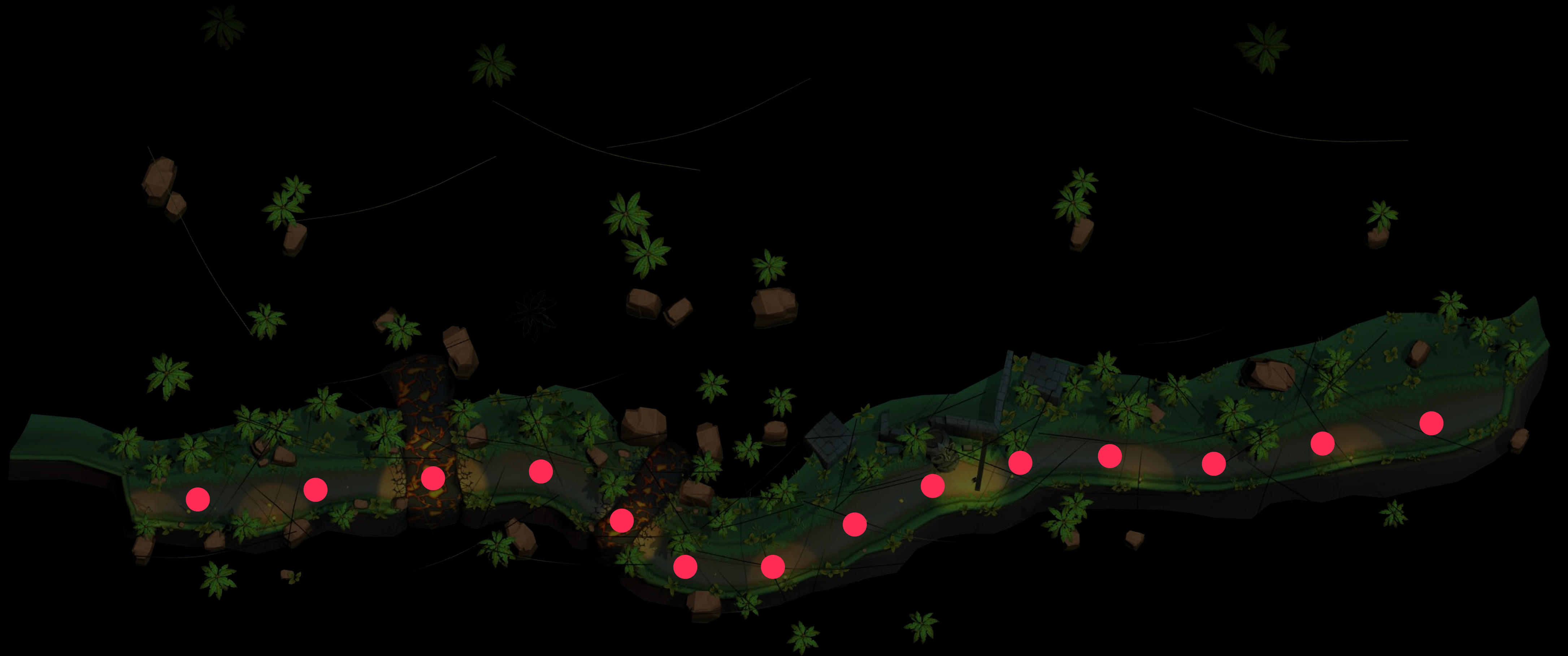
Behaviors

Moving characters



Behaviors

Moving characters



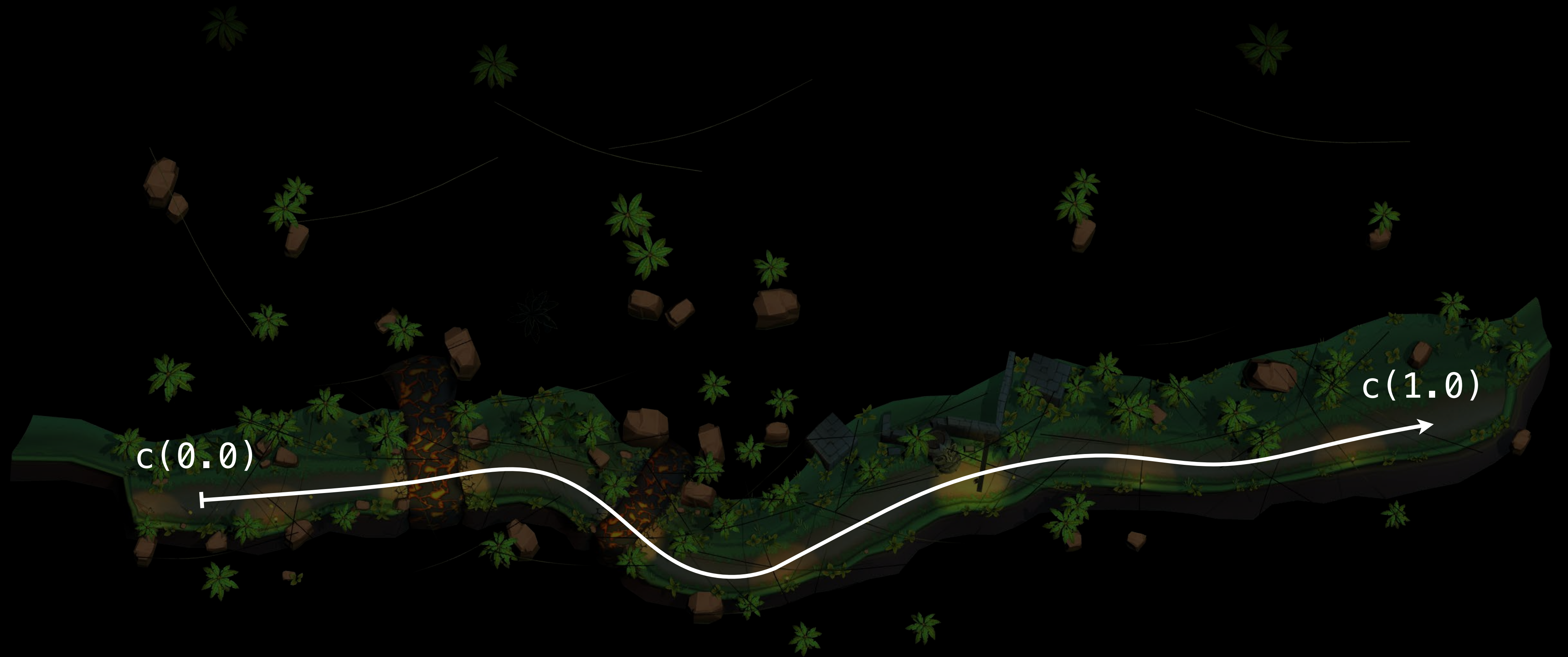
Behaviors

Moving characters



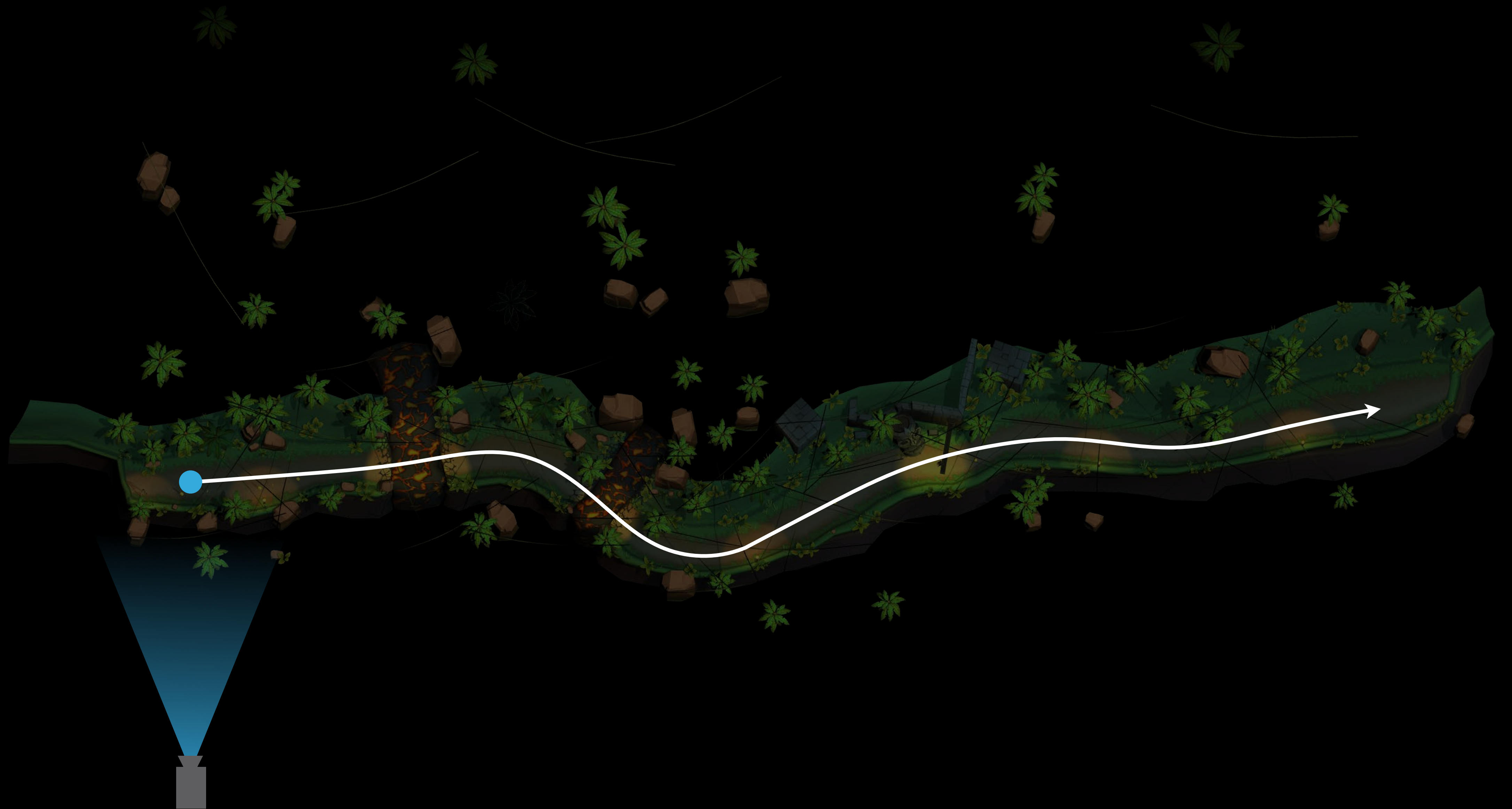
Behaviors

Moving the camera



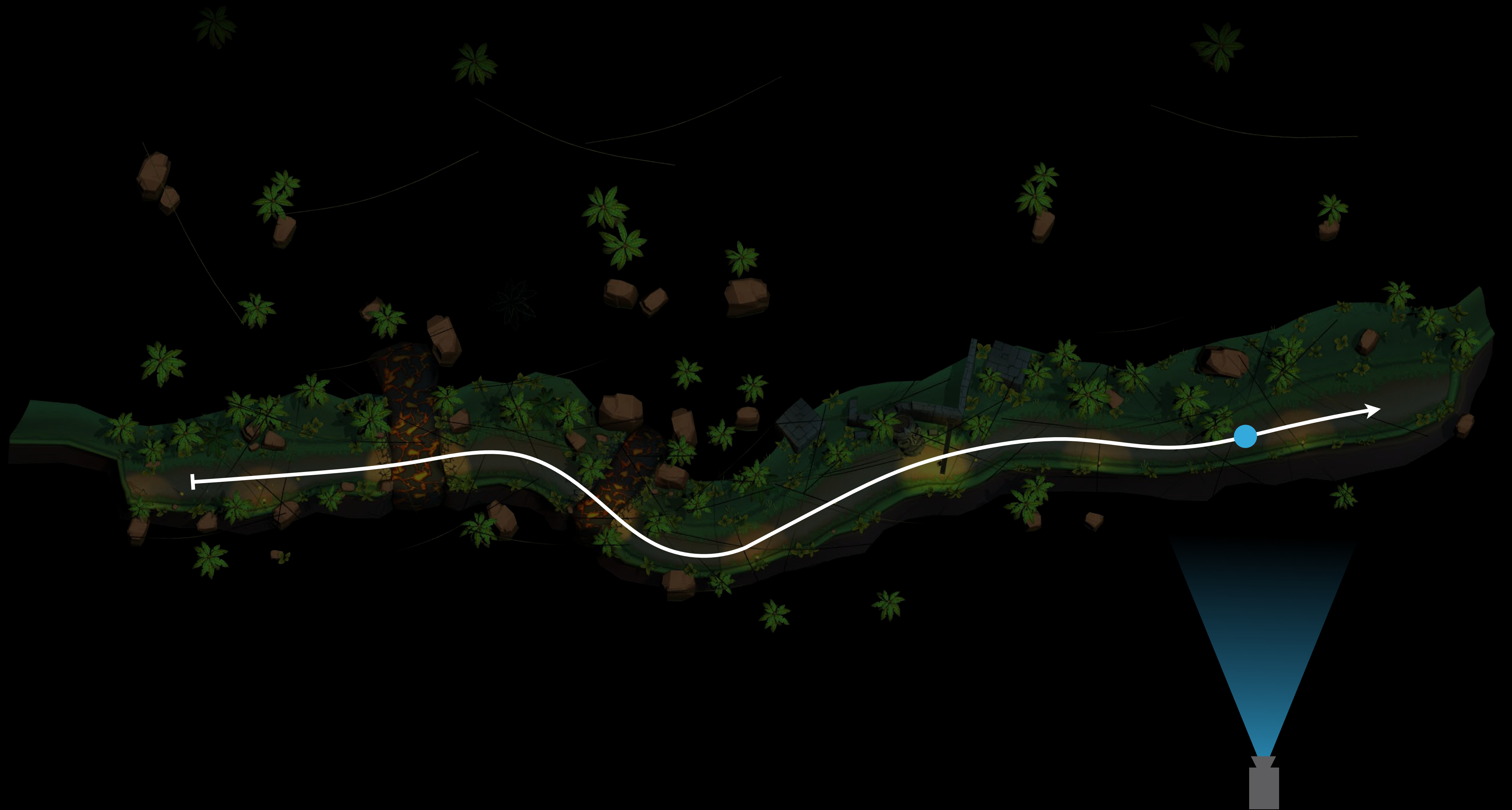
Behaviors

Moving the camera



Behaviors

Moving the camera



Collision Detection

Stay on the path

Avoid coconuts

Collect bananas

Jump over lava



Collision Detection

SCNPhysicsWorld

A white rounded square containing the word "NEW" in a colorful, outlined font.

```
// Be notified through delegation
- (void)physicsWorld:(SCNPhysicsWorld *)world
    didBeginContact:(SCNPhysicsContact *)contact;

// Or explicitly perform ray tests
- (NSArray *)rayTestWithSegmentFromPoint:(SCNVector3)origin
    toPoint:(SCNVector3)dest
    options:(NSDictionary *)options;
```

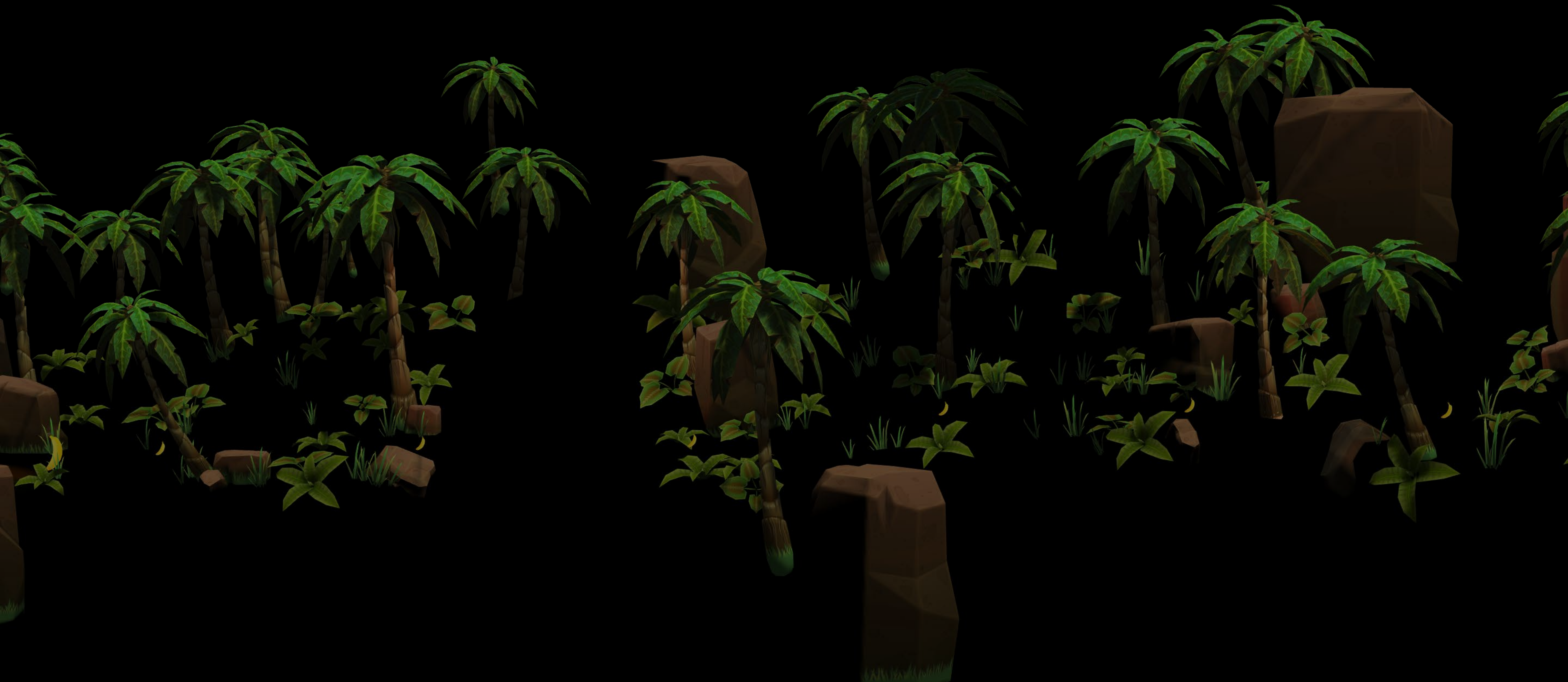
Behaviors

Animating items

A white rounded square containing the word "NEW" in a colorful, outlined font.

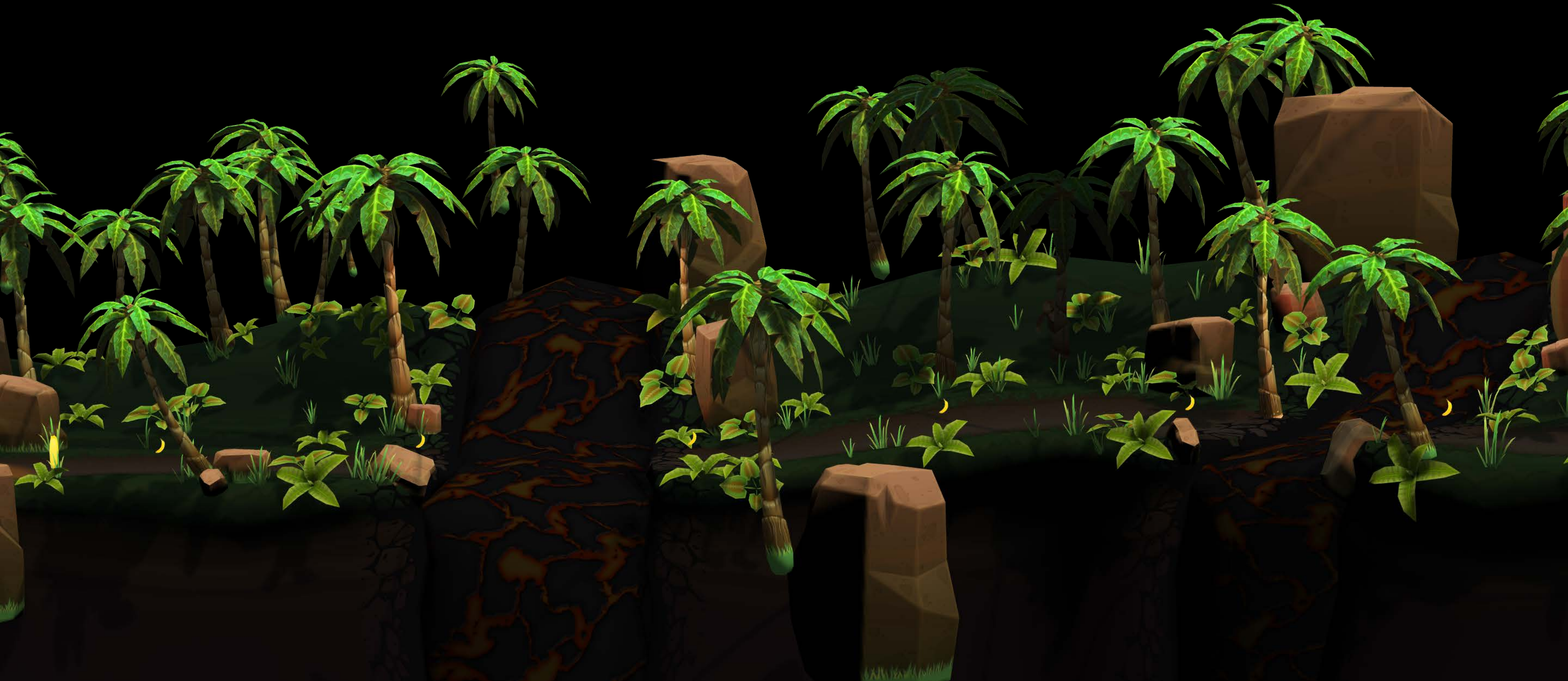
```
// Animate a banana
[aBanana runAction:
  [SCNAction repeatActionForever:
    [SCNAction rotateByX:0.0
      y:2.0 * M_PI
      z:0.0
      duration:2.0]]];
```


Lighting



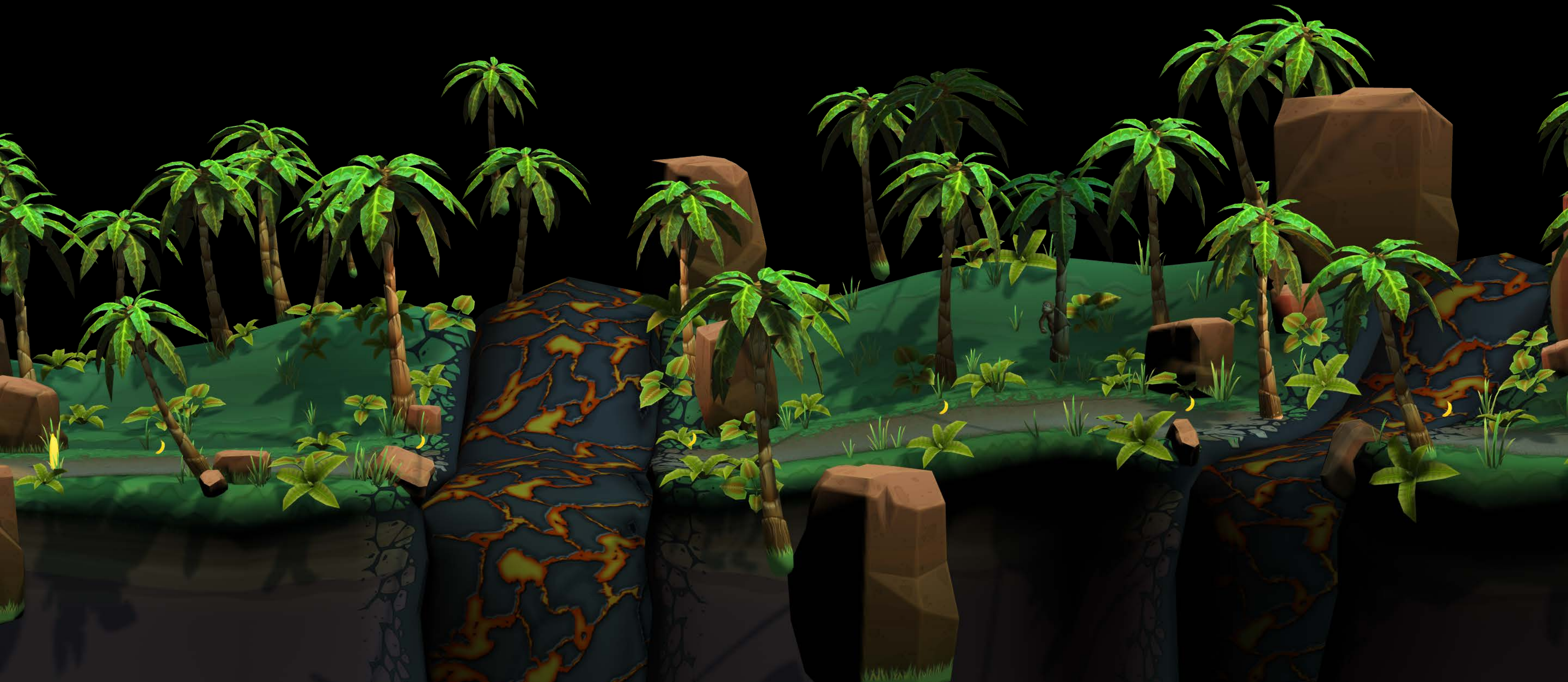
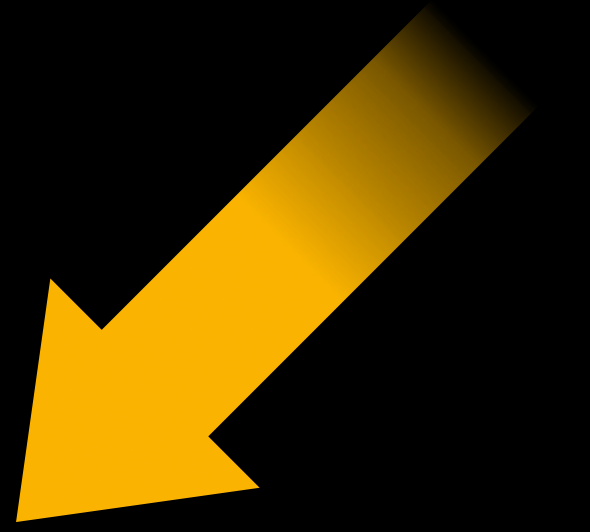
Lighting

Ambient



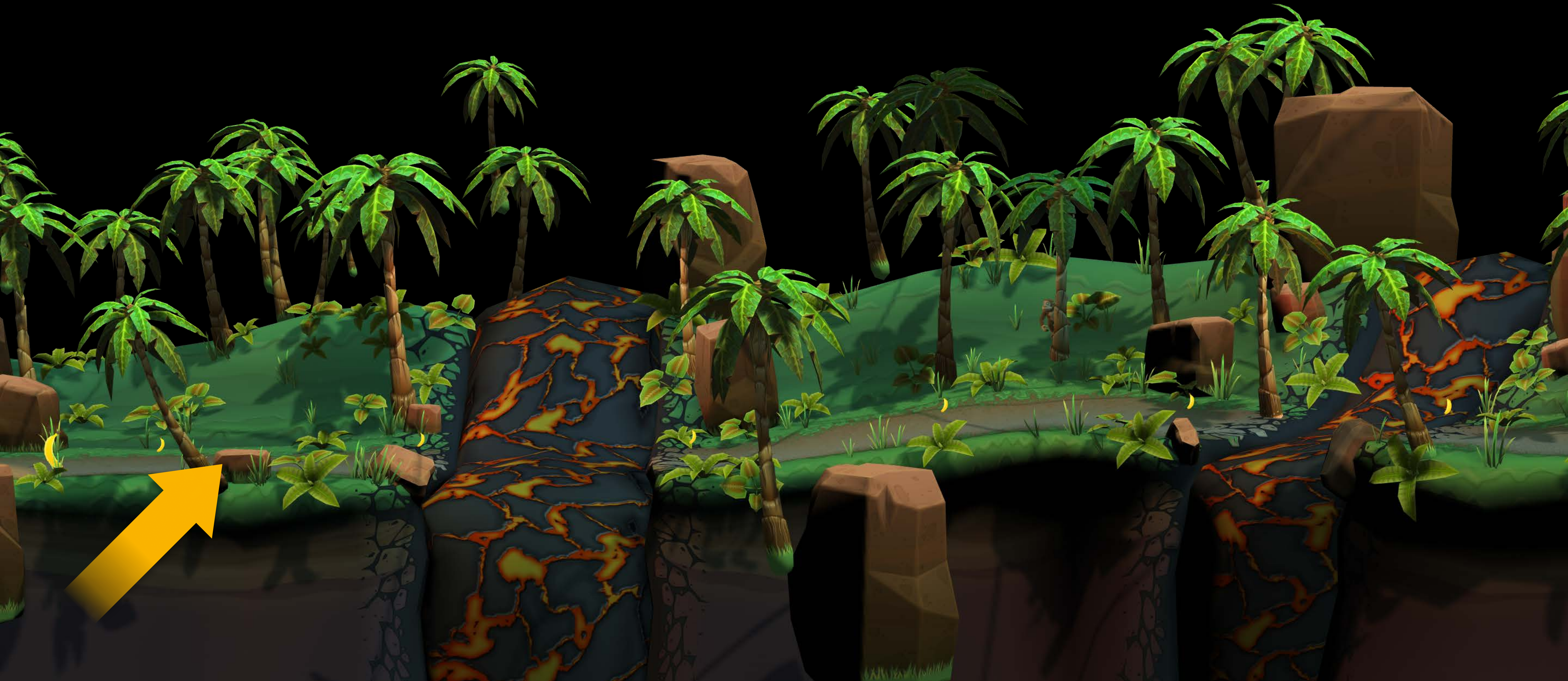
Lighting

Ambient



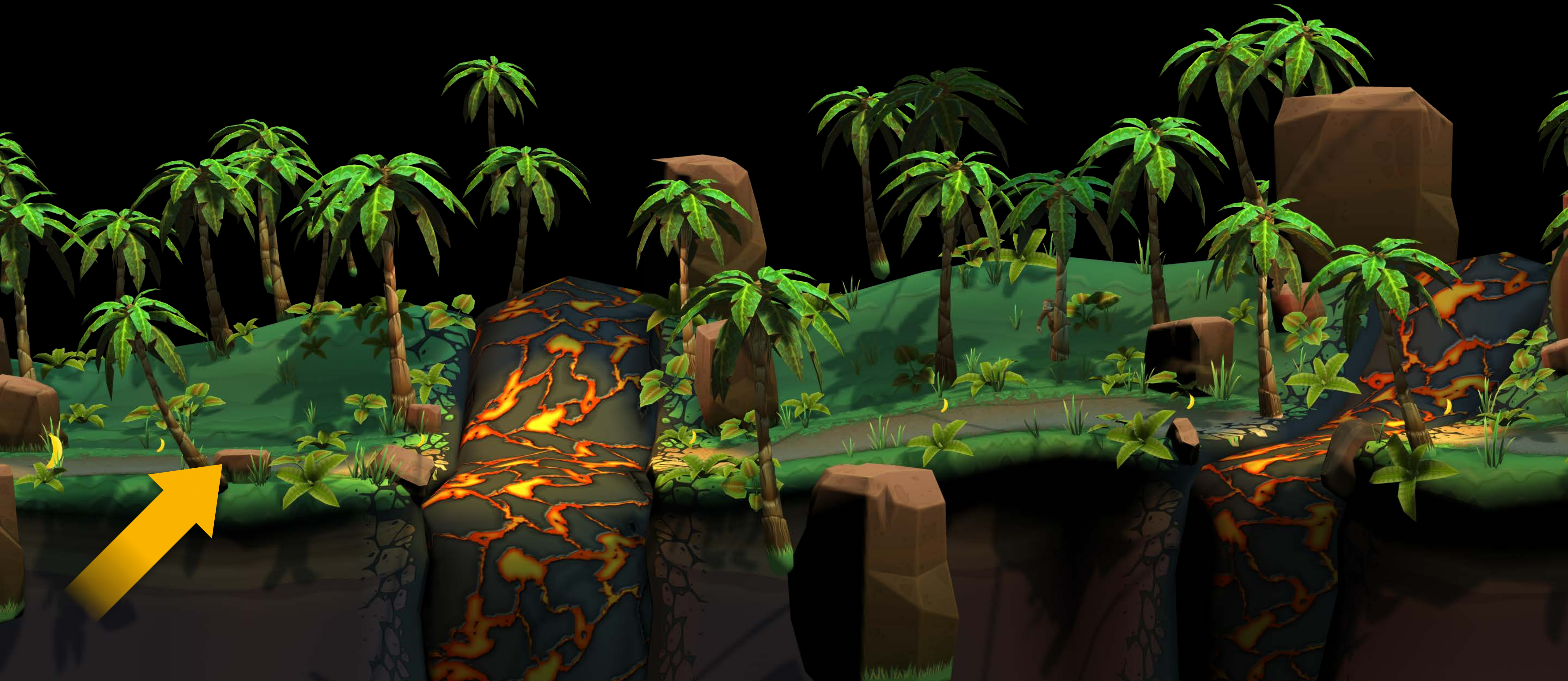
Lighting

Ambient



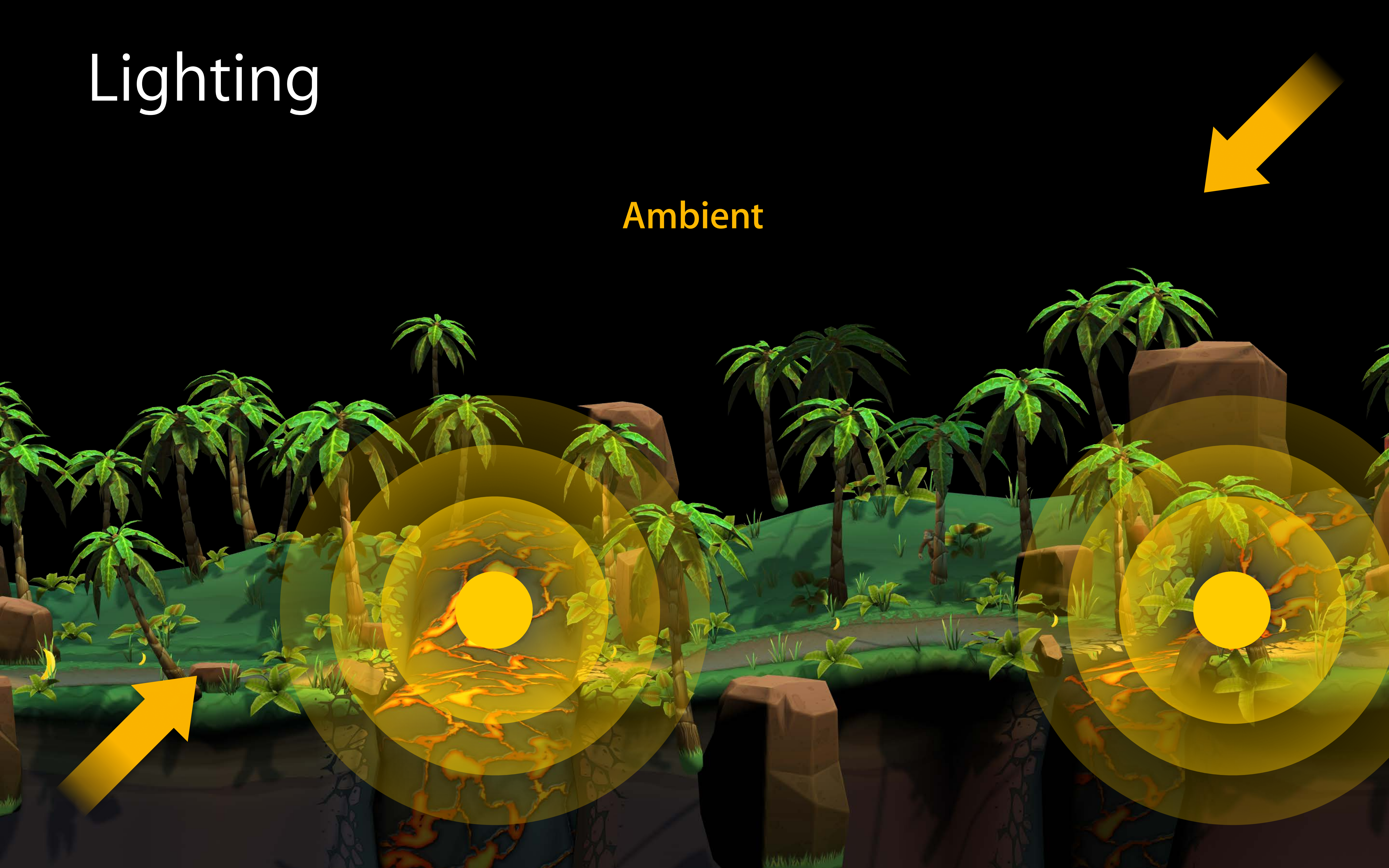
Lighting

Ambient



Lighting

Ambient



Shadows

Static shadows



Shadows

Static shadows



Shadows

Static shadows

Baked into textures

Don't adapt to changes

```
material.multiply.contents = shadowMap;
```



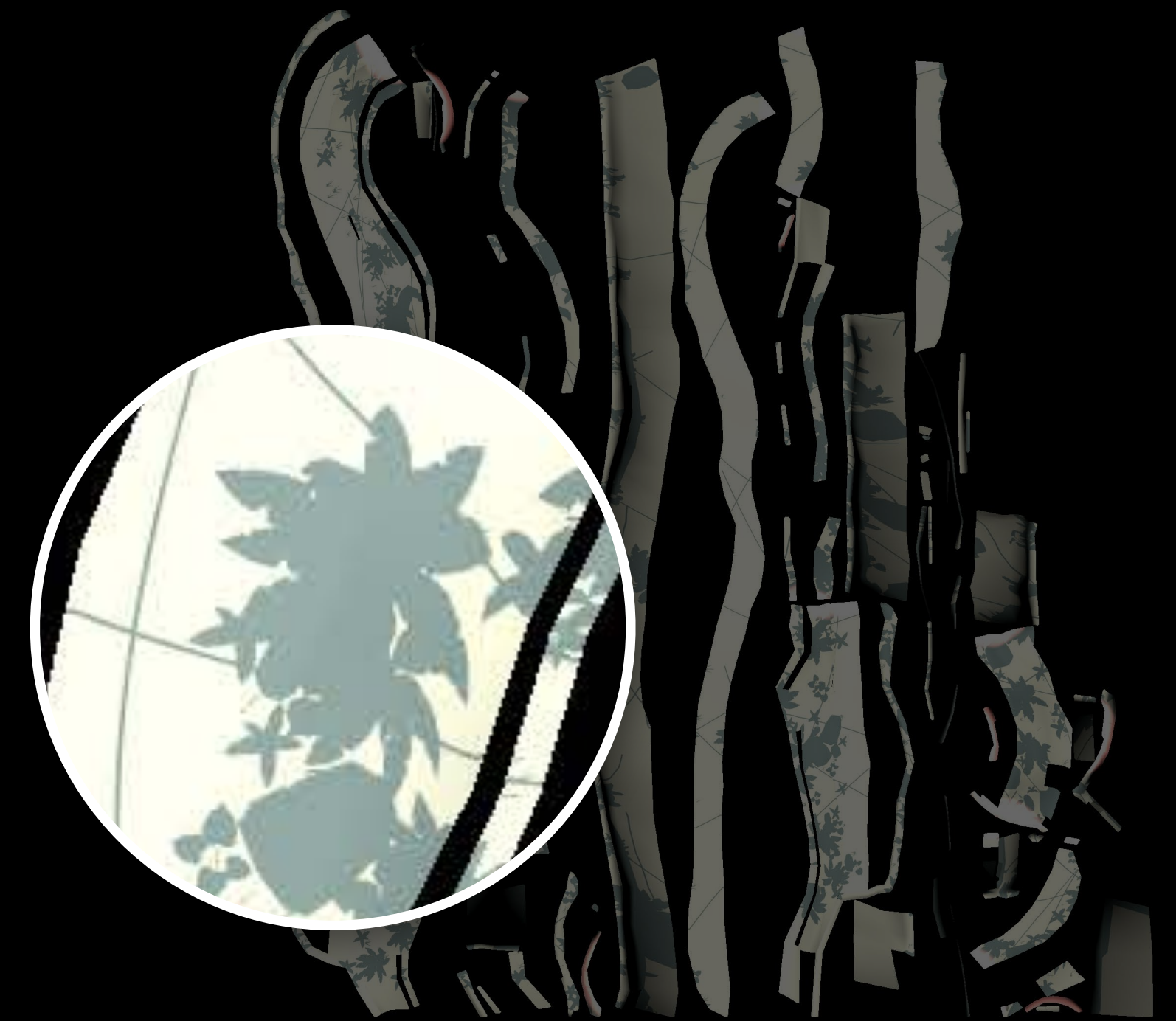
Shadows

Static shadows

Baked into textures

Don't adapt to changes

```
material.multiply.contents = shadowMap;
```



Shadows

Dynamic shadows



Shadows

Dynamic shadows



Shadows

Dynamic shadows

Shadow maps

Real-time shadows

Suitable for animated objects



Shadows

Dynamic shadows

Shadow maps

Real-time shadows

Suitable for animated objects

```
// Make a light cast shadows  
light.castsShadow = YES;
```



Shadows

Mixing techniques



```
// Make the light cast shadows
```

```
aLight.castsShadow = YES;
```

```
// Exclude nodes that use baked shadows
```

```
aNode.castsShadow = NO;
```

```
// We can also use bit masks (with kMainLightMask & kSecondaryLightMask = 0)
```

```
aLight.categoryBitMask = kMainLightMask;
```

```
aNode.categoryBitMask = kSecondaryLightMask;
```

Shadows

Mixing techniques



```
// Make the light cast shadows  
aLight.castsShadow = YES;
```

```
// Exclude nodes that use baked shadows  
aNode.castsShadow = NO;
```

```
// We can also use bit masks (with kMainLightMask & kSecondaryLightMask = 0)  
aLight.categoryBitMask = kMainLightMask;  
aNode.categoryBitMask = kSecondaryLightMask;
```


Shadows

Mixing techniques

A white rounded square containing the word "NEW" in a colorful, outlined font.

```
// Make the light cast shadows  
aLight.castsShadow = YES;
```

```
// Exclude nodes that use baked shadows  
aNode.castsShadow = NO;
```

```
// We can also use bit masks (with kMainLightMask & kSecondaryLightMask = 0)  
aLight.categoryBitMask = kMainLightMask;  
aNode.categoryBitMask = kSecondaryLightMask;
```

Shadows

Mixing techniques

A white rounded square containing the word "NEW" in a colorful, outlined font.

```
// Make the light cast shadows  
aLight.castsShadow = YES;
```

```
// Exclude nodes that use baked shadows  
aNode.castsShadow = NO;
```

```
// We can also use bit masks (with kMainLightMask & kSecondaryLightMask = 0)  
aLight.categoryBitMask = kMainLightMask;  
aNode.categoryBitMask = kSecondaryLightMask;
```


Shadows

Mixing techniques



```
// Make the light cast shadows  
aLight.castsShadow = YES;
```

```
// Exclude nodes that use baked shadows  
aNode.castsShadow = NO;
```

```
// We can also use bit masks (with kMainLightMask & kSecondaryLightMask = 0)  
aLight.categoryBitMask = kMainLightMask;  
aNode.categoryBitMask = kSecondaryLightMask;
```

Shadows

Dynamic shadows

Shadow maps

Real-time shadows

Suitable for animated objects



Shadows

Projected shadows



Shadows

Projected shadows



Shadows

Projected shadows

Simplified shadows

Suitable for animated objects

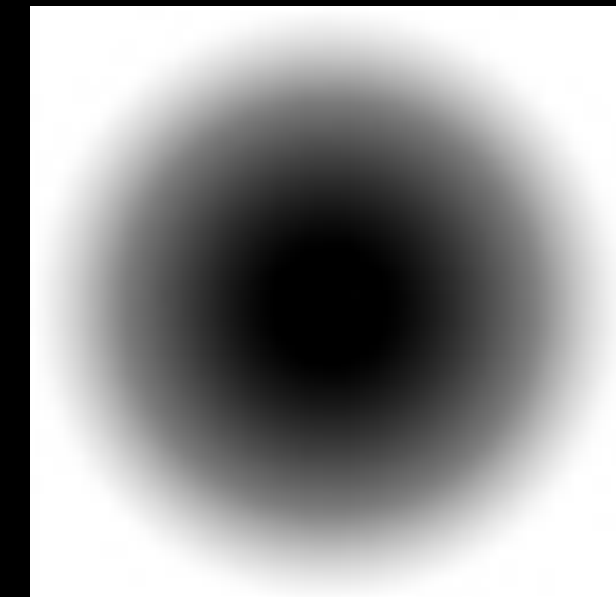


Shadows

NEW

Projected shadows

```
// Use modulated mode  
light.shadowMode = SCNShadowModeModulated;  
  
// Configure the projected shadow  
light.gobo.contents = aShadowImage;  
  
// Use bit masks to specify receivers  
light.categoryBitMask = kProjectorLightMask;  
floor.categoryBitMask = kProjectorLightMask;
```



Particles

SCNParticleSystem

Used for torches and dust



Particles

SCNParticleSystem



```
// Load a particle system
SCNParticleSystem *particleSystem =
    [SCNParticleSystem particleSystemNamed:@"dust.scnp
                                inDirectory:@"art.scnassets/particles"];

// Attach to a node
[character addParticleSystem:particleSystem];
```


Particles

SCNParticleSystem



```
// Load a particle system
SCNParticleSystem *particleSystem =
    [SCNParticleSystem particleSystemNamed:@"dust.scnp
                                inDirectory:@"art.scnassets/particles"];

// Attach to a node
[character addParticleSystem:particleSystem];

// Control emission
particleSystem.birthRate = shouldEmit ? aBirthRate : 0;
```


Visual Improvements

Shader modifiers

Geometry animation

- Vines

Texture animation

- Lava
- Volcano



Visual Improvements

Shader modifiers

```
NSString *modifier = @"_geometry.texcoords[0] +=  
    vec2(  
        sin(_geometry.position.z + u_time) * 0.01,  
        -0.05 * u_time  
    );";
```

```
lavaNode.geometry.shaderModifiers =  
    @{ SCNShaderModifierEntryPointGeometry : modifier};
```

Visual Improvements

Postprocessing

NEW

SCNTechnique

- Color effects
- Image deformation
- Highly customizable



Visual Improvements

Postprocessing

NEW



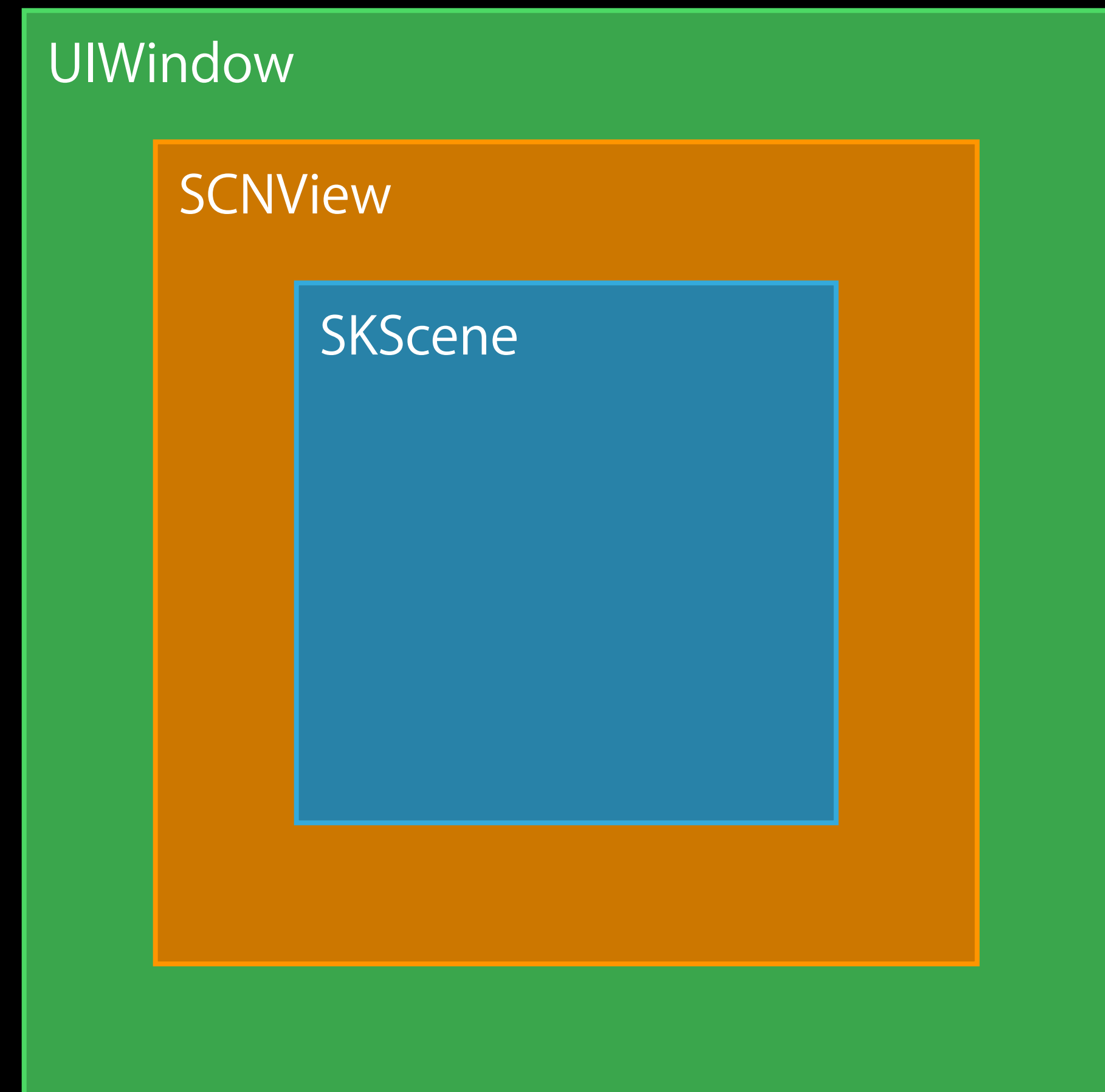
Visual Improvements

SpriteKit overlays

iOS and OS X

SKScene

- Score
- Timer
- Title screen



Visual Improvements

SpriteKit overlays



Sound

SpriteKit API

```
[SKAction playSoundFileNamed:@"jump.caf"  
          waitForCompletion:NO];
```

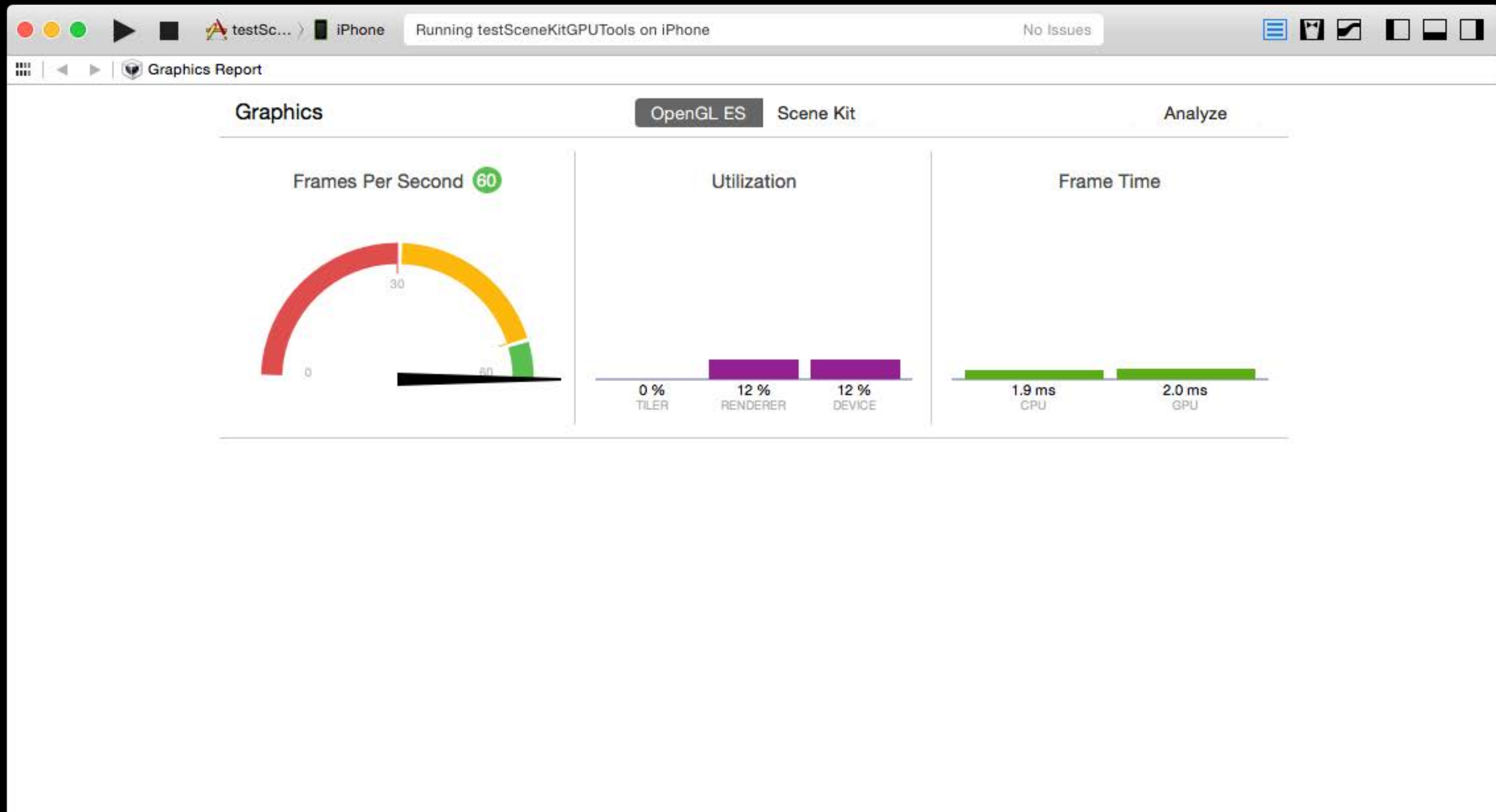

Performance Optimization

Thomas Goossens
Software Engineer

Performance Statistics

Xcode report

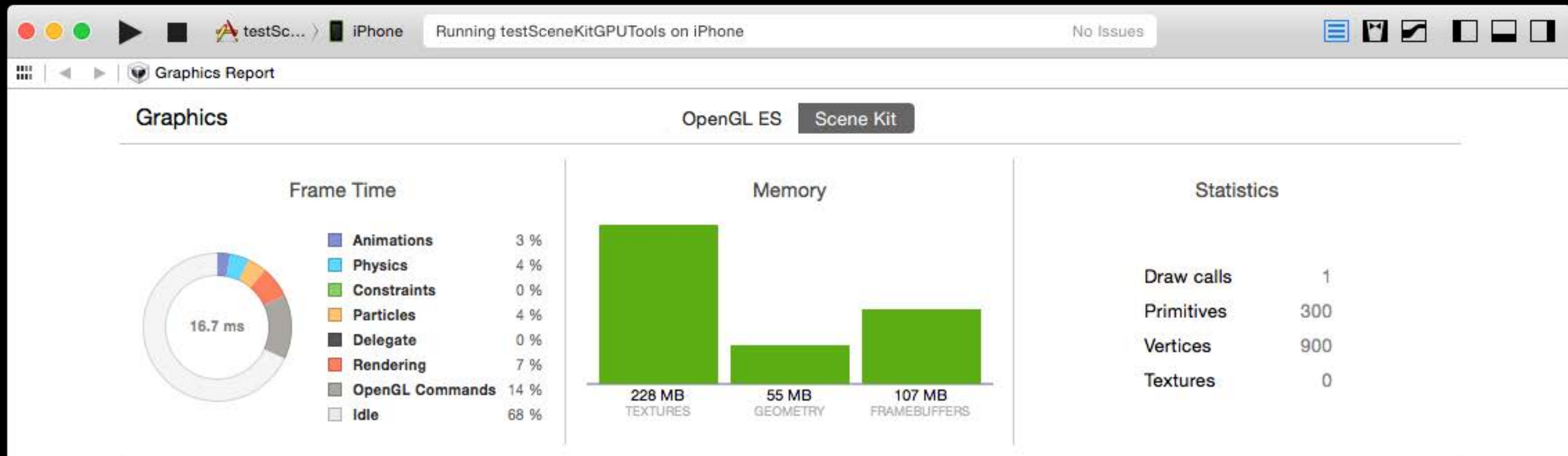
NEW



Performance Statistics

Xcode report

NEW

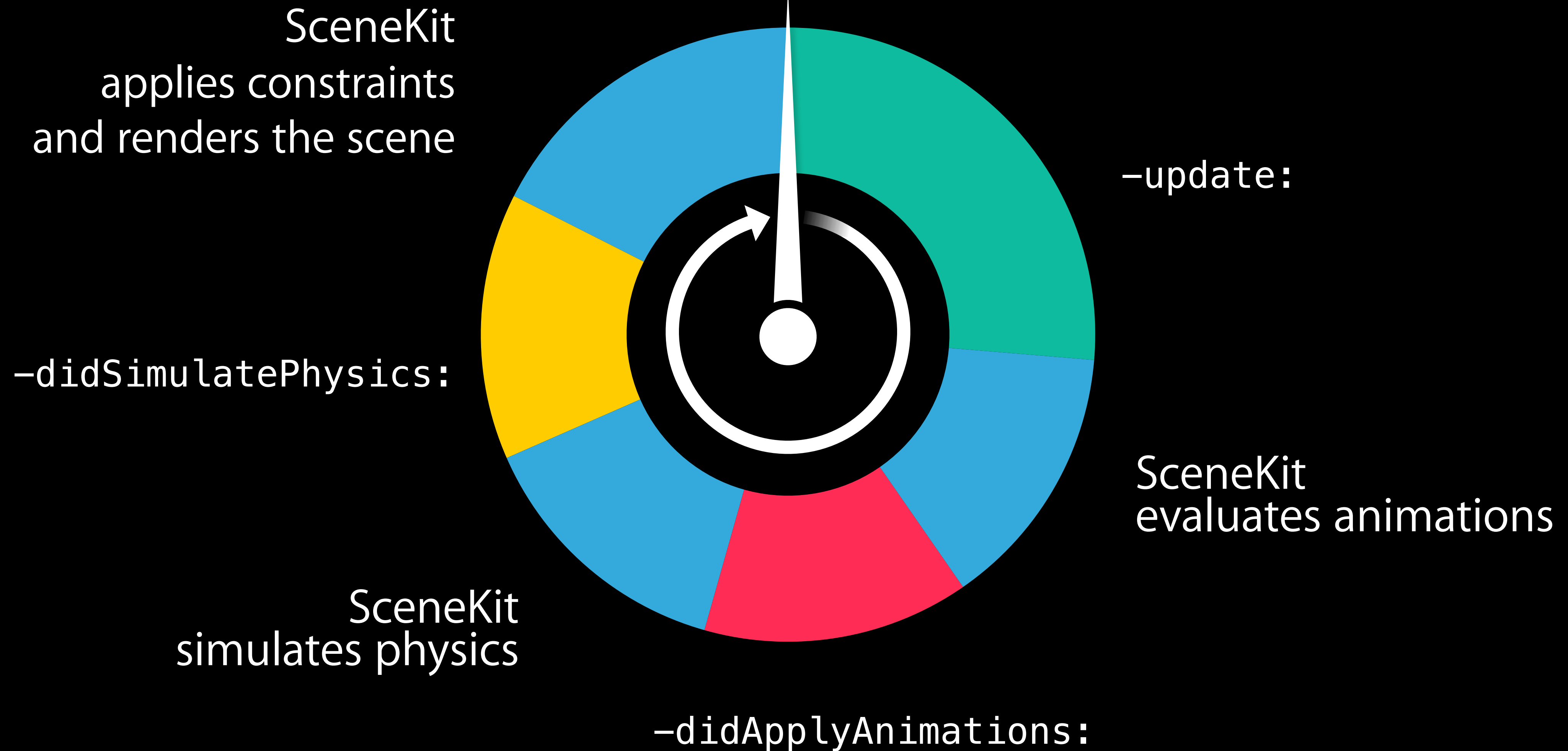


Performance Statistics

Xcode report



Each frame

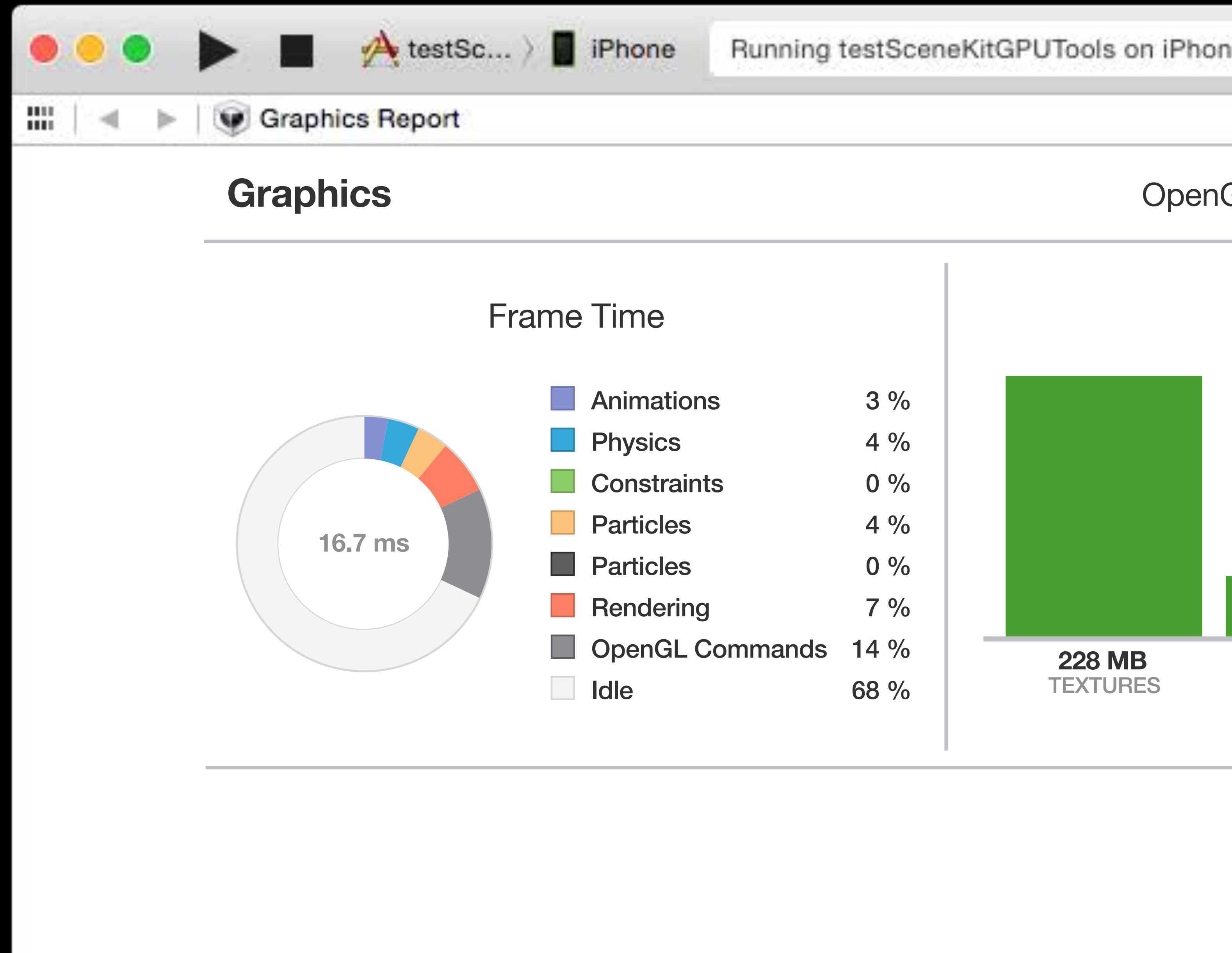


Performance Statistics

Xcode report

Reports CPU usage

Target 16 ms per frame

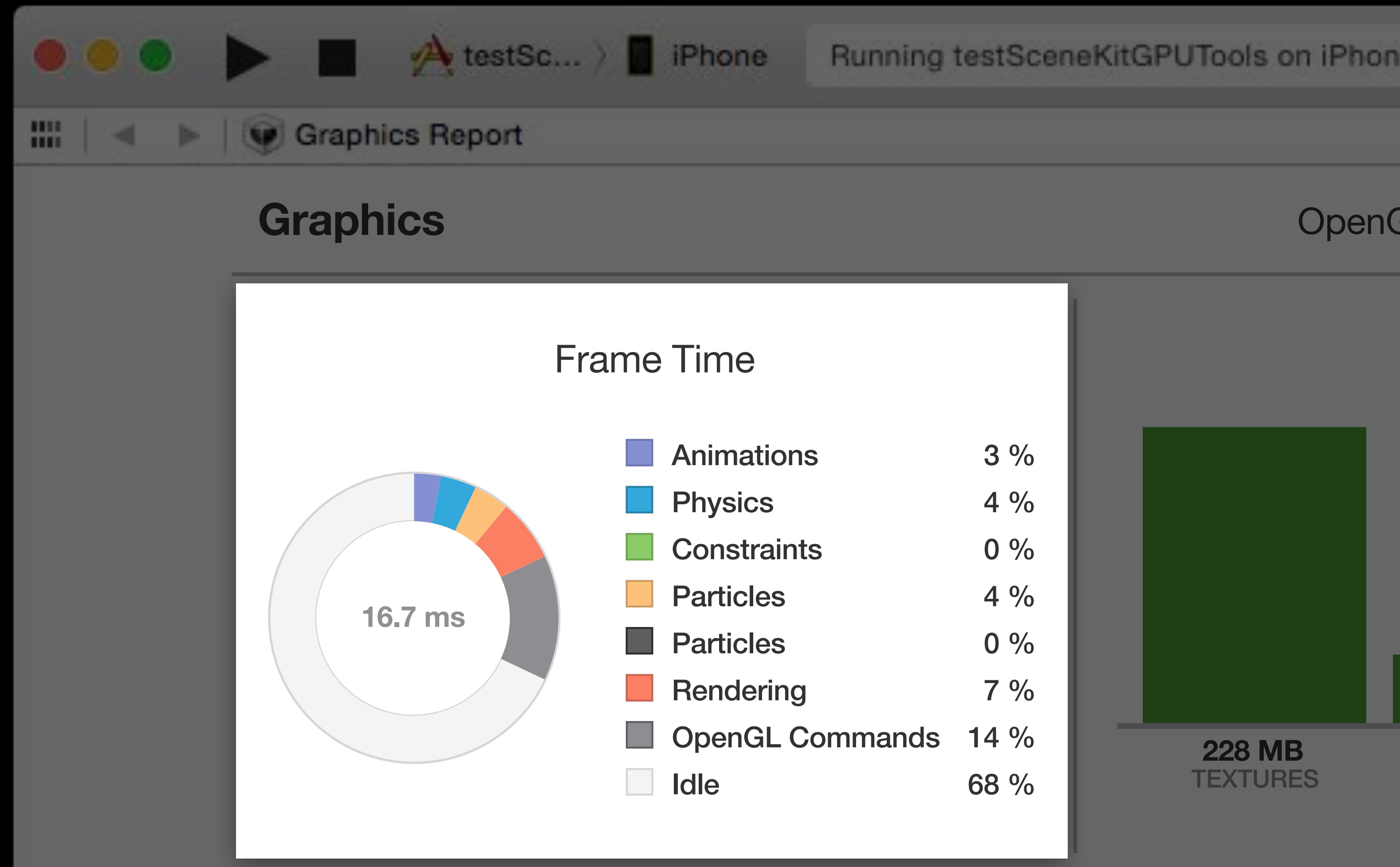


Performance Statistics

Xcode report

Reports CPU usage

Target 16 ms per frame



Performance Statistics

In-view statistics

```
aSCNView.showsStatistics = YES;
```



The screenshot displays a 3D scene with a colorful blimp, stars, and cannons. A performance statistics overlay is visible at the bottom of the window, showing various metrics such as FPS, time per frame, and counts for different resources.

SceneKit Mac Template

58fps

[time (ms/frame)] total: 5.17 anims: 0.07 constraints: 0.00 physics: 0.22 particles: 0.35
 culling: 0.07 lighting: 0.01 sort: 0.01 sync: 0.00
 delegate: 0.00 glFlush: 1.06 idle: 12.07

[count] prog:15 tex:23 fbo:1 vbo:16 ibo:14 rbo:1 vao:16
[memory] prog:21 tex:52.8MB fbo:1MB vbo:26.7MB ibo:285KB rbo:1MB
[switches] prog:21 tex:62 fbo:2 vbo:6 ibo:33 vao:32 enbl:567 rss:29

Performance Statistics

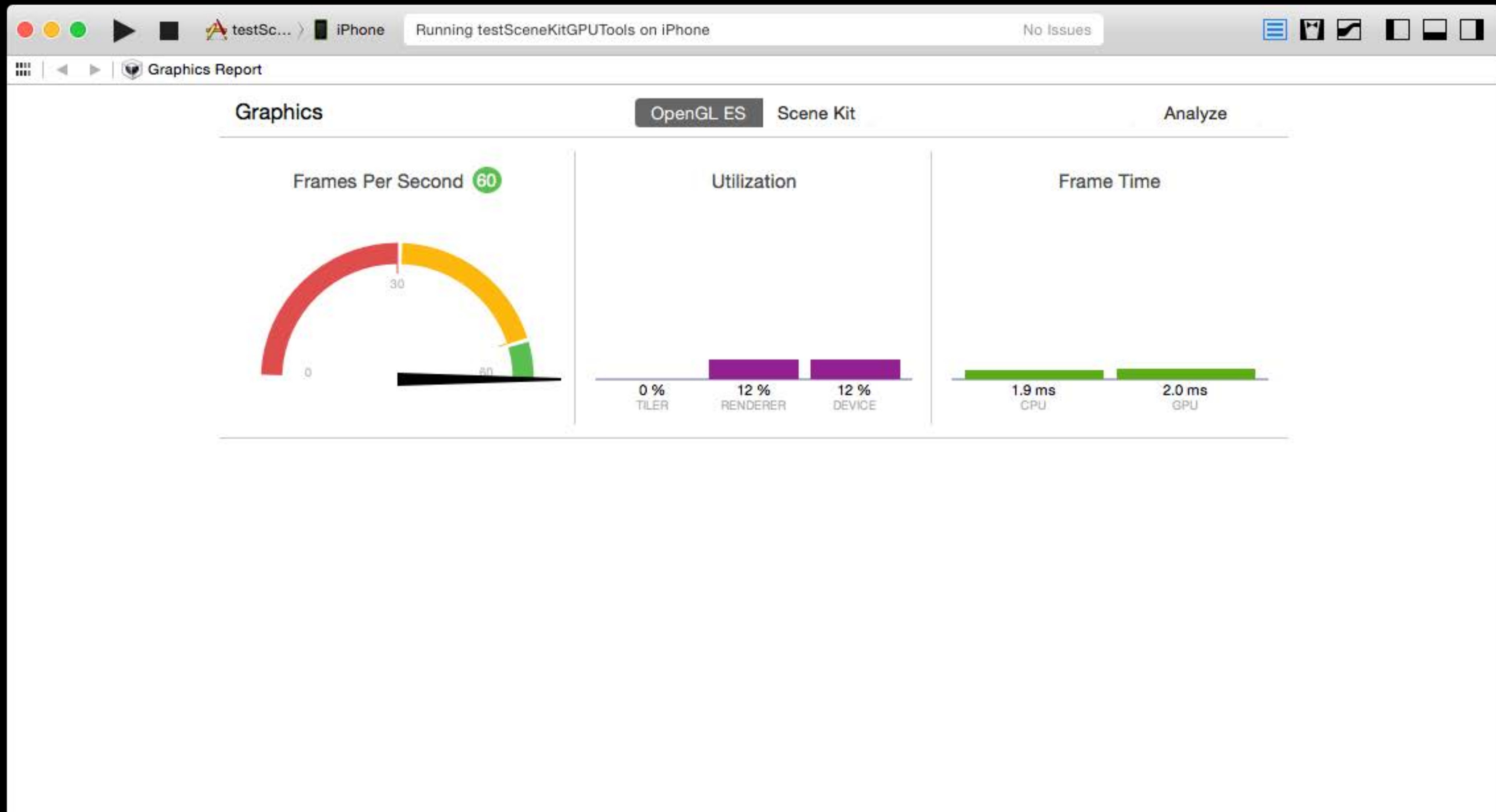
In-view statistics

```
aSCNView.showsStatistics = YES;
```



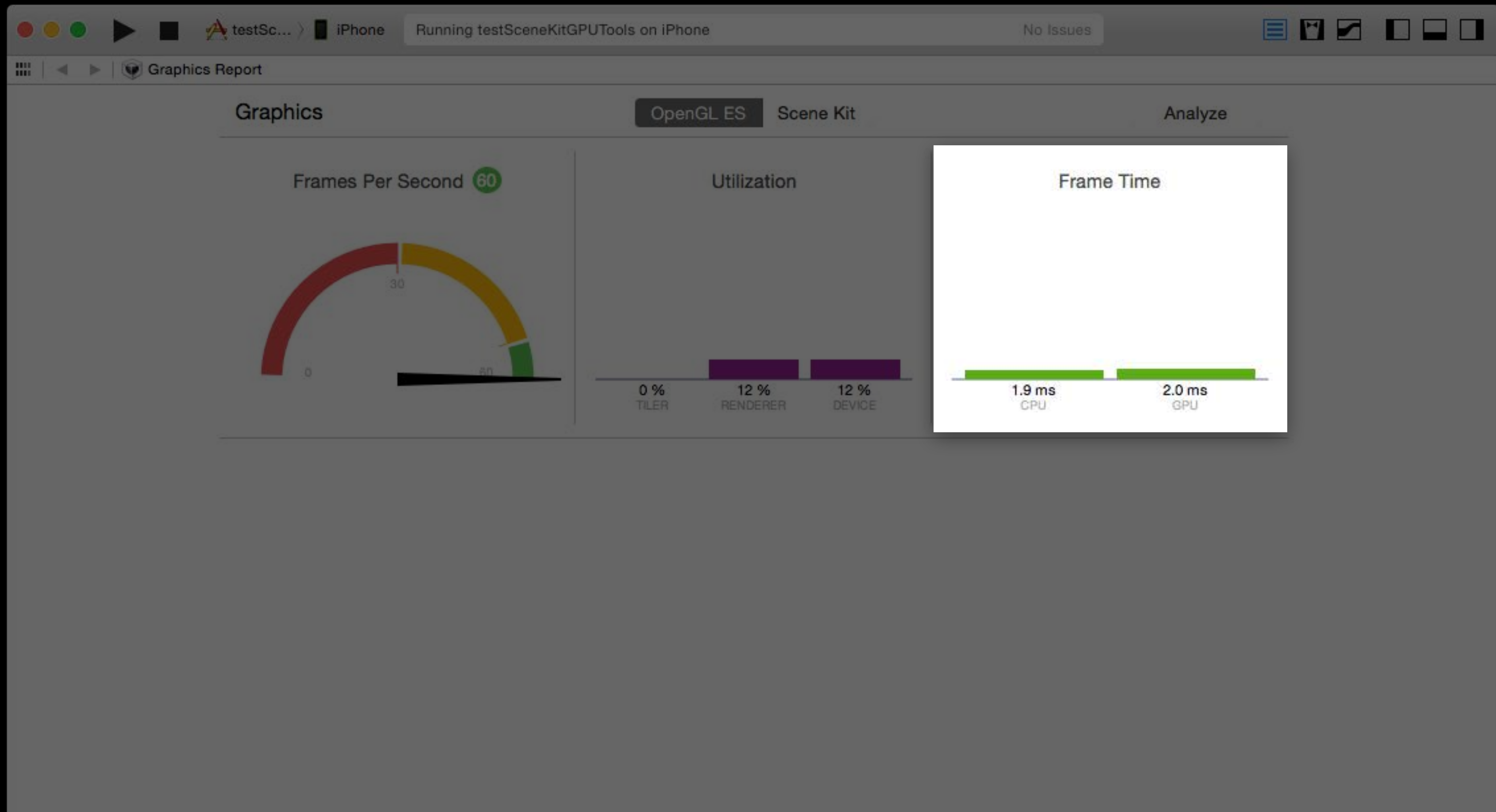
Analyzing Performance Statistics

Bottleneck—CPU or GPU?



Analyzing Performance Statistics

Bottleneck—CPU or GPU?



Troubleshooting

Bottleneck—CPU

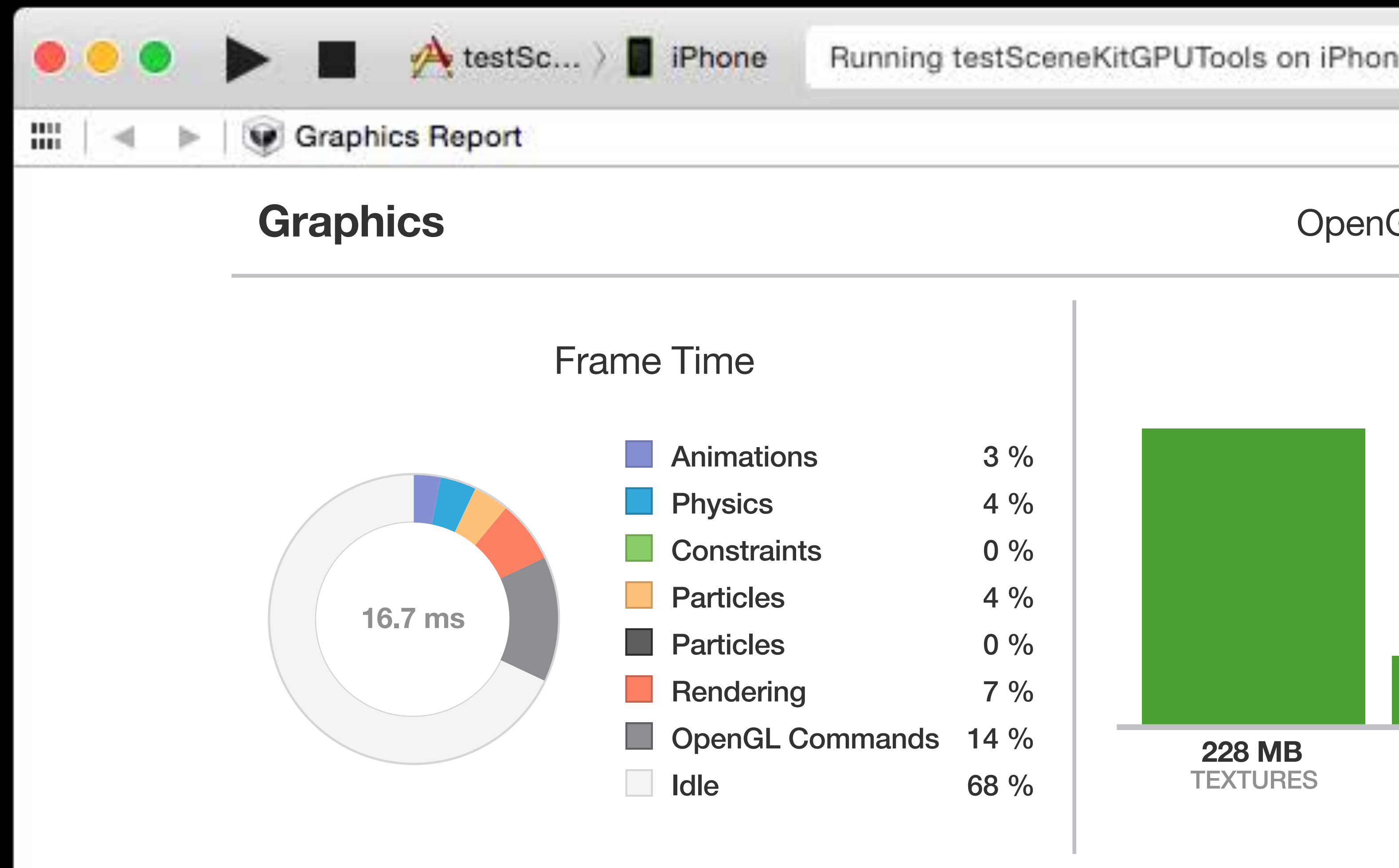
Less physic bodies

Less animations

Less actions

Less particles

Simpler game logic



Troubleshooting

Bottleneck—CPU

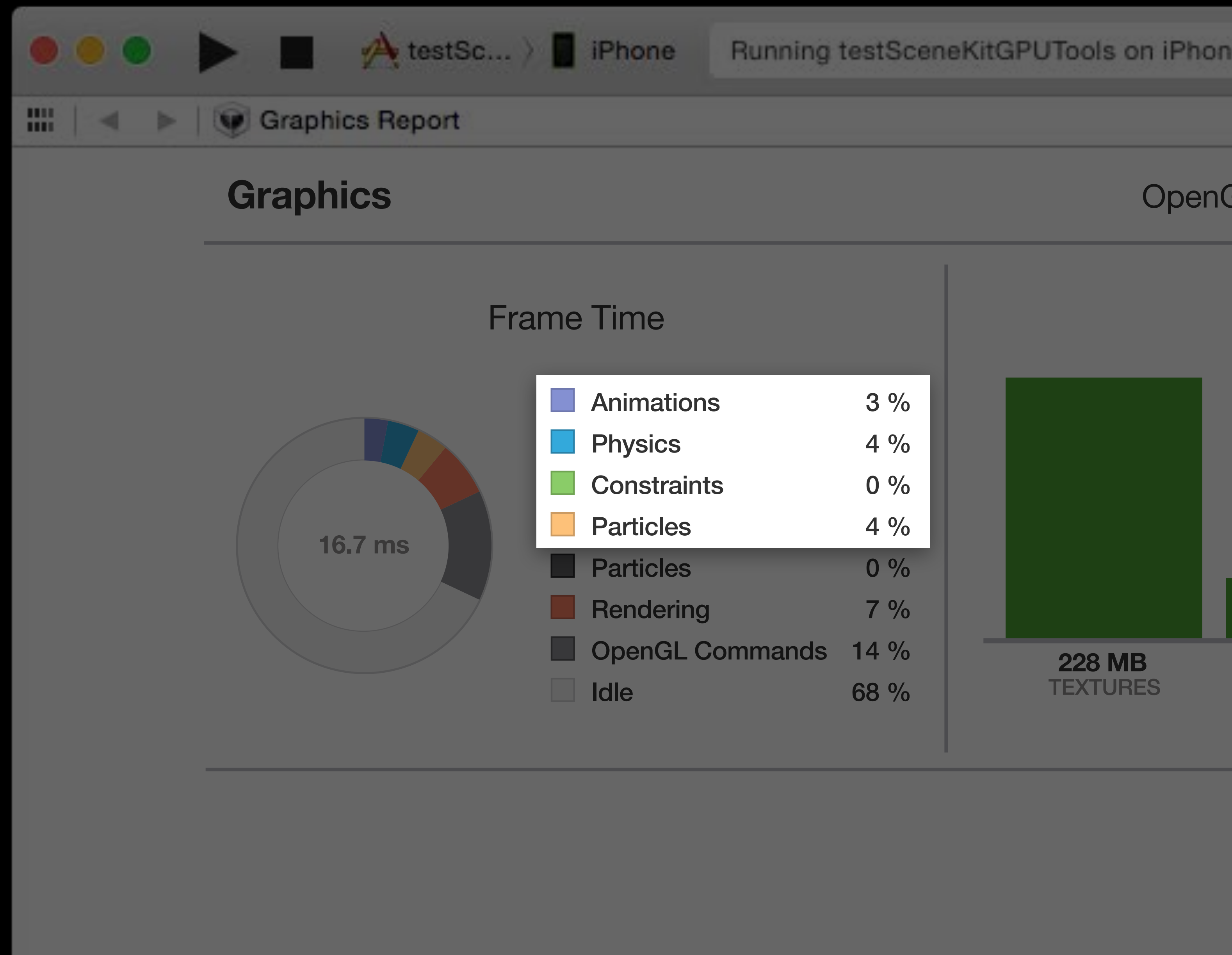
Less physic bodies

Less animations

Less actions

Less particles

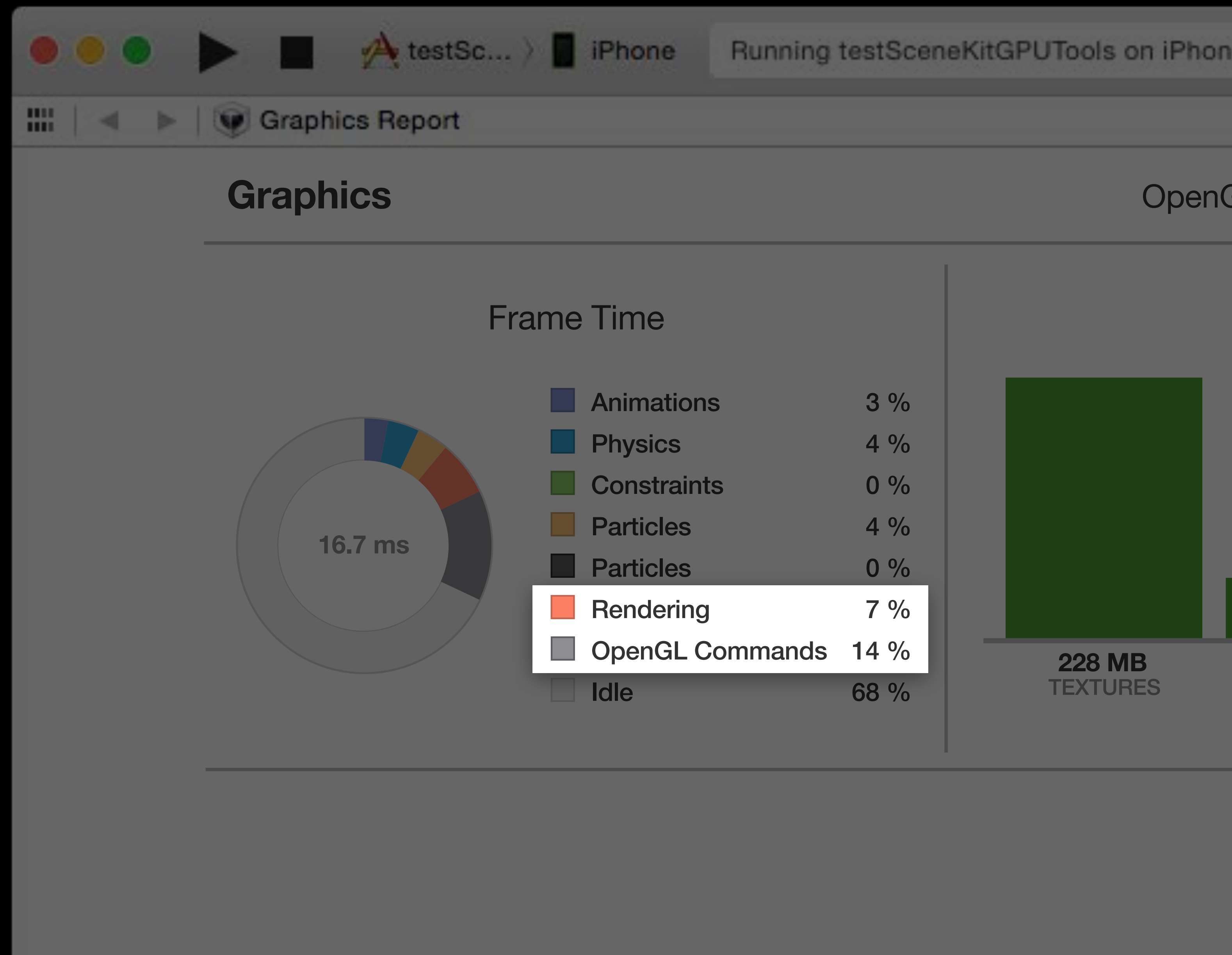
Simpler game logic



Troubleshooting

Bottleneck—CPU

Less draw calls

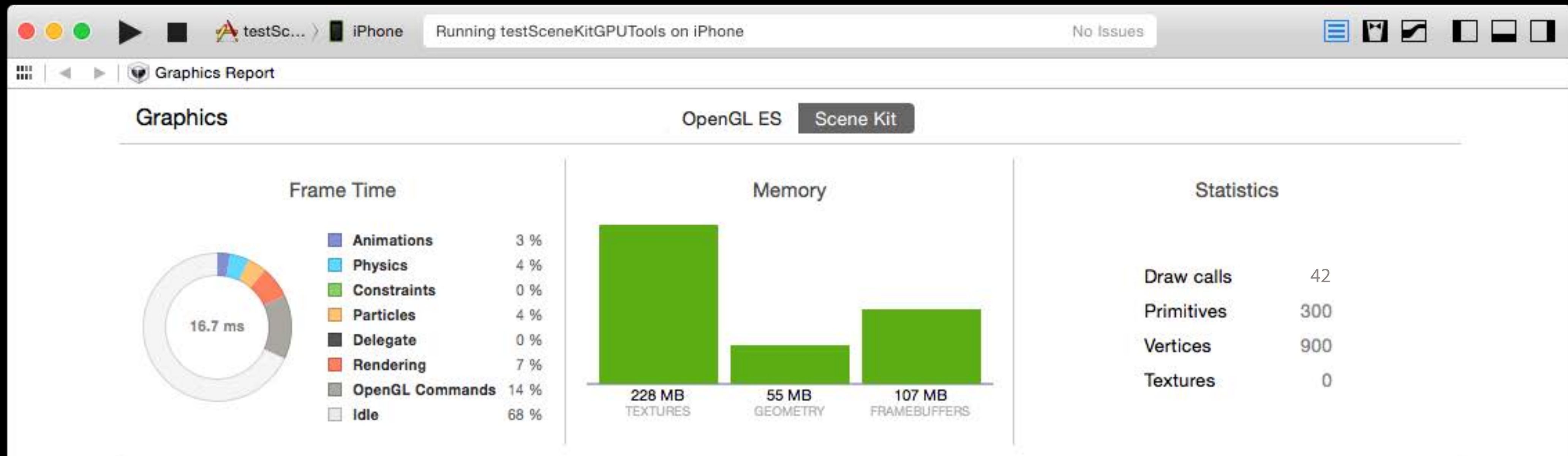


Draw Calls

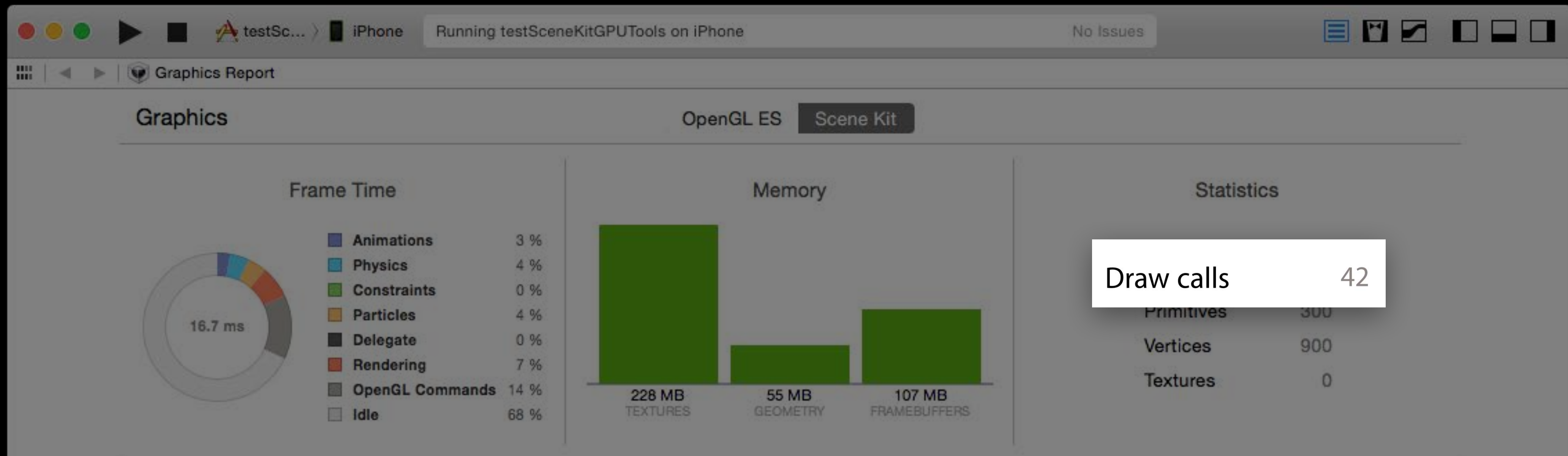
Number of draw calls is critical

CPU Impact

Draw Calls



Draw Calls



Draw Calls



Draw Calls



Flattening

Flatten to reduce the number of draw calls



Flattening

Flatten directly in 3D tools (recommended)

Flatten programmatically

```
// Flatten a node tree  
SCNNode *flattenedNode = [node flattenedClone];
```


Flattening

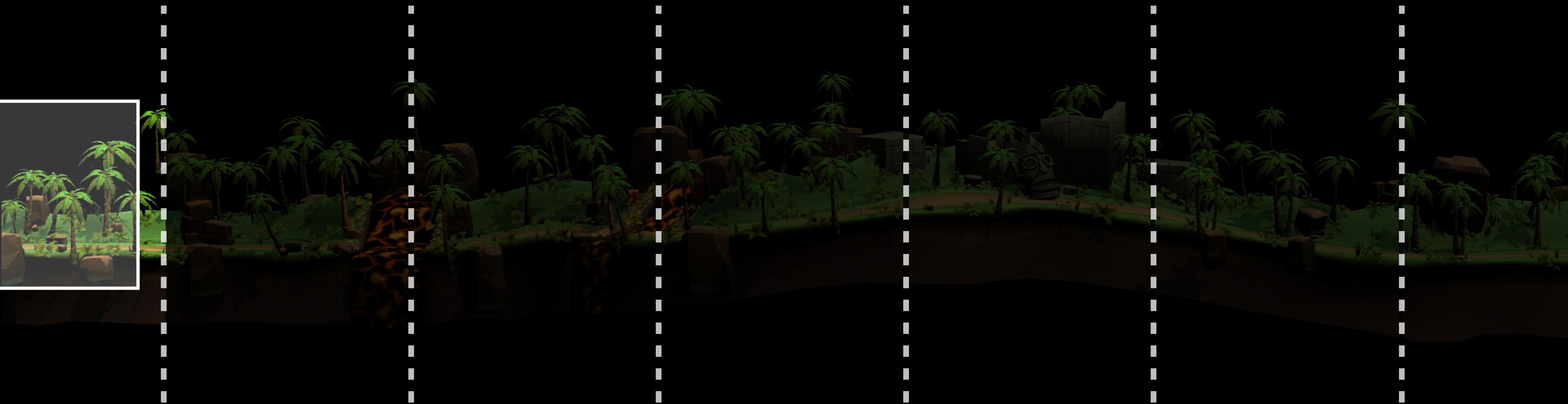
Don't flatten too much

Verify total number of polygons per frame

Verify number of influencing lights

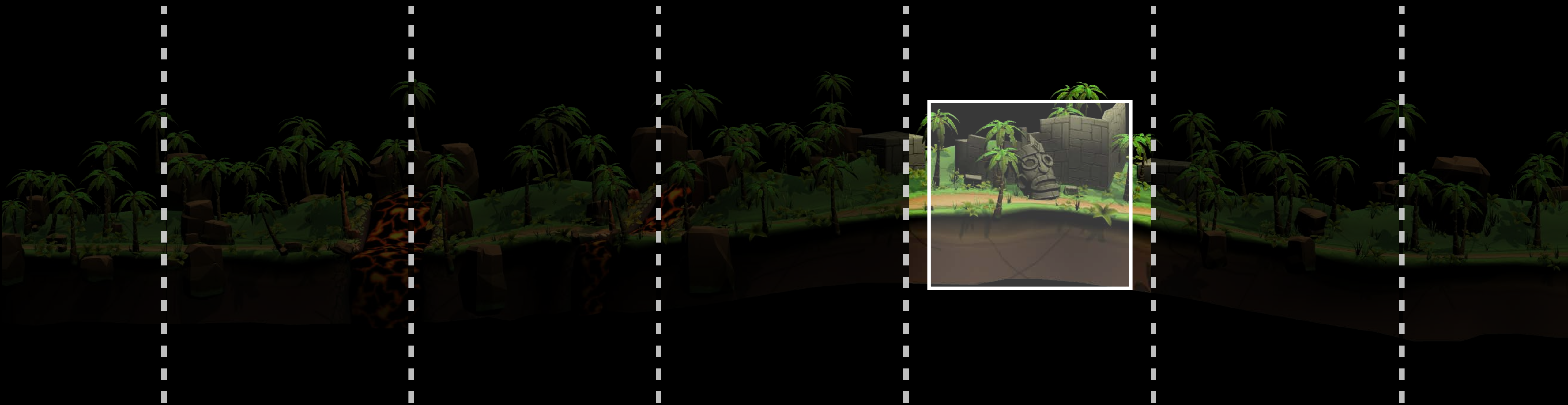
Flattening

Flattening and culling in Bananas



Flattening

Flattening and culling in Bananas

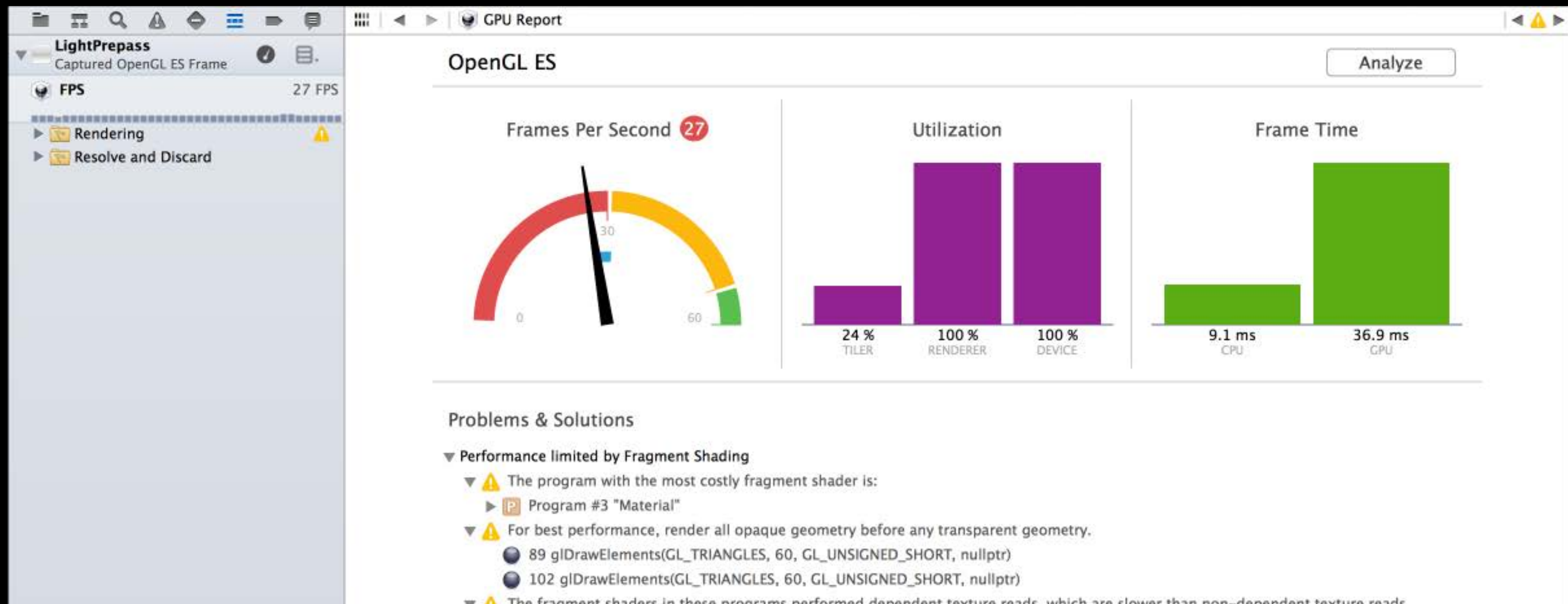


Troubleshooting

Bottleneck—GPU

Tiler

Renderer/device

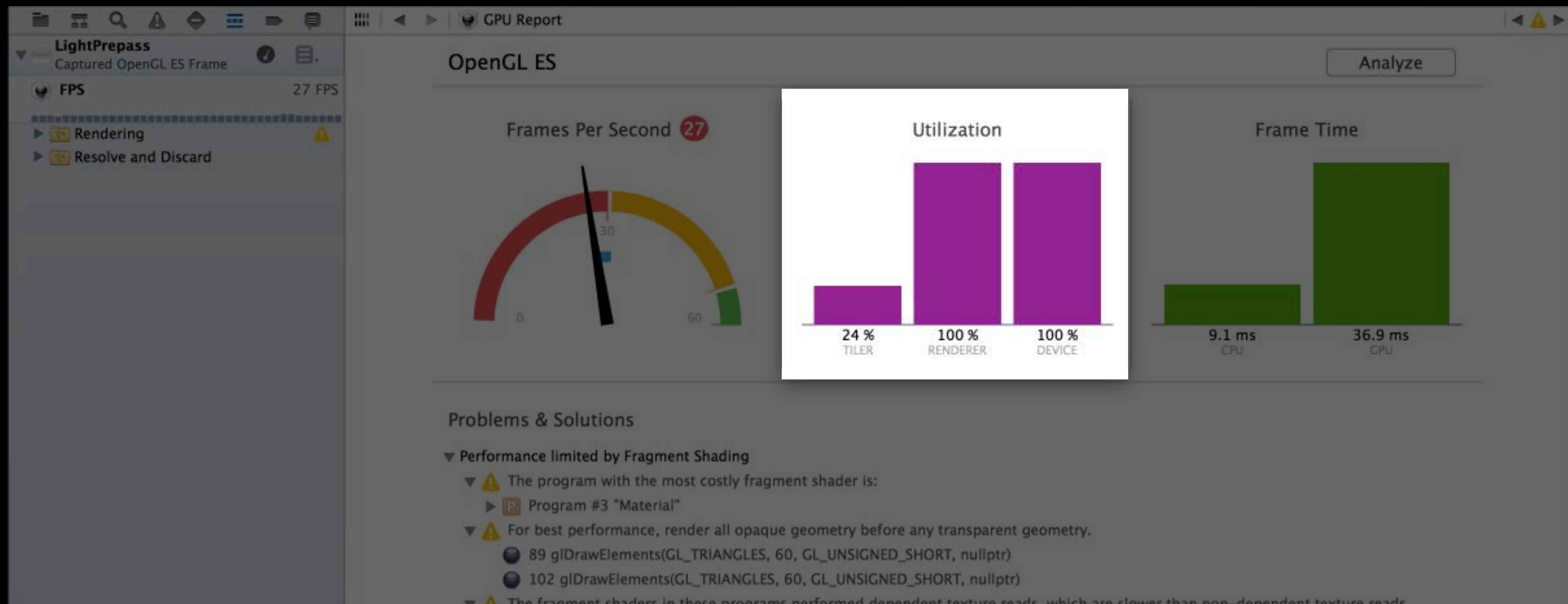


Troubleshooting

Bottleneck—GPU

Tiler

Renderer/device

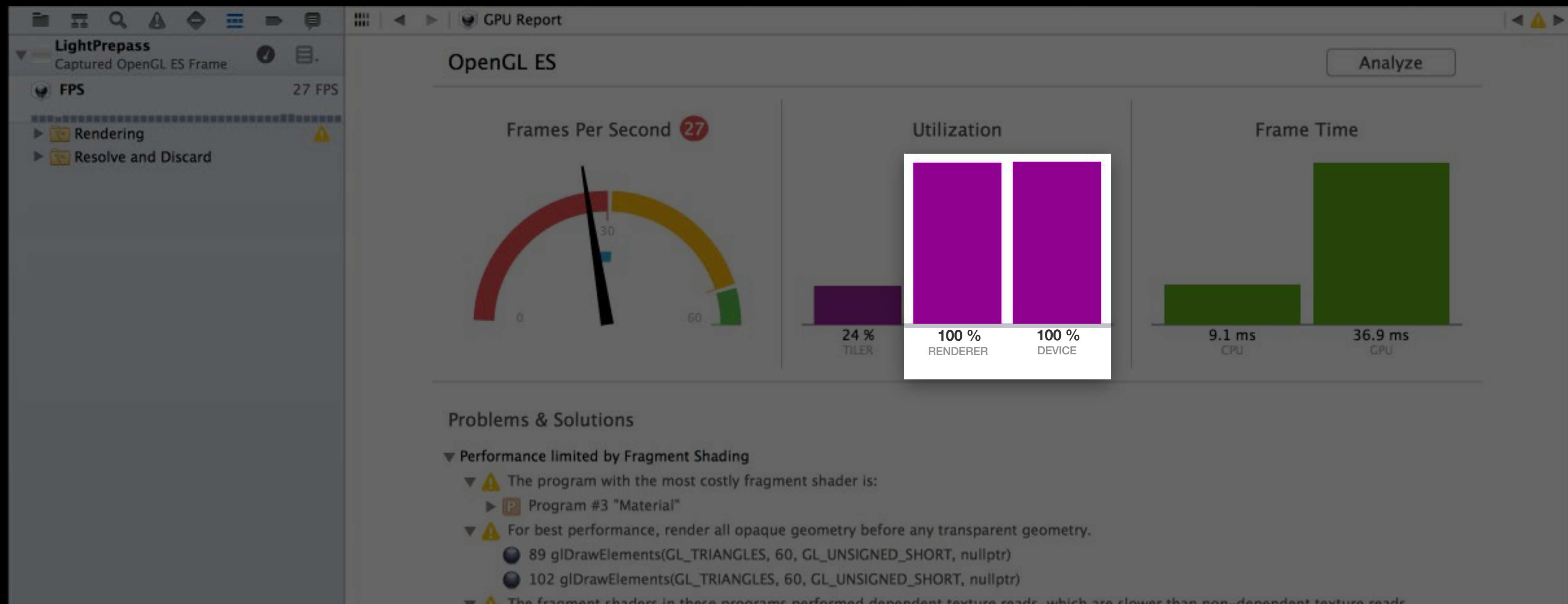


Troubleshooting

Bottleneck—Renderer/device

Fill rate limited

Fragment shaders limited



Fill Rate

Fill Rate

Contents scale factor

- 1x, 2x, intermediate

Fill Rate

Contents scale factor

- 1x, 2x, intermediate

Reduce screen space postprocesses

- Deferred shadows, depth of field, reflective floor
- Custom techniques

Fill Rate

Contents scale factor

- 1x, 2x, intermediate

Reduce screen space postprocesses

- Deferred shadows, depth of field, reflective floor
- Custom techniques

Anti-aliasing

```
//msaa 4x
```


```
aSCNView.antiAliasingMode = SCNAntiAliasingModeMultisampling4X;
```

```
//no antiAliasing
```

```
aSCNView.antiAliasingMode = SCNAntiAliasingModeNone;
```


Lighting

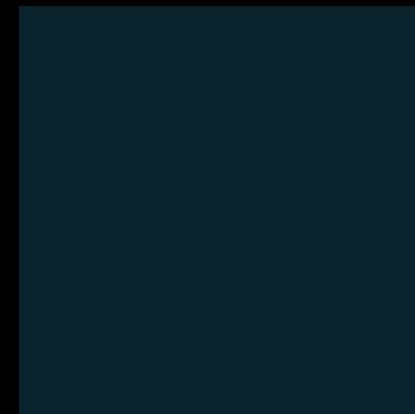
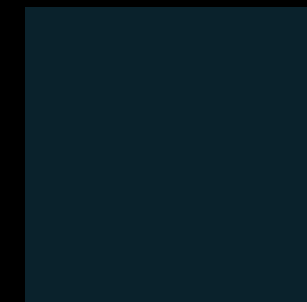
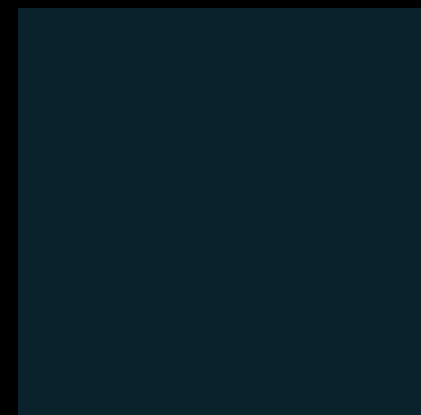
Dynamic vs. static

	Dynamic Lighting	Static Lighting
Quality	Good	Best
Performance	Costly	Fast
Work with animations		

Lighting

Area of influence

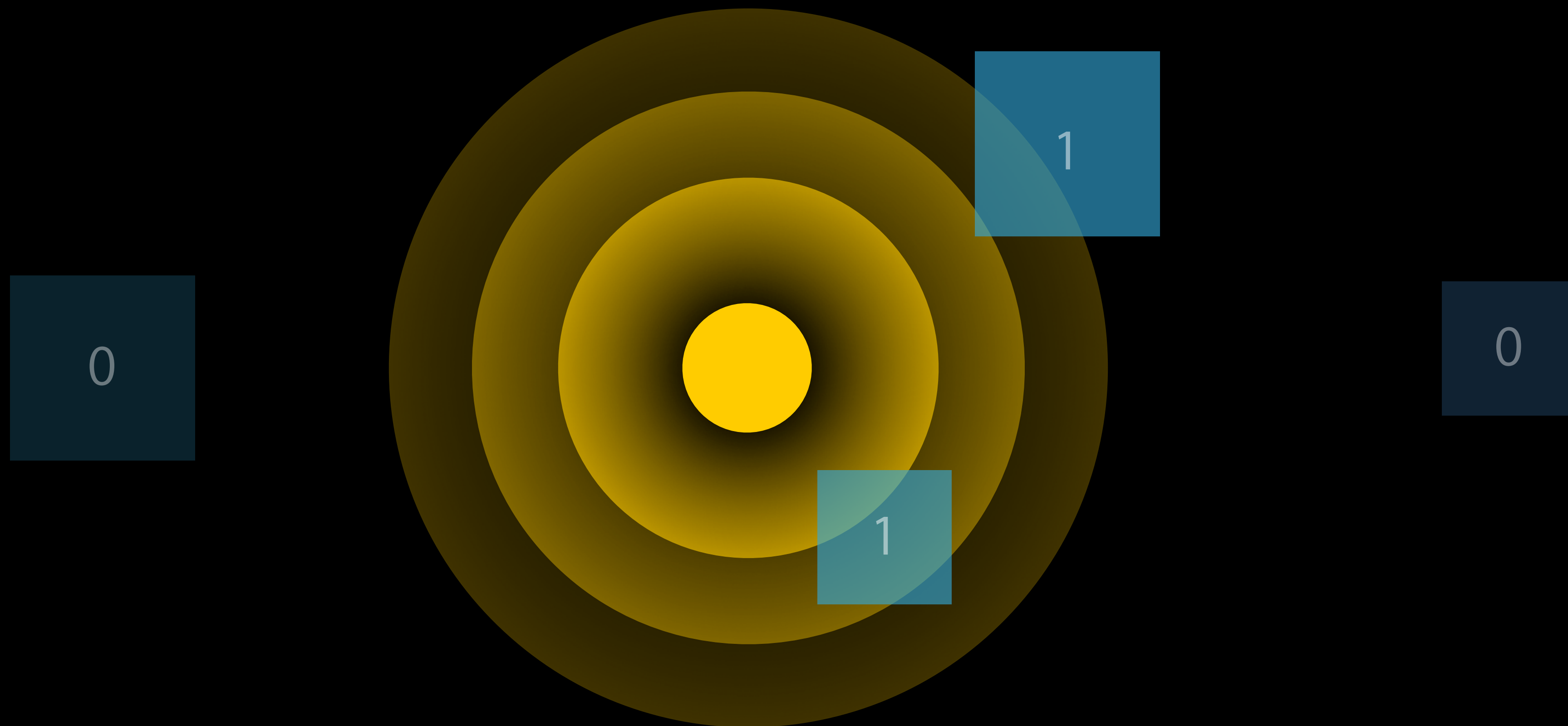
```
aLight.attenuationEndDistance = 100.0;
```



Lighting

Area of influence

```
aLight.attenuationEndDistance = 100.0;
```



Lighting

Influencing lights

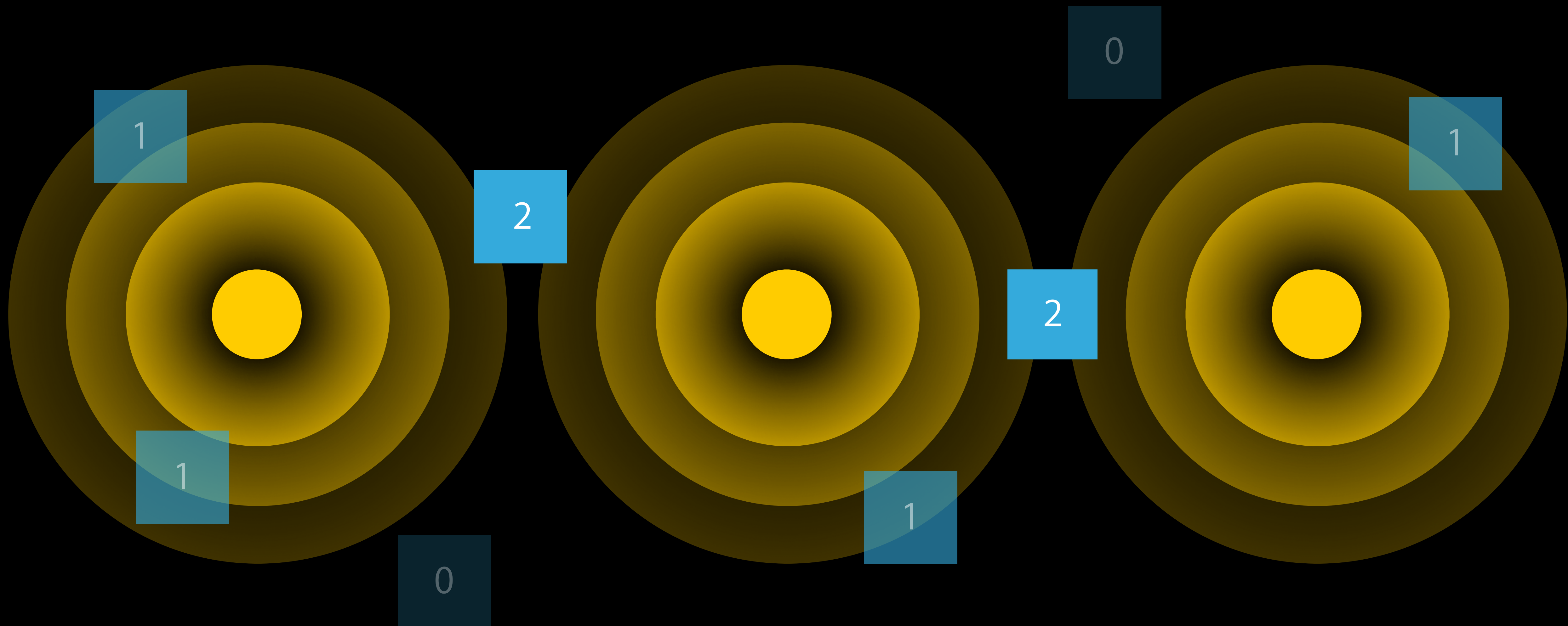
Number of influencing lights per object is what matters



Lighting

Influencing lights

Number of influencing lights per object is what matters



Lighting

Banana

Omni lights located far apart

Objects lit by 1 omni light at maximum

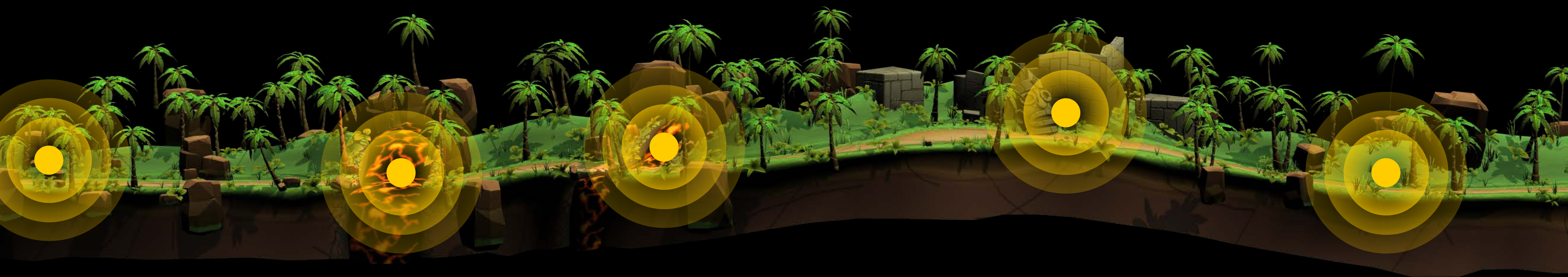


Lighting

Banana

Omni lights located far apart

Objects lit by 1 omni light at maximum



Lighting

Optimizing shadows

Dynamic



Static



Lighting

Optimizing shadows

Dynamic



Static



Shadows

Dynamic shadow modes

NEW



Forward



Projected

API

`SCNShadowModeForward`

`SCNShadowModeModulated`
+ `gobo` image

Performance
impact

High

Low

Quality

Best

Low

Shadows

Optimizing shadows

NEW

```
// Set the size of the shadow map  
aLight.shadowMapSize = CGSizeMake(512, 512);
```



Shadows

Optimizing shadows

NEW

```
// Set the size of the shadow map  
aLight.shadowMapSize = CGSizeMake(512, 512);
```



Shadows

Optimizing shadows

NEW

```
// Configure the number of sample in the shadow map  
aLight.shadowSampleCount = 1.0;
```



Shadows

Optimizing shadows

NEW

```
// Configure the number of sample in the shadow map  
aLight.shadowSampleCount = 1.0;
```



Texturing

Avoid unnecessarily large textures
Pack textures into texture atlases



Texturing

Mipmapping

Advantages

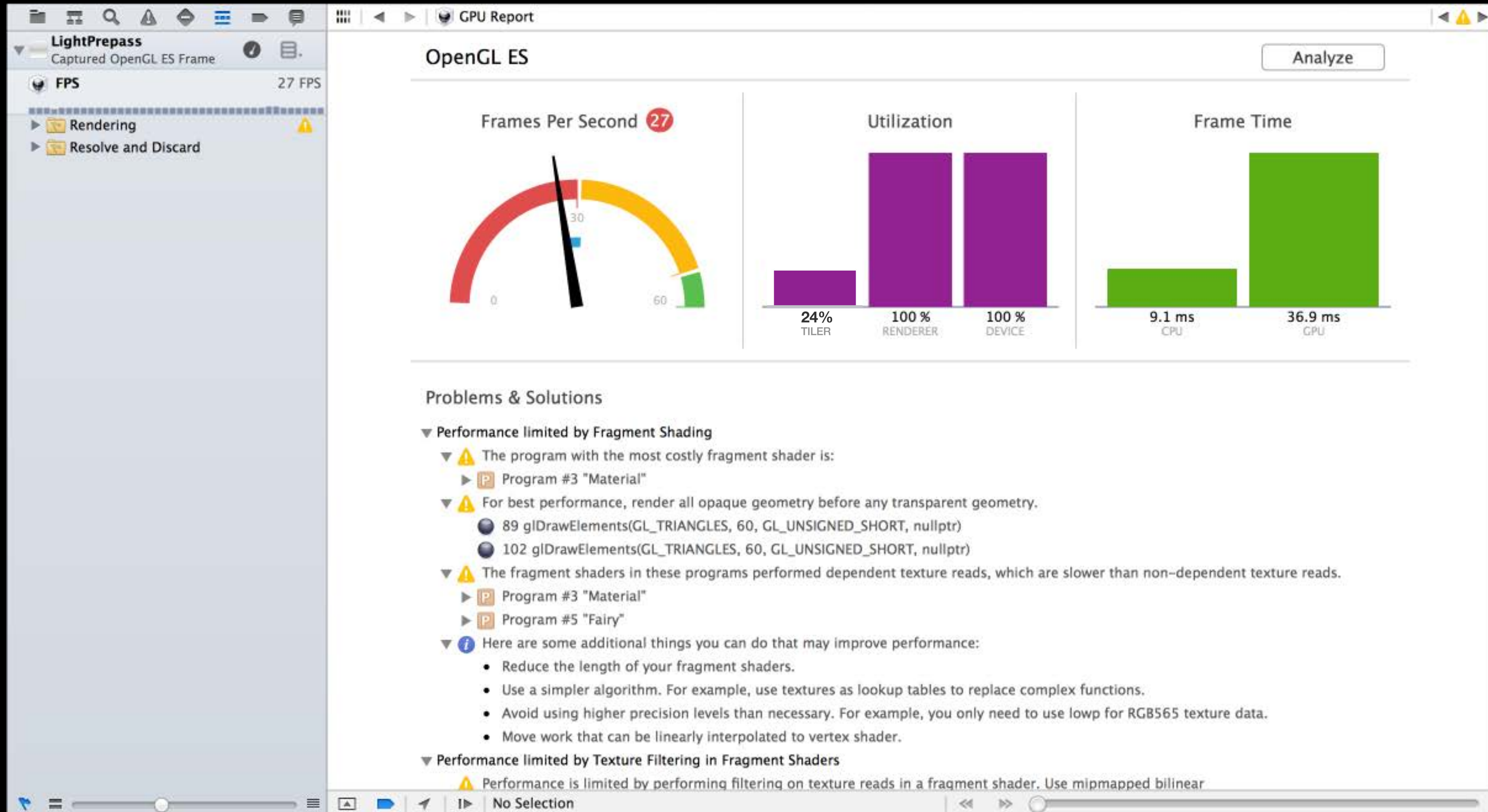
- Can improve performance significantly
- Reduce aliasing/Moiré effect

Drawbacks

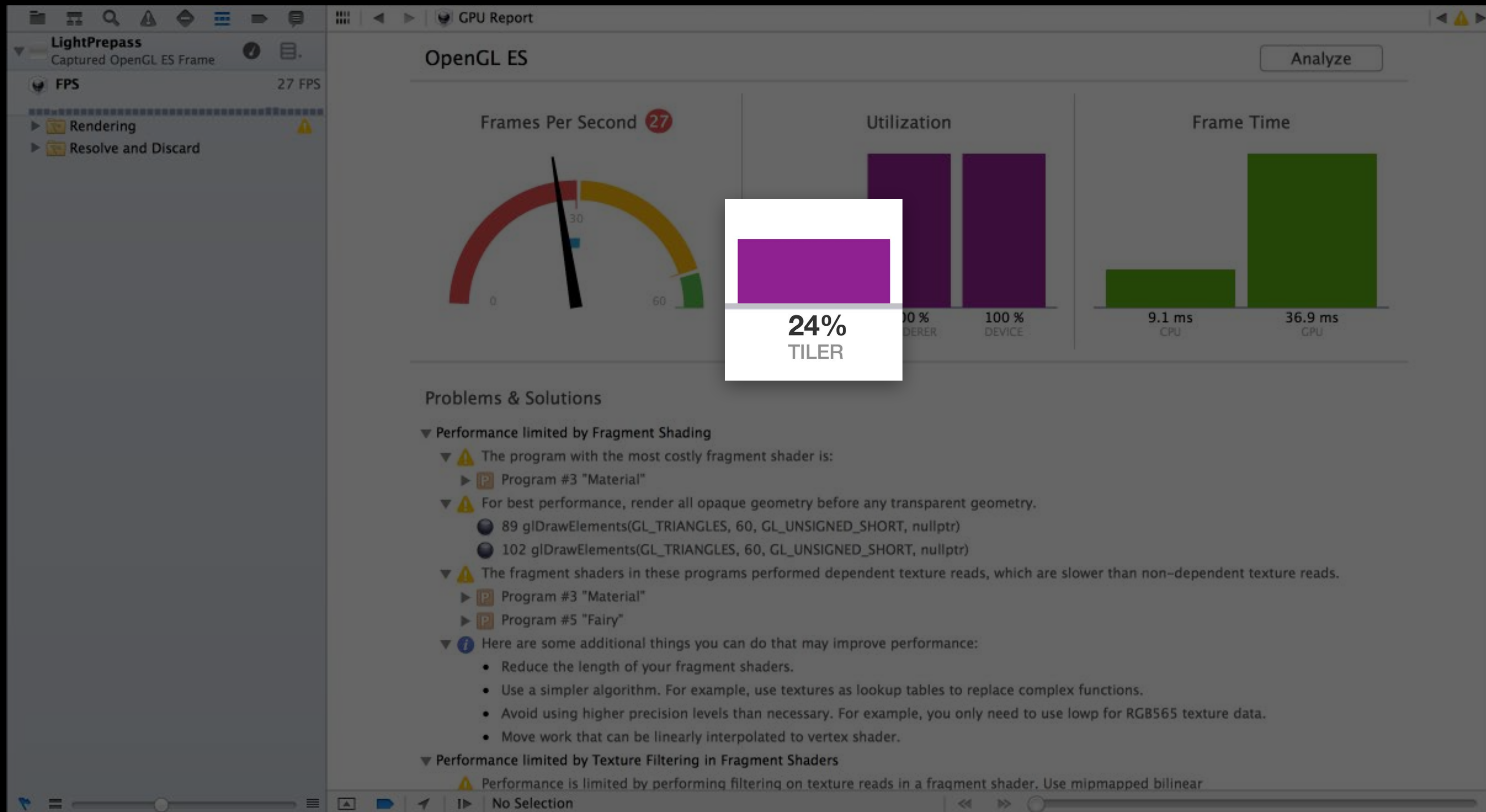
- Higher load time
- ~30% more memory

```
// Turn on mipmapping  
aMaterial.diffuse.mipFilter = SCNFilterModeLinear;
```

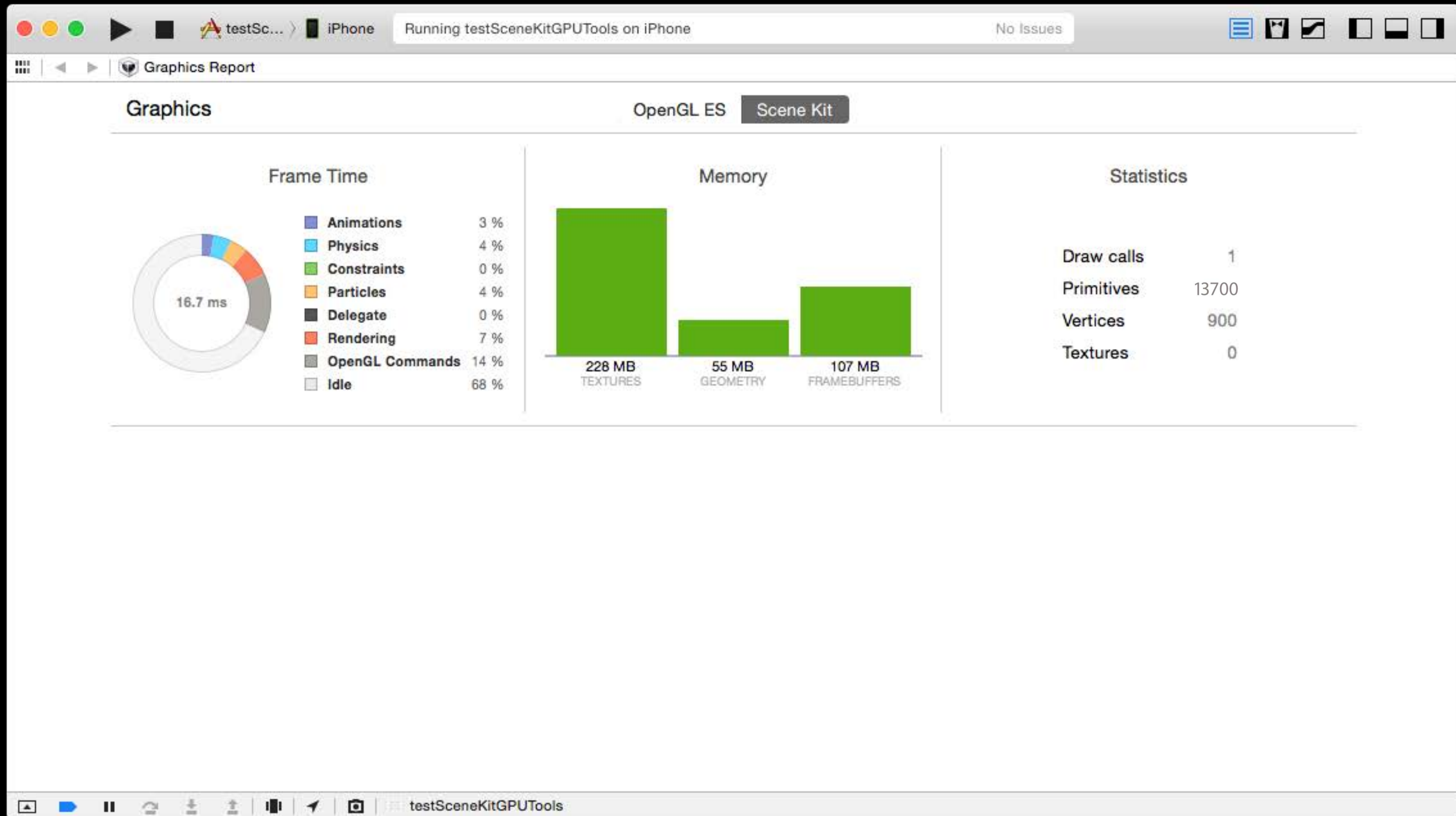

Polygon Rate



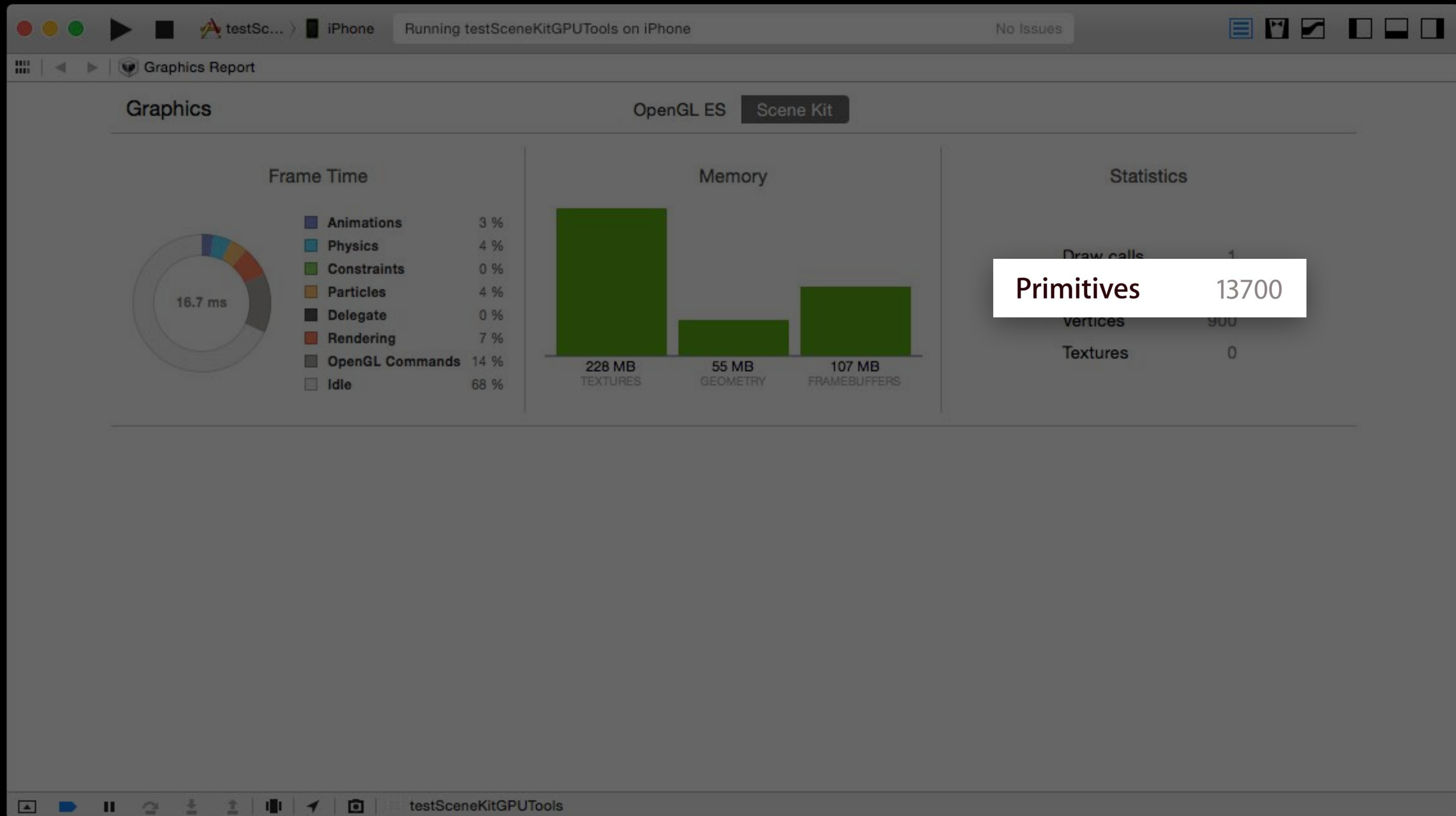
Polygon Rate



Polygon Rate



Polygon Rate



Polygon Rate



Polygon Rate



Levels of Detail

Reduce polygon count per frame

```
SCNLevelOfDetail *lod1 = [SCNLevelOfDetail  
levelOfDetailWithGeometry:aGeometry
```

```
worldSpaceDistance:aDistance];  
geometry.levelOfDetail = @[lod1, lod2, ...];
```



lod1



lod2

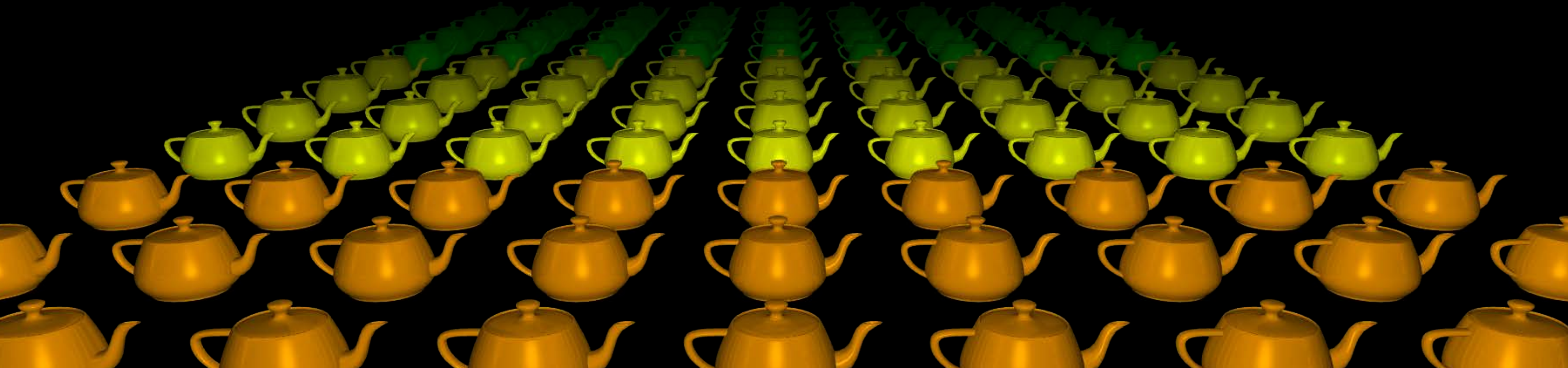


lod3

Levels of Detail

Reduce polygon count per frame

```
SCNLevelOfDetail *lod1 = [SCNLevelOfDetail levelOfDetailWithGeometry:aGeometry  
                           worldSpaceDistance:aDistance];  
geometry.levelOfDetail = @[lod1, lod2, ...];
```



Troubleshooting

Summary

Identify bottlenecks

- CPU
- Tiler
- Renderer or device

Troubleshooting

Bottleneck—CPU

Reduce draw calls by flattening

Less physic bodies

Less animations

Less actions

Troubleshooting

Bottleneck—Tiler

Levels of detail

Split scenes in chunks

Troubleshooting

Bottleneck—Renderer or device

Simpler materials

Less/simpler lights

Smaller textures, mipmapping

Downscaled contents size

Less postprocess

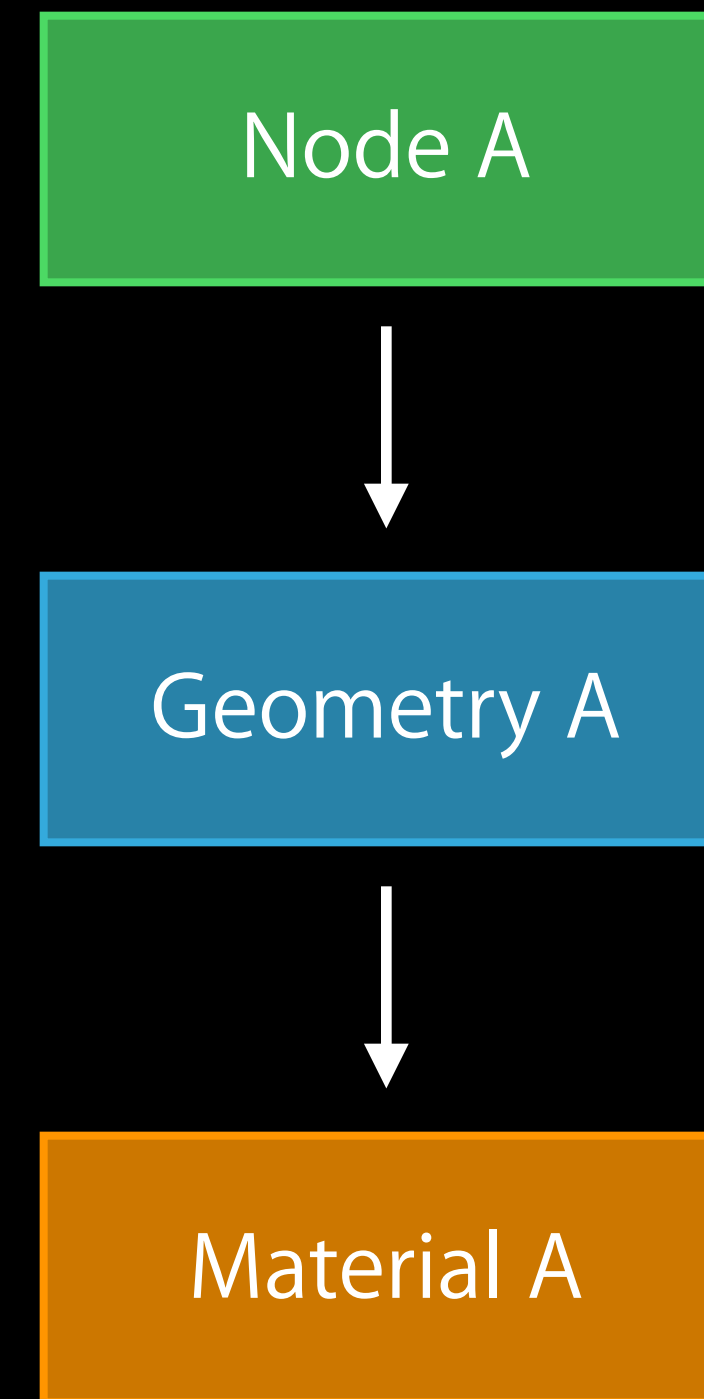
No multisampling

Performance Notes

Sharing

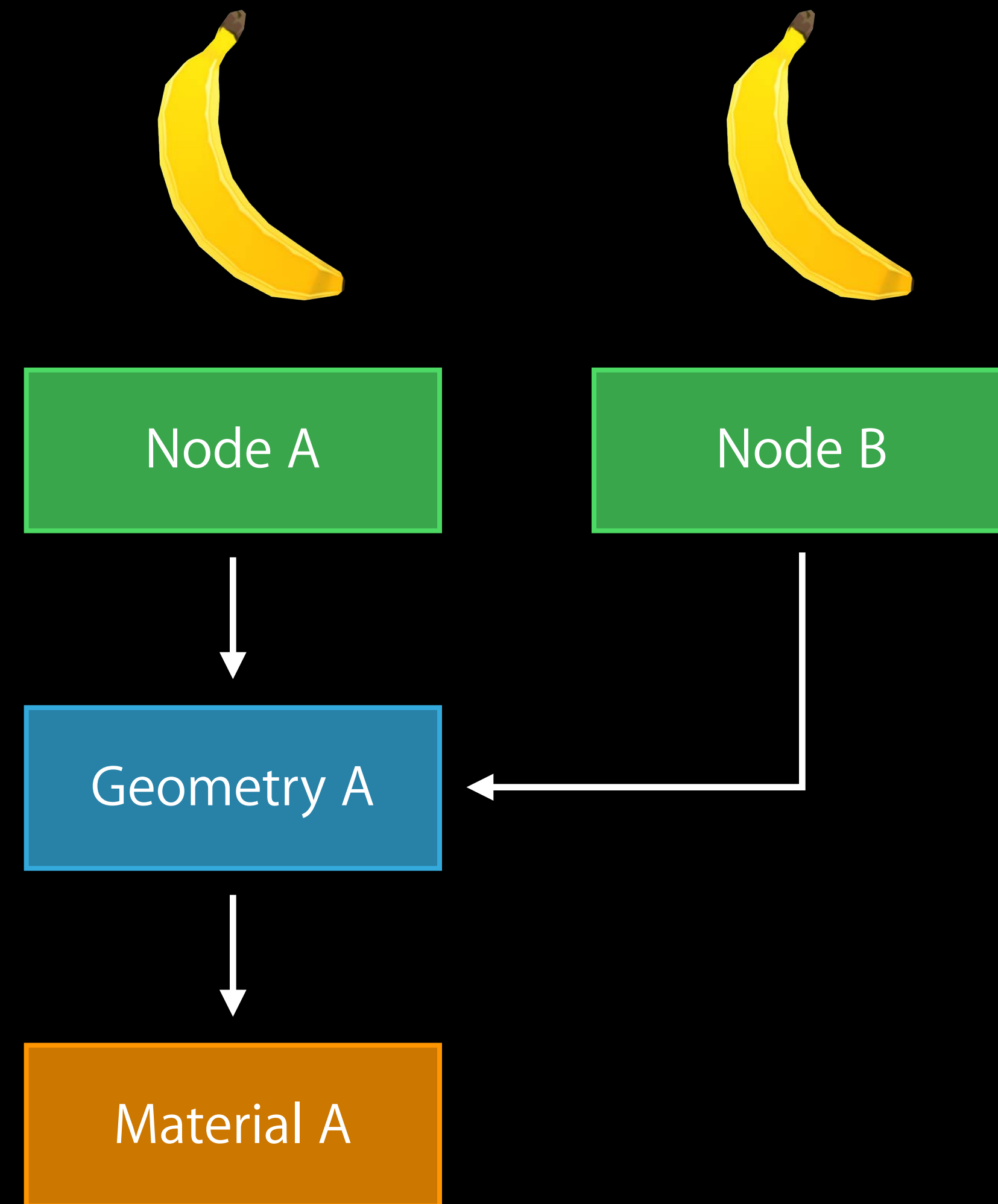


```
// Duplicate a node  
SCNNode *nodeB = [nodeA copy];
```



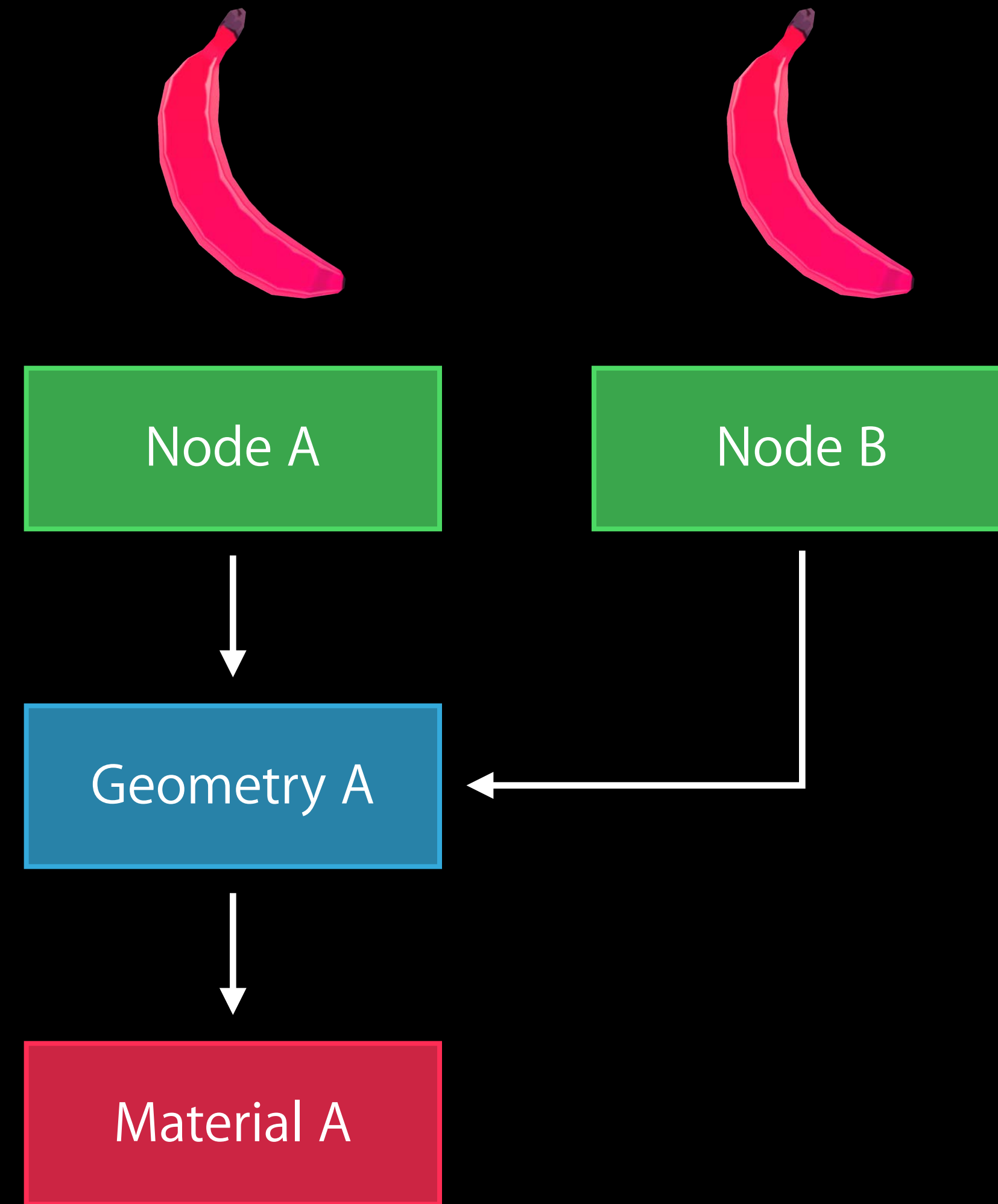
Sharing

```
// Duplicate a node  
SCNNode *nodeB = [nodeA copy];
```



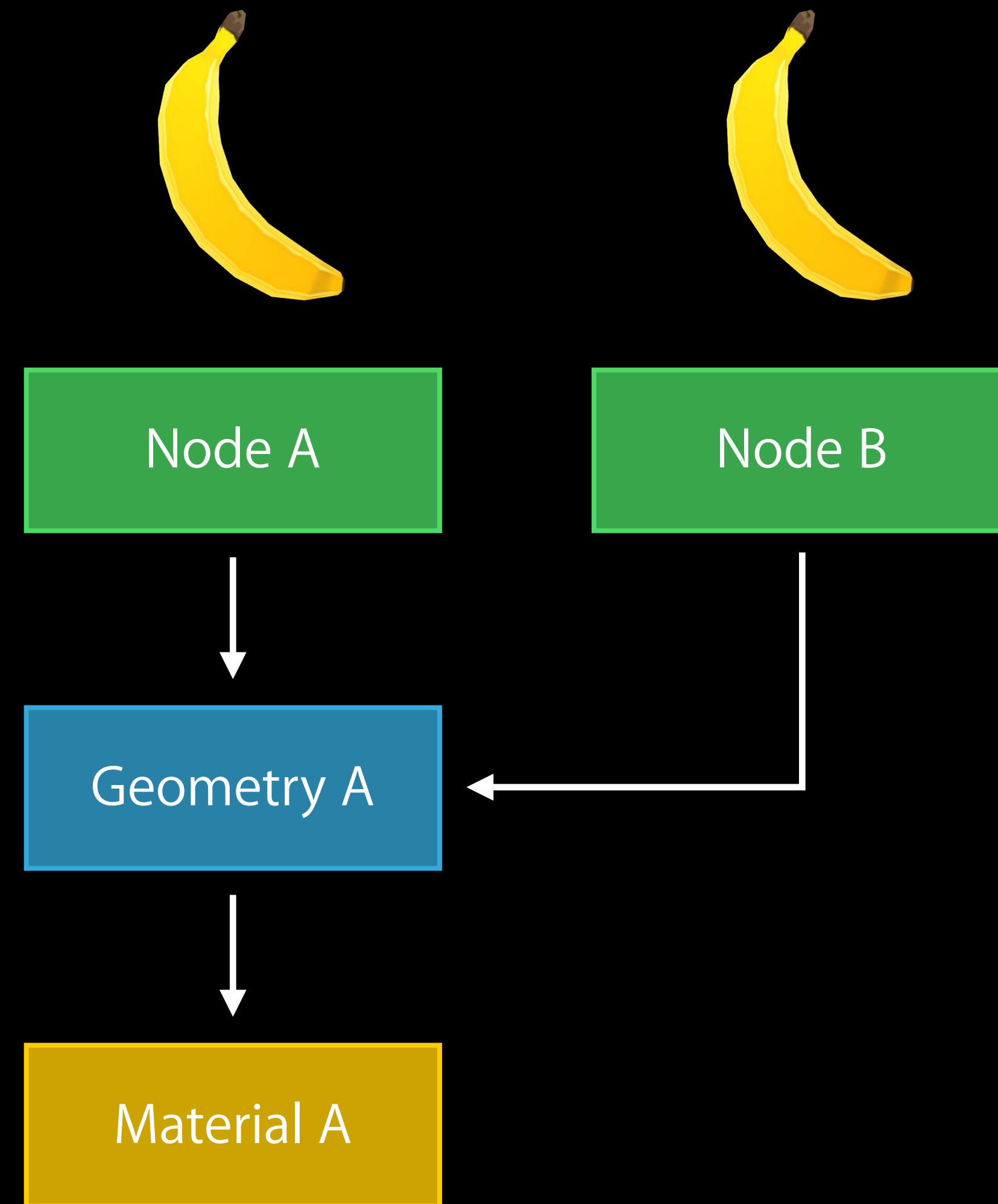
Sharing

```
// Duplicate a node  
SCNNode *nodeB = [nodeA copy];
```



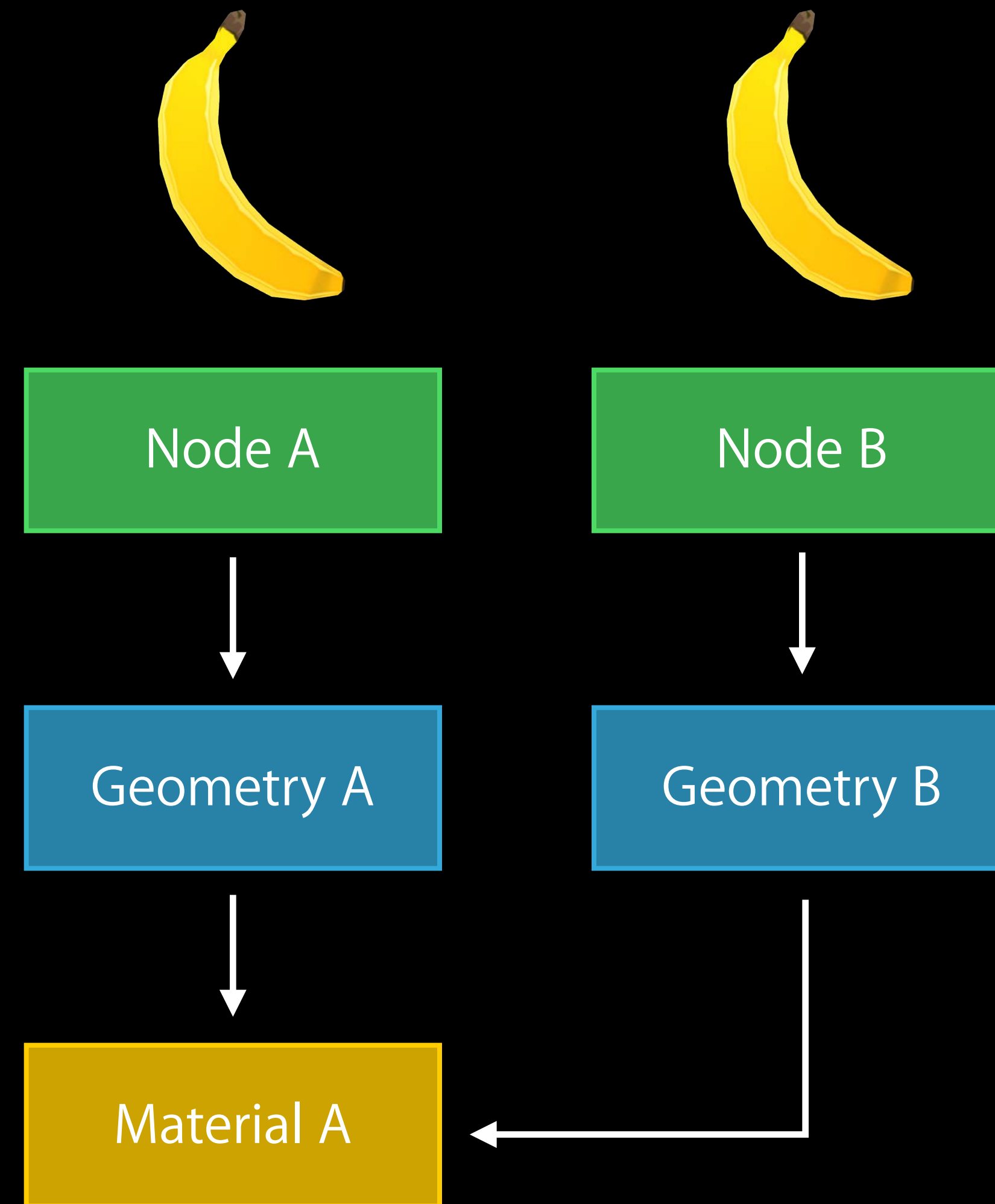
Sharing

```
// Duplicate a node  
SCNNode *nodeB = [nodeA copy];
```



Sharing

```
// Duplicate a node  
SCNNode *nodeB = [nodeA copy];  
  
// Un-share the geometry (cheap)  
nodeB.geometry = [nodeA.geometry copy];
```



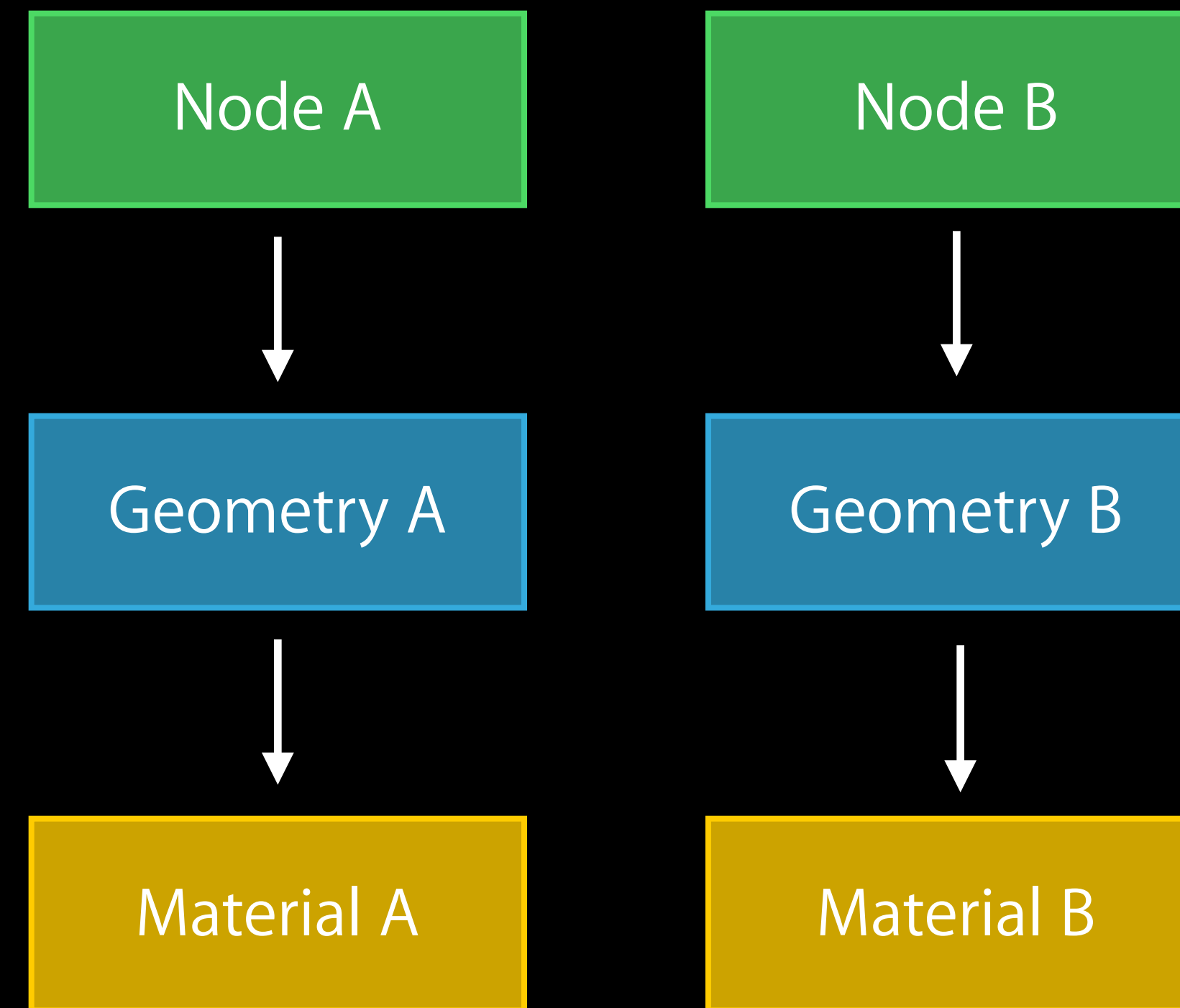
Sharing



```
// Duplicate a node
SCNNode *nodeB = [nodeA copy];

// Un-share the geometry (cheap)
nodeB.geometry = [nodeA.geometry copy];

// Un-share the material
nodeB.geometry.firstMaterial =
[nodeA.geometry.firstMaterial copy];
```



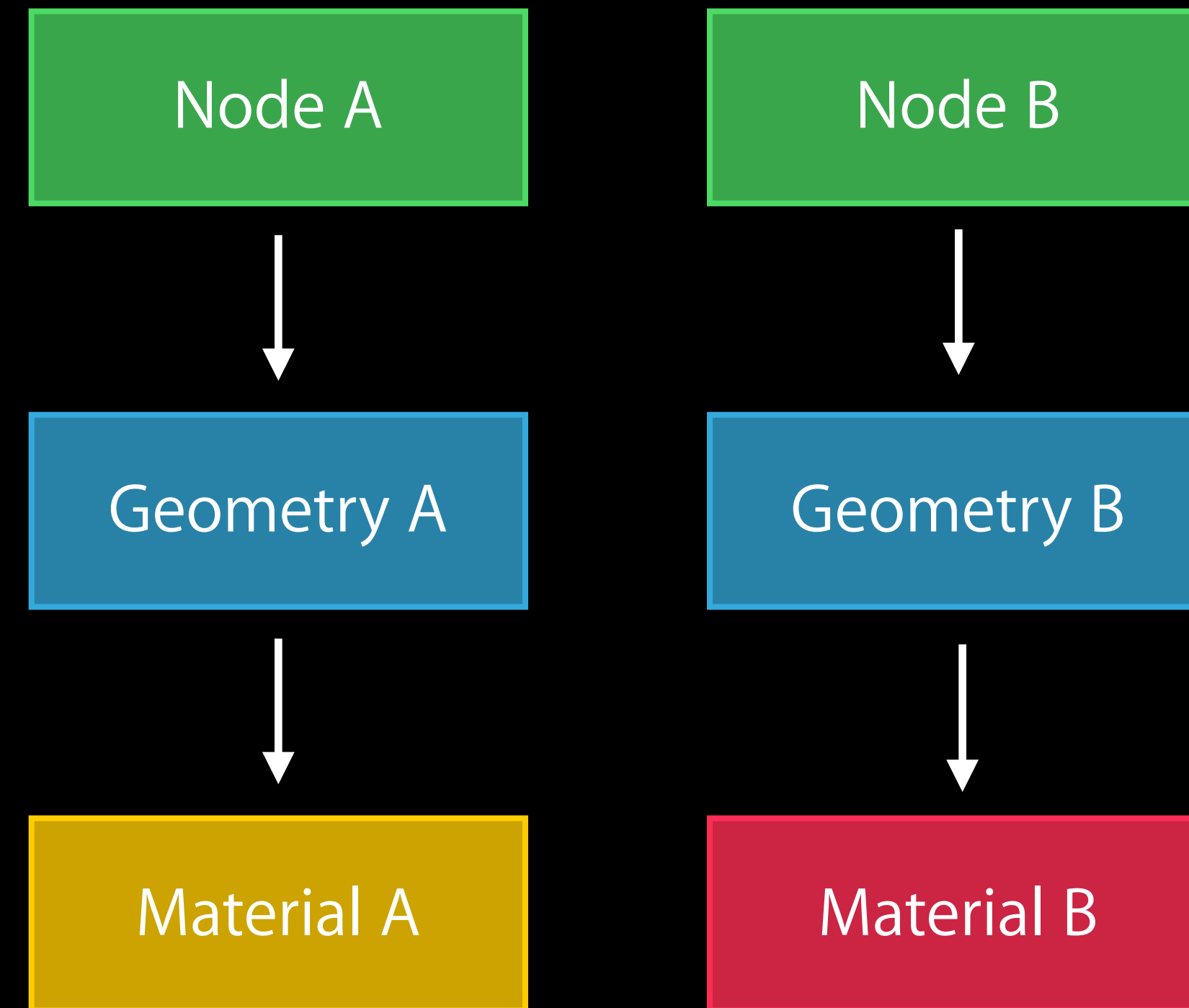
Sharing



```
// Duplicate a node
SCNNode *nodeB = [nodeA copy];

// Un-share the geometry (cheap)
nodeB.geometry = [nodeA.geometry copy];

// Un-share the material
nodeB.geometry.firstMaterial =
[nodeA.geometry.firstMaterial copy];
```



Preloading

NEW

Supported by `SCNView`, `SCNLayer` and `SCNRenderer`

```
- (void)prepareObjects:(NSArray *)objects  
  completionHandler:(BOOL (^)(BOOL success))completionHandler;
```

Preloading



Object

Resources

SCNMaterial

Textures

SCNGeometry

Geometry buffers and materials

SCNNode

Node tree and geometries

SCNScene

Node tree and shaders (lighting)

Creating Custom Tools

Archiving



Adopted by SCNScene, SCNNode, SCNGeometry, SCNMaterial, etc.

Conform to NSSecureCoding

```
// Archive a scene
NSData *archivedScene = [NSKeyedArchiver archivedDataWithRootObject:scene];

// Unarchive a scene
SCNScene *scene = [NSKeyedUnarchiver unarchiveObjectWithData:archivedScene];
```


Exporting

Export to Collada

Import back in 3D software

Only a subset of SceneKit can be exported

```
- (BOOL)writeToURL:(NSURL *)url
    options:(NSDictionary *)options
    delegate:(id <SCNSceneExportDelegate>)delegate
    completionHandler:(SCNSceneExportProgressHandler)progressHandler;
```

Archiving vs. Exporting

Archiving

Collada export

Loadable on iOS

Can be read by DCC tools

Scene graph
(nodes, geometry, materials...)

Animations

Actions

Physics

Particles

Custom shaders

Archiving vs. Exporting

Archiving

Collada export

Loadable on iOS



Through Xcode

Can be read by DCC tools

Scene graph
(nodes, geometry, materials...)

Animations



Actions

Physics

Particles

Custom shaders

Archiving vs. Exporting

	Archiving	Collada export
Loadable on iOS		Through Xcode
Can be read by DCC tools		
Scene graph (nodes, geometry, materials...)		
Animations		
Actions		
Physics		
Particles		
Custom shaders		

Archiving vs. Exporting

	Archiving	Collada export
Loadable on iOS	✓	Through Xcode
Can be read by DCC tools		✓
Scene graph (nodes, geometry, materials...)	✓	✓
Animations	✓	✓
Actions		
Physics		
Particles		
Custom shaders		

Archiving vs. Exporting

	Archiving	Collada export
Loadable on iOS	✓	Through Xcode
Can be read by DCC tools		✓
Scene graph (nodes, geometry, materials...)	✓	✓
Animations	✓	✓
Actions	✓	
Physics	✓	
Particles	✓	
Custom shaders	✓	

Scripting



JavaScript bridge

SceneKit is fully scriptable

```
void SCNExportJavaScriptModule(JSContext *context);
```

Code Samples

Sample code "Bananas"

Checkout the three other code samples

More Information

Allan Schaffer

Graphics and Game Technologies Evangelist
aschaffer@apple.com

Filip Iliescu

Graphics and Game Technologies Evangelist
filiescu@apple.com

Documentation

SceneKit Framework Reference
<http://developer.apple.com>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

-
- What's New in SpriteKit Pacific Heights Wednesday 2:00PM
 - What's New in SceneKit Pacific Heights Thursday 10:15AM
-

Labs

-
- SceneKit Lab

Graphics and
Games Lab A

Thursday 12:45PM

 WWDC14