Macoun

„Ich halte meine Vorlesungen immer am Freitag von 16 bis 18 Uhr.

Dann weiss ich, dass nur diejenigen Hörer kommen, die das auch interessiert.”

Prof. Dr. Klaus G. Troitzsch, Universität Koblenz, 1991 zur Vorlesung „Modellbildung und Simulation in den Sozialwissenschaften”

Scene Kit „Interaktiv”

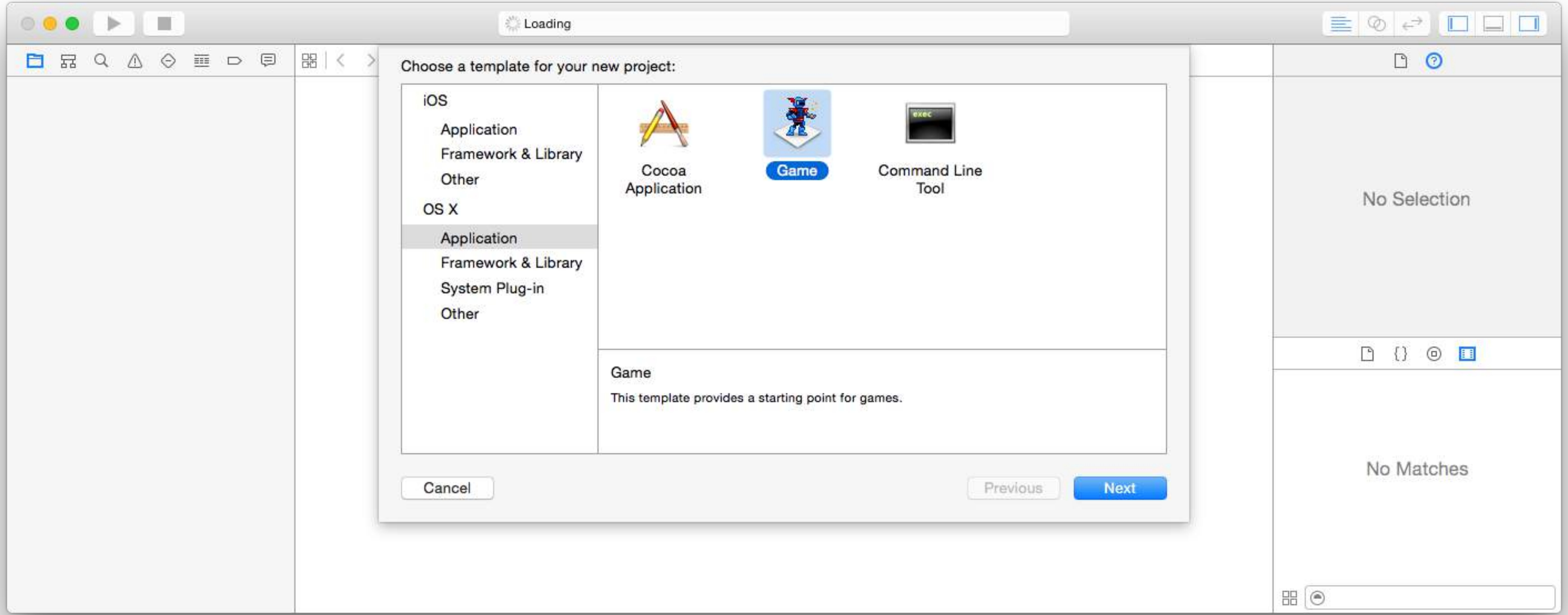
Christoph Wick

~~Scene Kit~~

~~SceneKit is an Objective-C (and Swift) framework for building apps and games that use 3D graphics, combining a high-performance rendering engine with a high-level, descriptive API.~~

~~SceneKit supports the import, manipulation, and rendering of 3D assets.~~

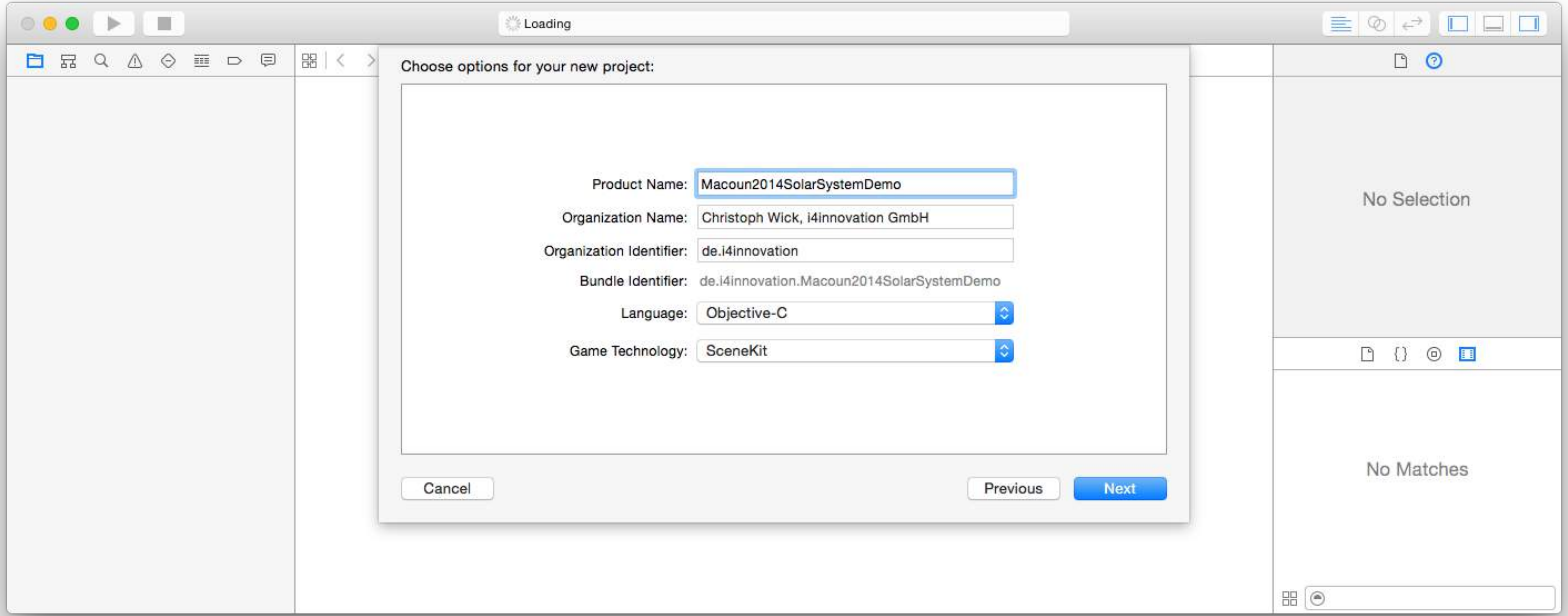
~~https://developer.apple.com/library/ios/documentation/SceneKit/Reference/SceneKit_Framework/index.html~~



Choose a template for your new project:

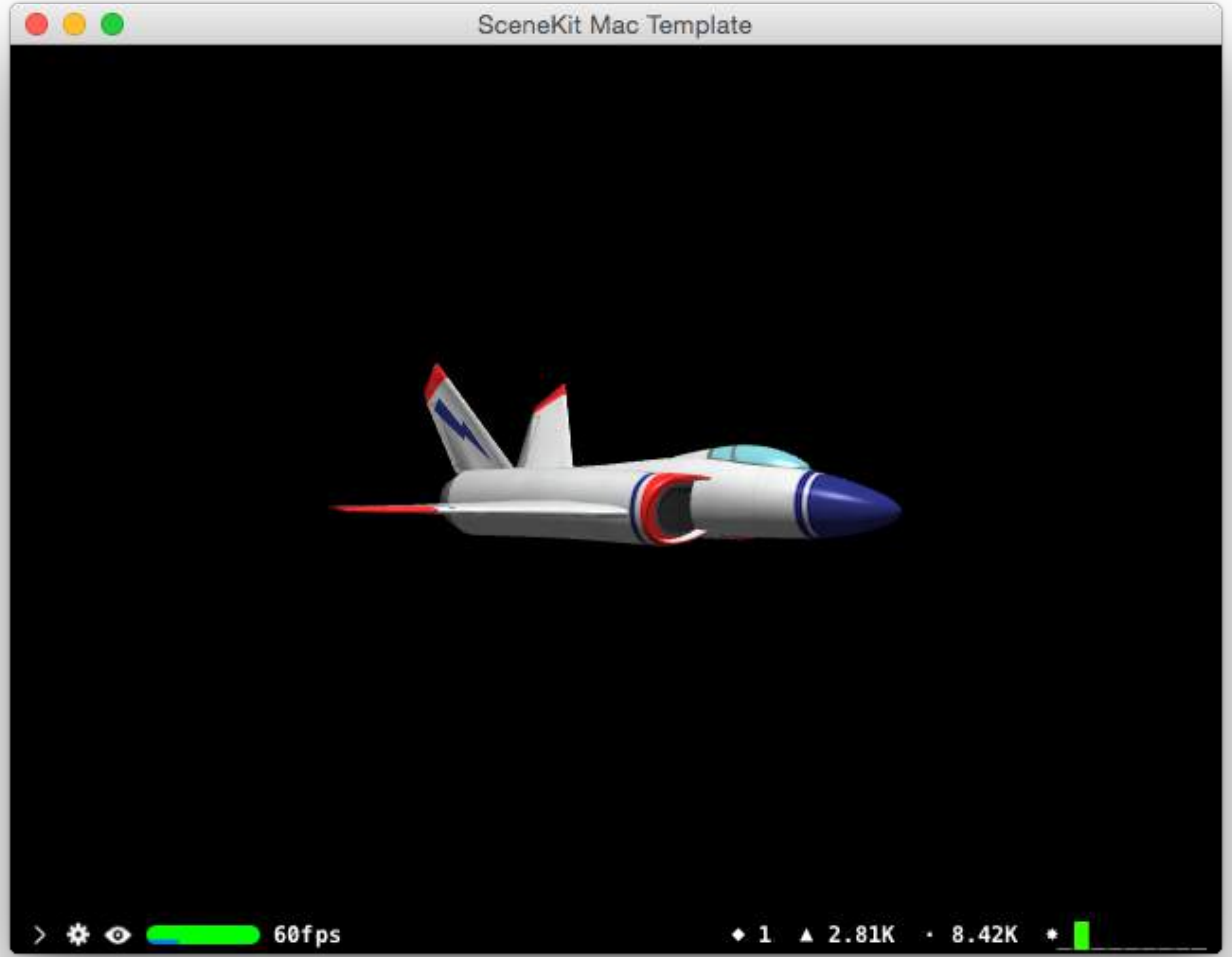
Game

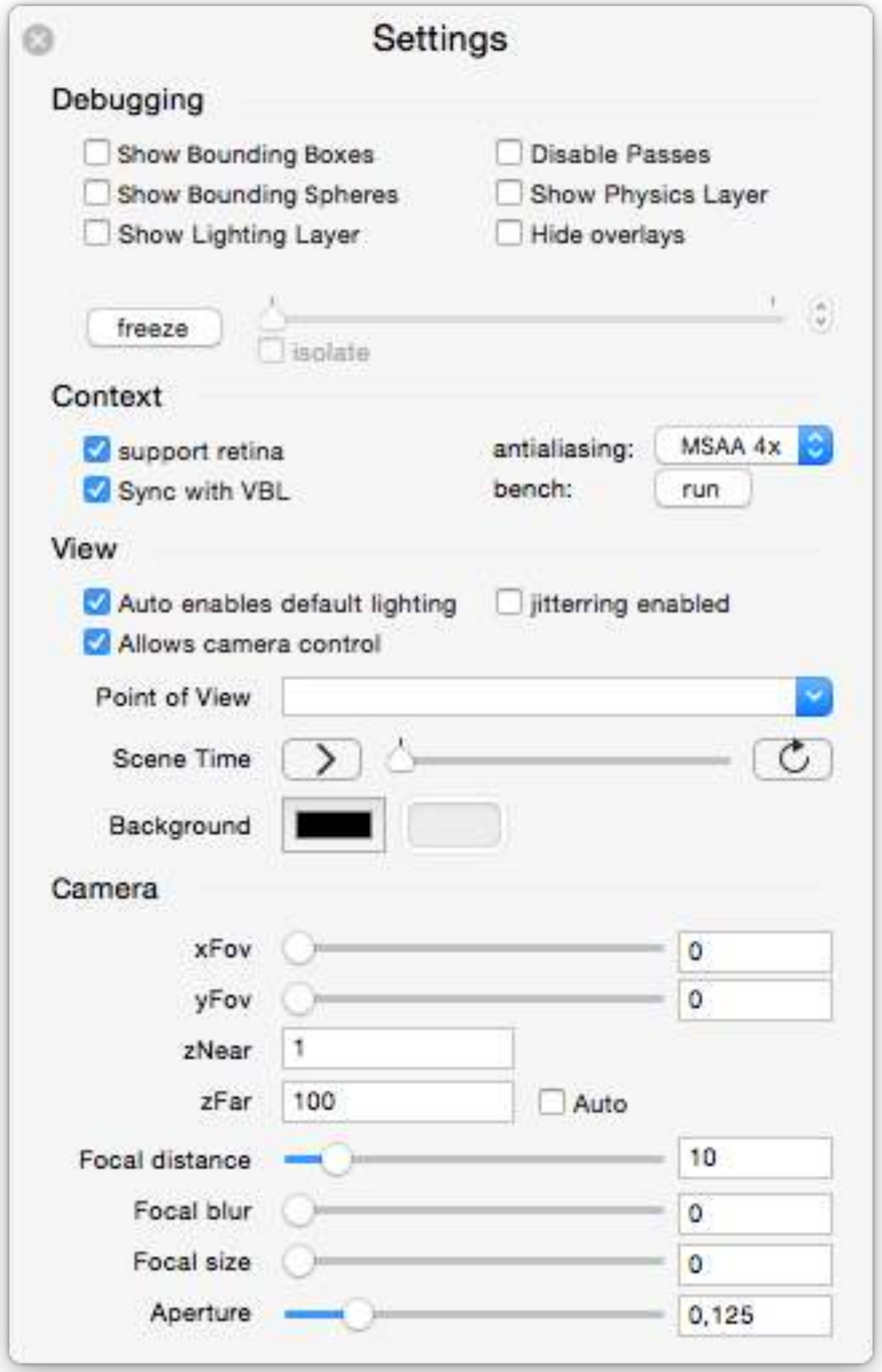
This template provides a starting point for games.



Choose options for your new project:

Product Name: Macoun2014SolarSystemDemo
Organization Name: Christoph Wick, i4innovation GmbH
Organization Identifier: de.i4innovation
Bundle Identifier: de.i4innovation.Macoun2014SolarSystemDemo
Language: Objective-C
Game Technology: SceneKit




Settings

Debugging

☐ Show Bounding Boxes
☐ Show Bounding Spheres
☐ Show Lighting Layer
☐ Disable Passes
☐ Show Physics Layer
☐ Hide overlays

freeze ☐ isolate

Context

☑ support retina
☑ Sync with VBL
antialiasing: MSAA 4x
bench: run

View

☑ Auto enables default lighting
☑ Allows camera control
☐ jitterring enabled

Point of View

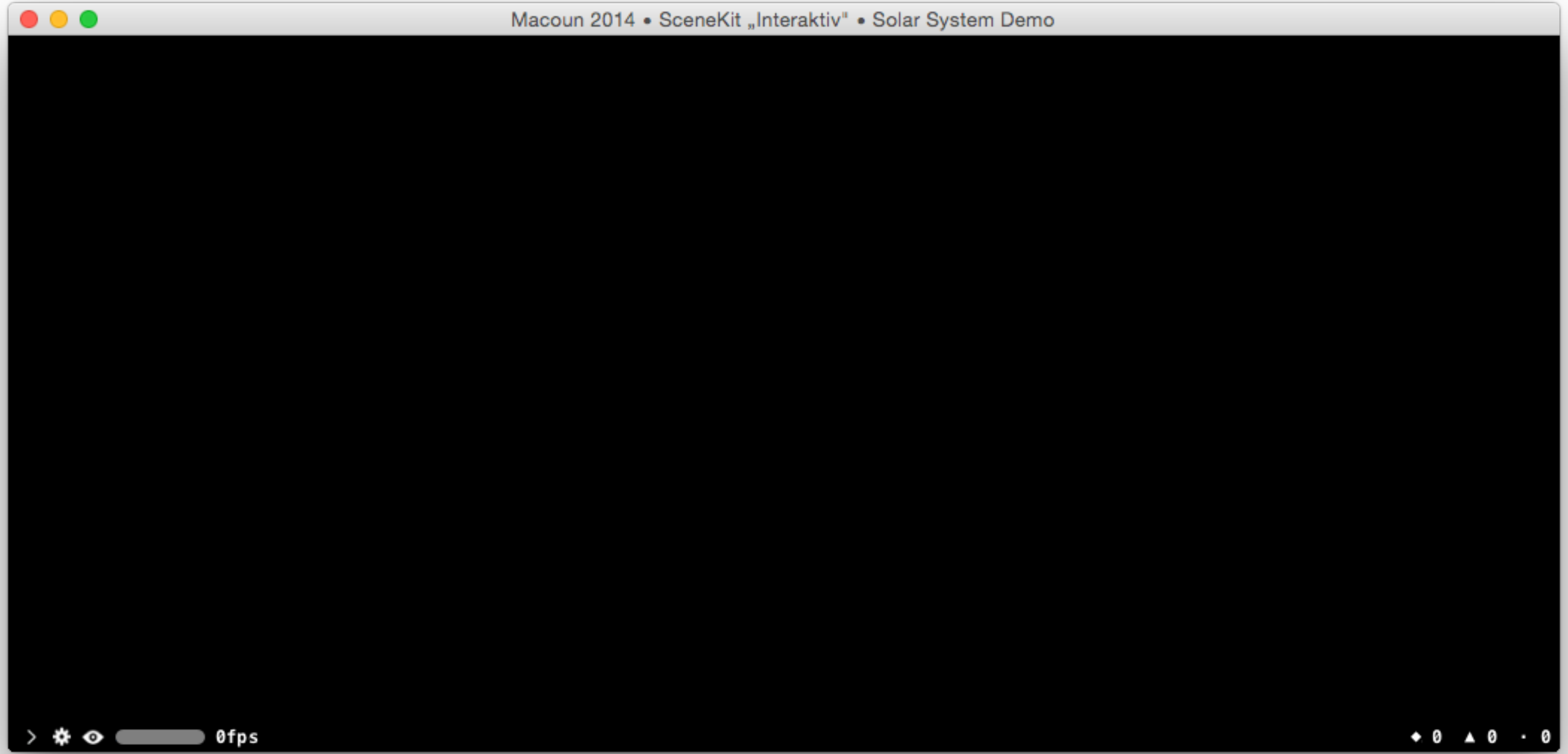
Scene Time

Background

Camera

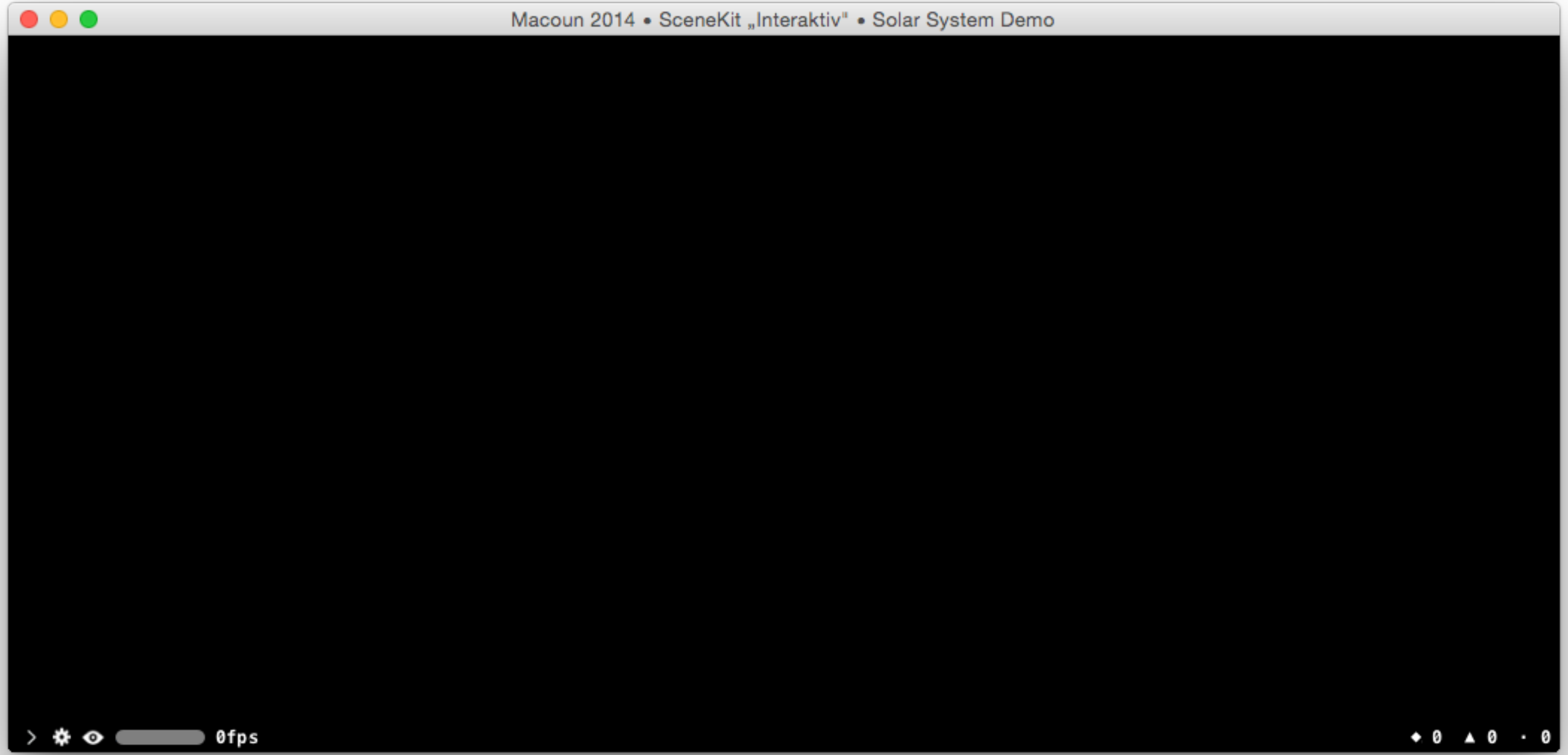
xFov 0
yFov 0
zNear 1
zFar 100 ☐ Auto
Focal distance 10
Focal blur 0
Focal size 0
Aperture 0,125

```
@implementation GameViewController

-(void)awakeFromNib
{
    // create a new scene
    SCNScene *scene = [SCNScene scene];

    // create and add a camera to the scene
    SCNNode *cameraNode = [SCNNode node];
    cameraNode.camera = [SCNCamera camera];
    cameraNode.camera.xFov = 45.0;
    cameraNode.position = SCNVector3Make(0, 2, 15);
    cameraNode.rotation = SCNVector4Make(1.0, 0.0, 0.0, -M_PI/20);

    [scene.rootNode addChildNode:cameraNode];
}
```



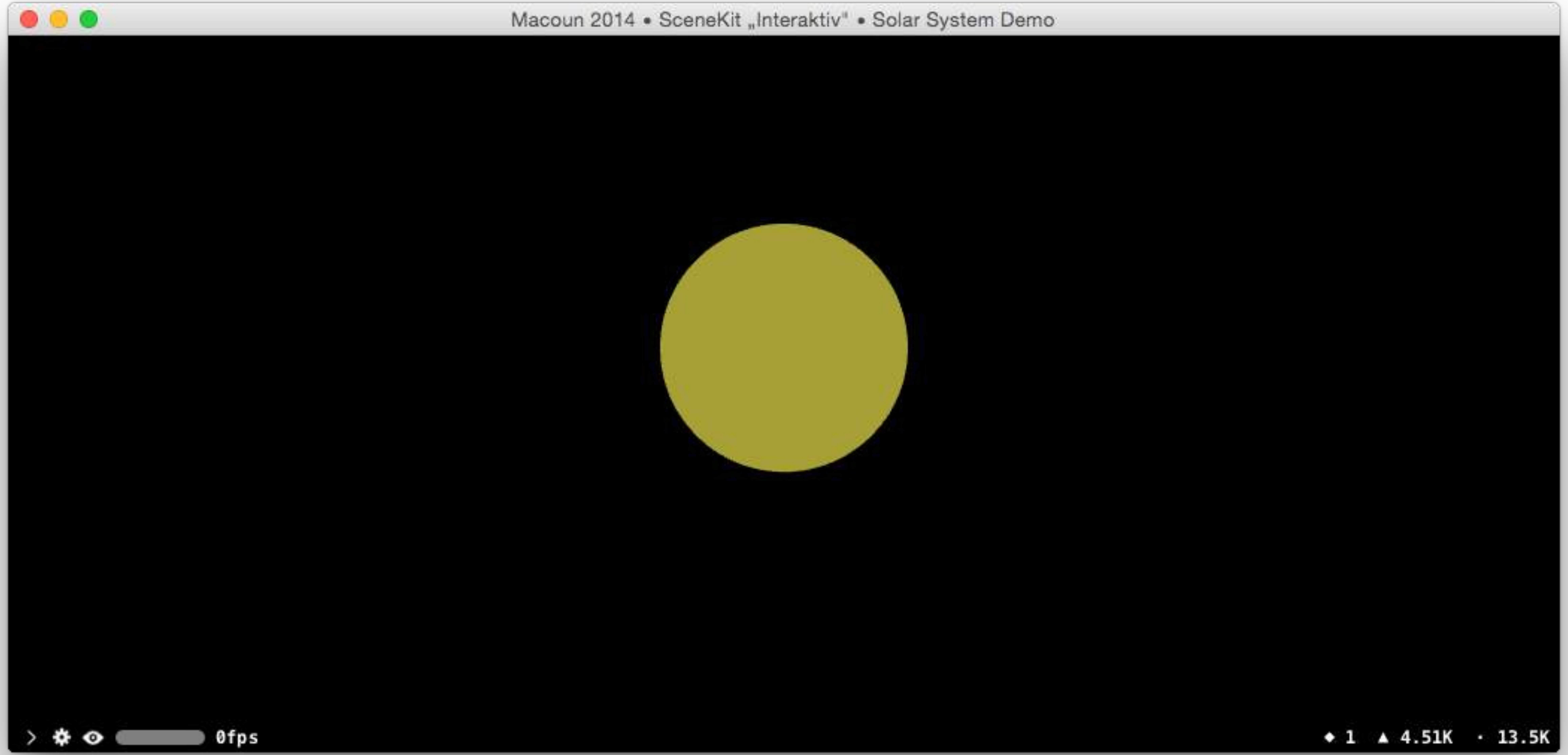
Macoun 2014 • SceneKit „Interaktiv" • Solar System Demo

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...
   // create and add the light
   // that comes out of the sun in every direction
   SCNNode *sunLightNode = [SCNNode node];
   sunLightNode.light = [SCNLight light];
   sunLightNode.light.type = SCNLightTypeOmni;
   sunLightNode.light.color =
      [NSColor colorWithCalibratedRed:1.0 green:1.0 blue:1.0 alpha:0.0];
   sunLightNode.position = SCNVector3Make(0, 0, 0);
   [scene.rootNode addChildNode:sunLightNode];
}
```

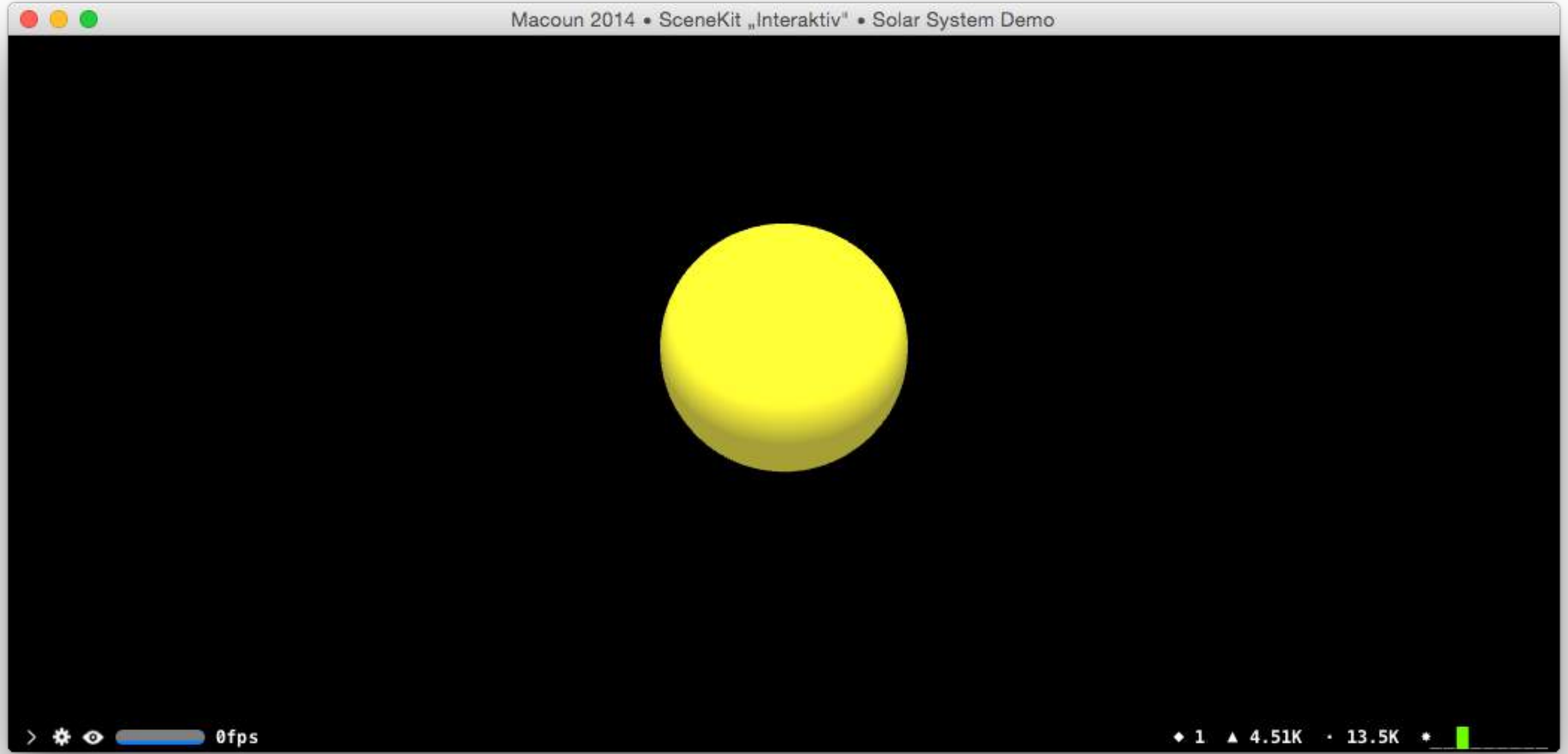


Macoun 2014 • SceneKit „Interaktiv“ • Solar System Demo

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...

   // Create the sphere geometry and node
   SCNSphere *sun = [SCNSphere sphereWithRadius:1.0];

   sun.firstMaterial.emission.contents =
      [NSColor colorWithDeviceCyan:0.0 magenta:0.0
                           yellow:1.0 black:0.5 alpha:0.5];

   SCNNode *sunNode = [SCNNode nodeWithGeometry:sun];
   sunNode.position = SCNVector3Make(0.0, 0.0, 0.0);
   [scene.rootNode addChildNode:sunNode];
}
```

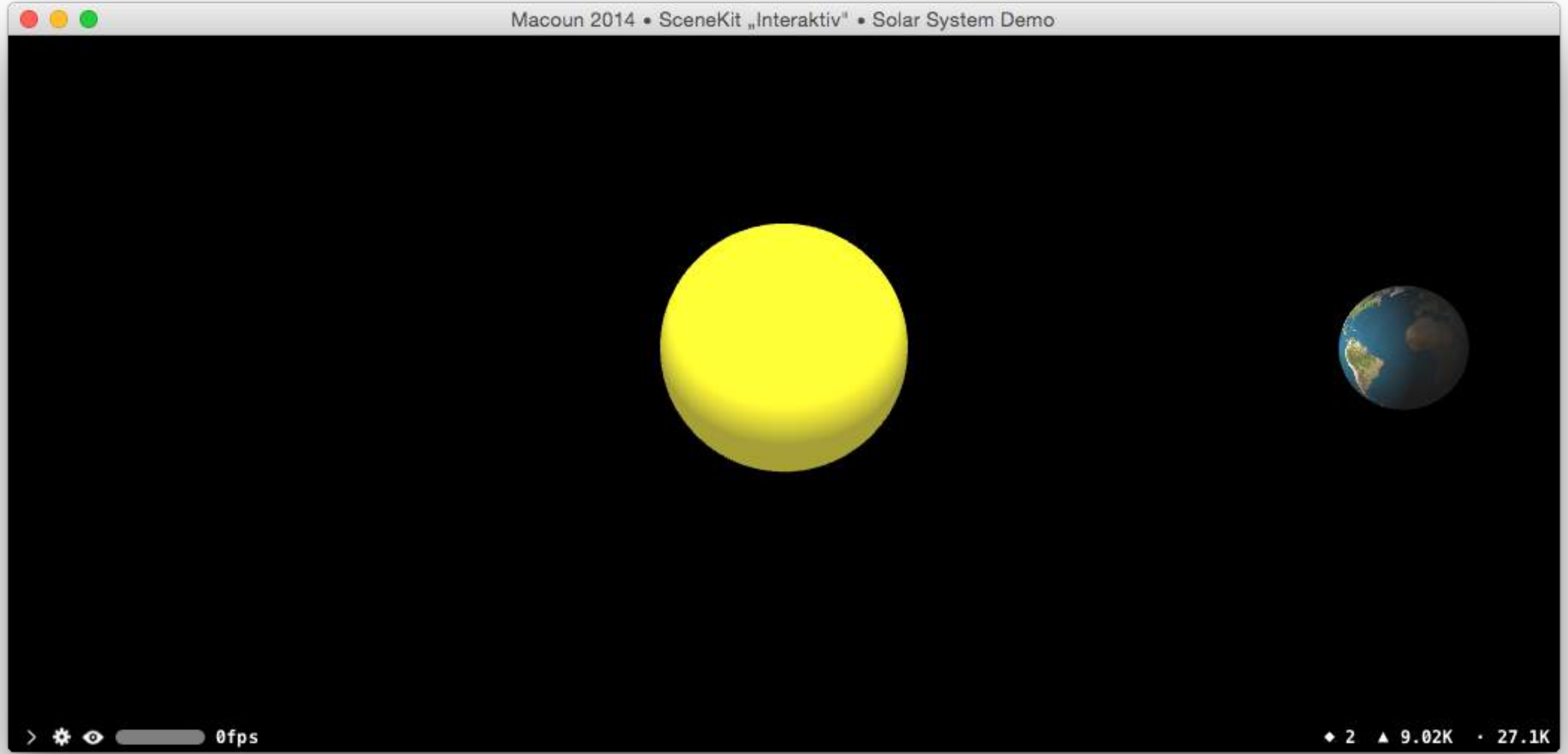

Macoun 2014 • SceneKit „Interaktiv" • Solar System Demo
0fps
◆ 1
▲ 4.51K
· 13.5K

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...

   // create and add an additional light that lights the sun
   SCNNode *sunLightNode2 = [SCNNode node];
   sunLightNode2.light = [SCNLight light];
   sunLightNode2.light.type = SCNLightTypeSpot;
   sunLightNode2.light.color =
      [NSColor colorWithCalibratedRed:1.0
                  green:1.0 blue:1.0 alpha:0.0];

   sunLightNode2.position = SCNVector3Make(0, 10, 10);
   sunLightNode2.rotation = SCNVector4Make(1.0, 0.0, 0.0, -M_PI_4);
   [scene.rootNode addChildNode:sunLightNode2];
}
```



Macoun 2014 • SceneKit „Interaktiv" • Solar System Demo

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...
   SCNNode *earthAroundSunRotationNode = [SCNNode node];
   earthAroundSunRotationNode.position = SCNVector3Make(0.0, 0.0, 0.0);
   [scene.rootNode addChildNode:earthAroundSunRotationNode];

   SCNNode *earthFromSunTranslationNode = [SCNNode node];
   earthFromSunTranslationNode.position = SCNVector3Make(5.0, 0.0, 0.0);
   [earthAroundSunRotationNode addChildNode:earthFromSunTranslationNode];

   SCNNode *earthAroundItselfRotationNode = [SCNNode node];
   [earthFromSunTranslationNode addChildNode:earthAroundItselfRotationNode];

   SCNSphere *earth = [SCNSphere sphereWithRadius:0.5];
   earth.firstMaterial.diffuse.contents = [NSImage imageNamed:@"earthmap1k.jpg"];
   earthAroundItselfRotationNode.geometry = earth;
}
```


Macoun 2014 • SceneKit „Interaktiv" • Solar System Demo
0fps
◆ 2 ▲ 9.02K · 27.1K

```
@implementation GameViewController

-(void)awakeFromNib
{
    ...
   // animate earth around sun and around itself
   CABasicAnimation *yearAnimation =

      [CABasicAnimation animationWithKeyPath:@"rotation"];

   yearAnimation.toValue =
         [NSValue valueWithSCNVector4:SCNVector4Make(0, -1, 0, M_PI*2)];

   yearAnimation.duration = 10;
   yearAnimation.repeatCount = MAXFLOAT; //repeat forever

   [earthAroundSunRotationNode addAnimation:yearAnimation forKey:nil];

}
```

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...
   SCNNode *moonAroundEarthRotationNode = [SCNNode node];
   [earthFromSunTranslationNode addChildNode:moonAroundEarthRotationNode];

   SCNNode *moomFromEarthTranslatioNode = [SCNNode node];
   moomFromEarthTranslatioNode.position = SCNVector3Make(1.0, 0.0, 0.0);
   [moonAroundEarthRotationNode addChildNode:moomFromEarthTranslatioNode];

   SCNNode *moonNode = [SCNNode node];
   SCNSphere *moonGeometry = [SCNSphere sphereWithRadius:0.2];
   moonGeometry.firstMaterial.diffuse.contents =
      [NSImage imageNamed:@"moonmap1k.jpg"];

   moonNode.geometry = moonGeometry;

   [moomFromEarthTranslatioNode addChildNode:moonNode];
```

```
@implementation GameViewController

-(void)awakeFromNib
{
   ...
   CABasicAnimation *monthAnimation =
      [CABasicAnimation animationWithKeyPath:@"rotation"];

   monthAnimation.toValue =
      [NSValue valueWithSCNVector4:SCNVector4Make(0, 1, 0, M_PI*2)];

   monthAnimation.duration = 3;
   monthAnimation.repeatCount = MAXFLOAT; //repeat forever

   [moonAroundEarthRotationNode addAnimation:monthAnimation forKey:nil];

}
```

Der Vollständigkeit halber …

```
@implementation GameViewController

-(void)awakeFromNib
{
    ...
    // set the scene to the view
    self.gameView.scene = scene;

    // allows the user to manipulate the camera
    self.gameView.allowsCameraControl = NO;

    // show statistics such as fps and timing information
    self.gameView.showsStatistics = YES;

    // configure the view
    self.gameView.backgroundColor = [NSColor blackColor];
}
```

Scene Kit

SceneKit is an Objective-C (and Swift) framework for building apps and games that use 3D graphics, combining a high-performance rendering engine with a high-level, descriptive API.

SceneKit supports the import, manipulation, and rendering of 3D assets.

https://developer.apple.com/library/ios/documentation/SceneKit/Reference/SceneKit_Framework/index.html

Vorteile von SceneKit

- stammt von Apple → beste Integration in OS X und iOS
  - wird aktuell von Apple gepflegt
- in Objective-C und Swift programmierbar
- mit anderen Apple-Technologien zusammen verwendbar
  - z.B. SpriteKit, CoreAnimation
  - auf OS X: als View in ein GUI einbettbar
- kostenlos verwendbar

Alternativen zu SceneKit

1. Open GL / Metal
2. Open Inventor / Coin3D
3. Game-Engines, z.B. Unity3d

Open GL / Metal

PRO

- low-level APIs
  - sehr hardware-nah
- cross-platform (Open GL)

KONTRA

- low-level APIs
  - sehr hardware-nah
- Apple-only (Metal)

Open Inventor / Coin3D

PRO

- „Mutter” aller szenegraph-basierten 3D-APIs
- „cross-platform”
- extrem ausgereift
- im wissenschaftlich/technischen Bereich weit verbreitet

KONTRA

- C++ / Qt
- nicht für iOS
- (vom Vortragenden „gefühlte”) ungewisse Zukunft
- kein Physics-Engine oder Partikel-System

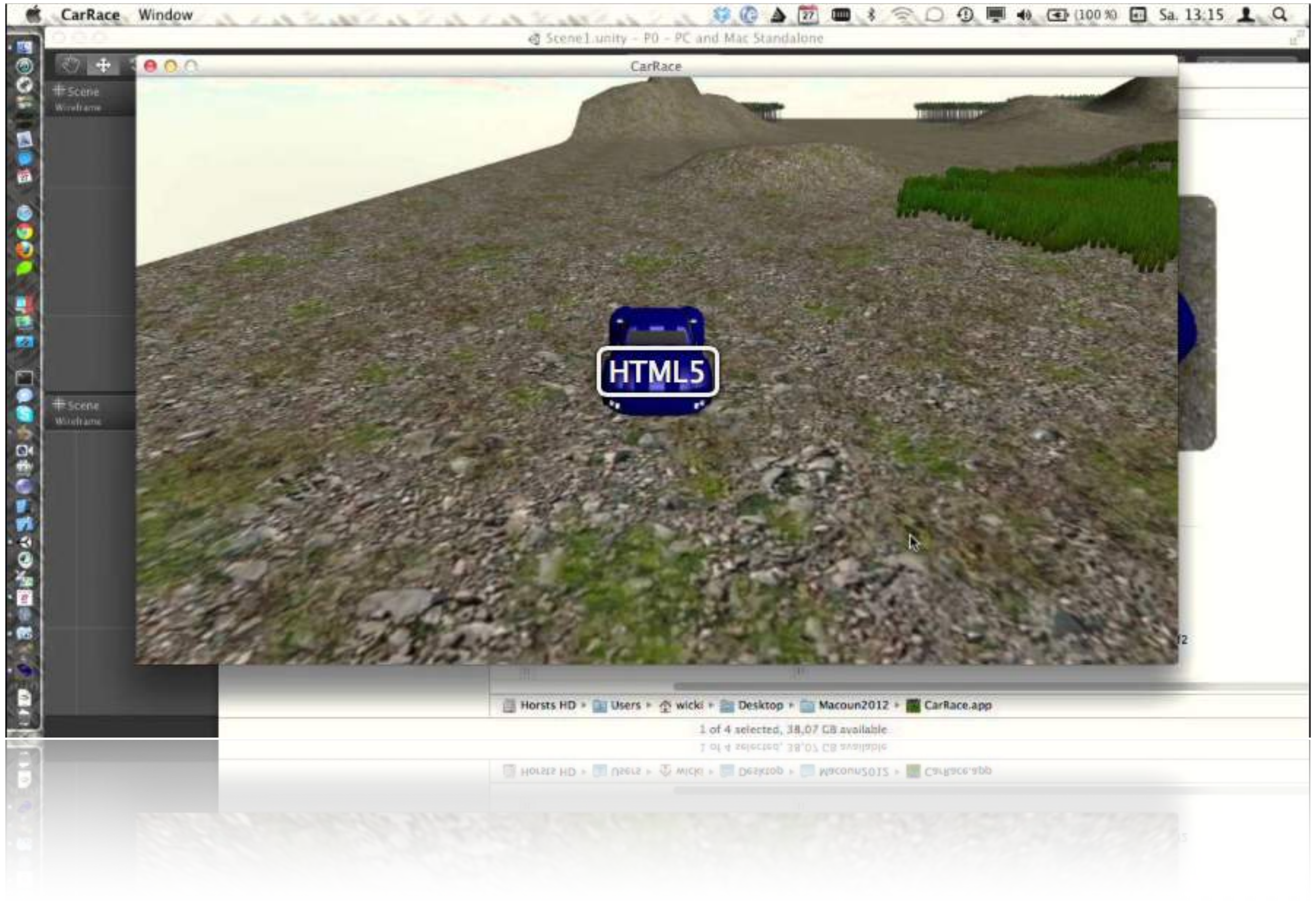
Game-Engines, z.B. Unity3d

PRO

- „cross-platform”
- sehr ausgereift
- weit verbreitet
- Physics-Engine und Partikel-System

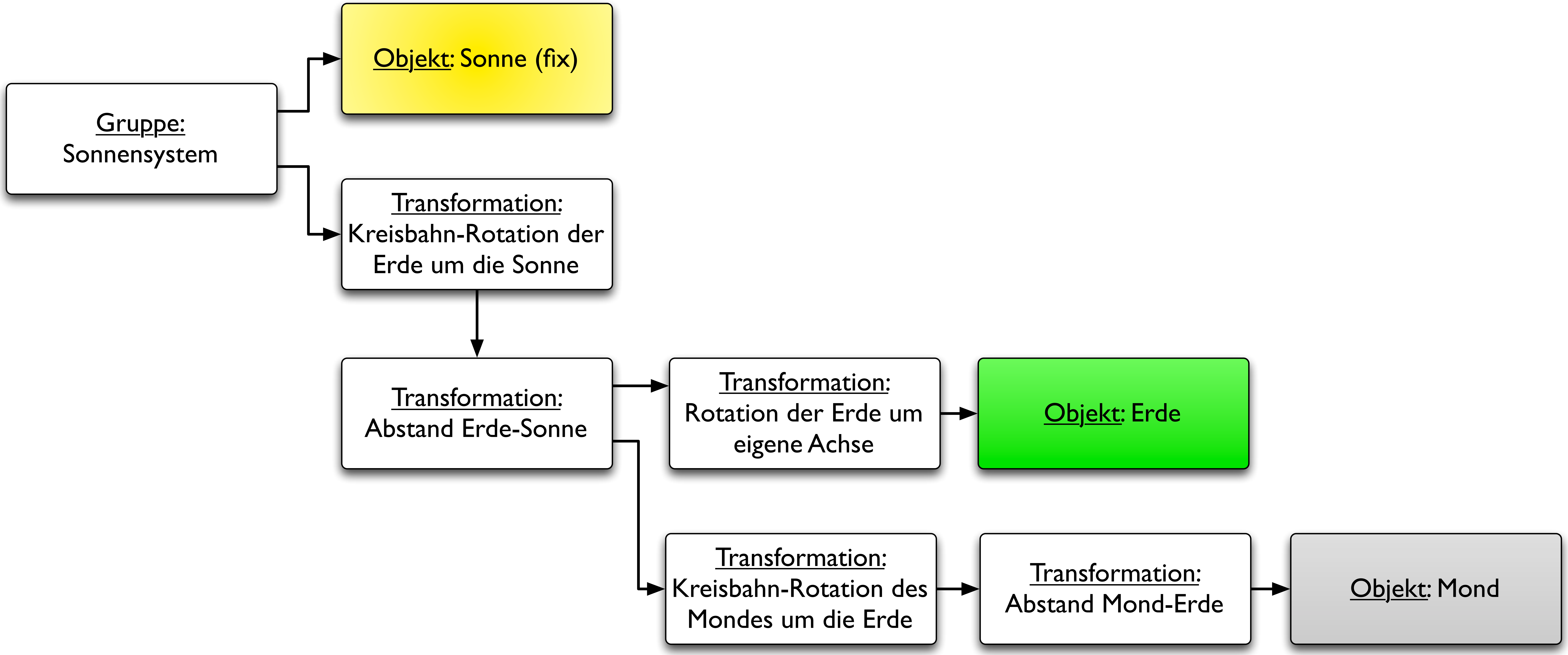
KONTRA

- „cross-platform”
- in C# (u.a.) zu programmieren
- „Single-View“-Anwendungen (auch auf OS X)
- nicht kostenlos


CarRace
Window
Sa. 13:15
Scene1.unity - P0 - PC and Mac Standalone
CarRace
Scene
Wireframe
HTML5
Horsts HD ▸ Users ▸ wicki ▸ Desktop ▸ Macoun2012 ▸ CarRace.app
1 of 4 selected, 38,07 GB available

https://macoun.de/video2012kssa2.php

Szene-Graphen



Gruppe: Sonnensystem

Objekt: Sonne (fix)

Transformation: Kreisbahn-Rotation der Erde um die Sonne

Transformation: Abstand Erde-Sonne

Transformation: Rotation der Erde um eigene Achse

Objekt: Erde

Transformation: Kreisbahn-Rotation des Mondes um die Erde

Transformation: Abstand Mond-Erde

Objekt: Mond

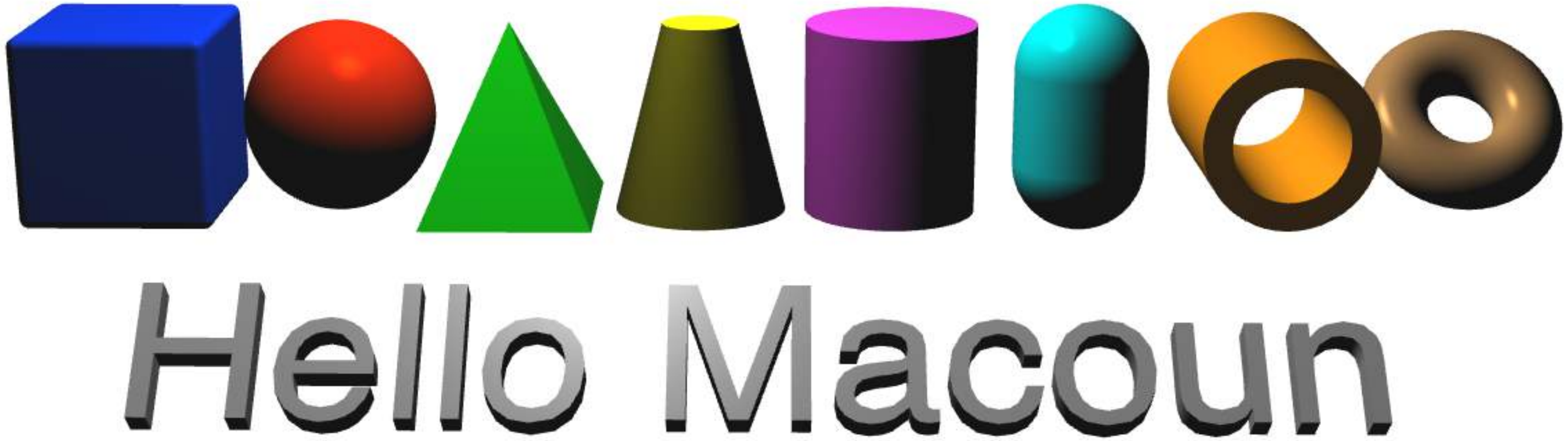
Nodes

- SCNScene
- SCNNode
- SCNGeometry
- SCNLight
- SCNCamera

SCNGeometry

- Primitive:

  SCNBox, SCNSphere, SCNPyramid, SCNCone, SCNCylinder, SCNCapsule, SCNTube, SCNTorus, SCNText …



Licht

- **Ambient**: „überall”-Licht
- **Omni**: aus einer Punkt-Quelle in alle Richtungen
- **Directional**: überall, in eine Richtung
- **Spot:** ein begrenzter Strahl in eine Richtung

Neu in SceneKit 2014: **Schatten** werden von **Spotlight** und **DirectonalLight** geworfen

Physics

- SceneKit hat eine eingebaute Physics-Engine
- simuliert:
    - Schwerkraft
    - Kollisionen

Einen Boden …

```
@implementation GameViewController

-(void)awakeFromNib
{
    ...
   SCNFloor *floorGeometry = [SCNFloor floor];
   floorGeometry.reflectivity = 0.01;
   floorGeometry.firstMaterial.diffuse.contents =
      [NSImage imageNamed:@"Grid.png"];
   SCNNode *floorNode = [SCNNode nodeWithGeometry:floorGeometry];
   floorNode.physicsBody = [SCNPhysicsBody staticBody];
   [scene.rootNode addChildNode:floorNode];
}
```

und eine fallende Kugel

```
@implementation GameView

- (void)keyDown:(NSEvent *)theEvent
{
   ...
   SCNSphere *sphereGeometry = [SCNSphere sphereWithRadius:2.0];
   sphereGeometry.firstMaterial.diffuse.contents =
      [NSImage imageNamed:@"billard.png"];
   SCNNode *sphereNode = [SCNNode nodeWithGeometry:sphereGeometry];

   sphereNode.position = SCNVector3Make(0.0,20.0,0.0);
   sphereNode.physicsBody = [SCNPhysicsBody dynamicBody];
   sphereNode.physicsBody.mass = 10;
   sphereNode.physicsBody.velocity = SCNVector3Make(0, -15, 0);

   [self.scene.rootNode addChildNode:sphereNode];
```

falls die Physik zu „langsam” ist

```
scene.physicsWorld.speed = 4.0;
```

SCNPhysicsWorld definiert die physikalischen Eigenschaften einer

- speed (CGFloat)
- gravity (SCNVector3)

SCNPhysicsBody

- staticBody
- dynamicBody
- kinematicBody

SCNPhysicsVehicle

- simuliert ein Objekt, dass sich auf Rädern bewegt (z.B. ein Auto)
- benötigt einen SCNNode mit einem SCNDynamicBody
- und Räder vom Typ SCNPhysicsVehicleWheel

```
carPhysics = [SCNPhysicsVehicle vehicleWithChassisBody:carNode.physicsBody
  wheels:@[lfPhysicsWheel, rfPhysicsWheel,lrPhysicsWheel, rrPhysicsWheel]];
```

Hit test

- erlaubt direkte Interaktion mit dem SCNView, z.B in `mouseDown`
- schickt einen „Strahl” von der Maus durch die Szene
- liefert ein NSArray von `SCNHitTestResult` zurück

Hit test

```
-(void)mouseDown:(NSEvent *)theEvent
{
   // check what nodes are clicked
   NSPoint p = [self convertPoint:[theEvent locationInWindow] fromView:nil];
   NSArray *hitResults = [self hitTest:NSPointToCGPoint(p) options:nil];

   // check that we clicked on at least one object
   if([hitResults count] > 0){

      // retrieved the first clicked object
      for (SCNHitTestResult *result in hitResults) {

         SCNNode *resultNode = result.node;
```

Hit test

```
// retrieved the first clicked object

SCNNode *resultNode = hitResults.objectAtIndex(0).node;

   if ([resultNode.name isEqual: @"moreBallsNode"]) {
         [self makeSomething];
   } else { // look in parent nodes
      while (resultNode.parentNode != nil) {
         resultNode = resultNode.parentNode;
         if ([resultNode.name isEqual: @"moreBallsNode"]) {
            [self makeSomething];
         }
      }
   }
}
```

SCNHitTestResult

- @property(nonatomic, readonly) SCNVector3 localCoordinates
- @property(nonatomic, readonly) SCNVector3 worldCoordinates
- @property(nonatomic, readonly) SCNVector3 localNormal
- @property(nonatomic, readonly) SCNVector3 worldNormal

Tipps

„Minimal“-Team

1. ein 3D-Modeller (3D-Studio, Maya o.ä.)
2. ein Programmierer mit 3D-Affinität

Fragen?

<cw@i4innovation.de>

Vielen Dank

<cw@i4innovation.de>

Macoun

Quellen-Nachweis

- Texture der Erde und des Mondes:
  © James Hastings-Trew
  http://planetpixelemporium.com/planets.html

- karikatur, gott
  © dedMazay / canStockPhoto
  http://www.canstockphoto.de/karikatur-gott-9585999.html