

MBS SceneKit Plugin

Stefanie Juchmes

Stefanie Juchmes

Born in 1992

Bachelor of Science 2018

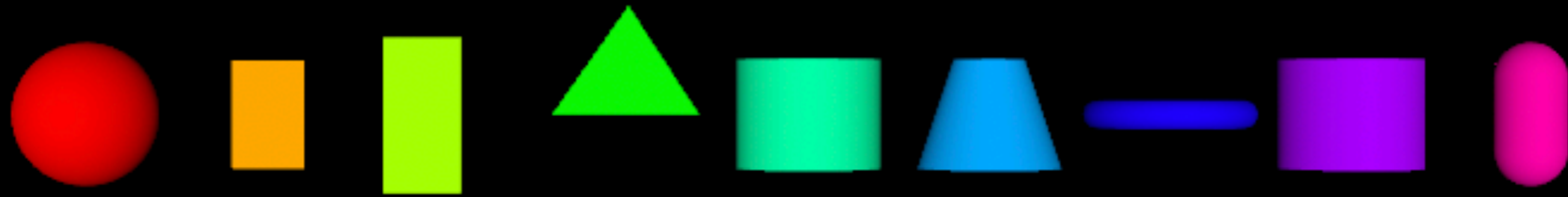
Since January 2019
working at Monkeybread
Software



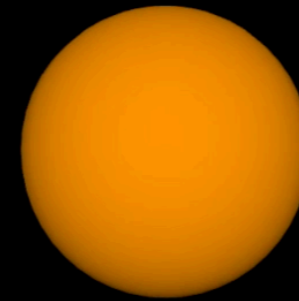
MBS Xojo Plugin

More than 65,000
functions



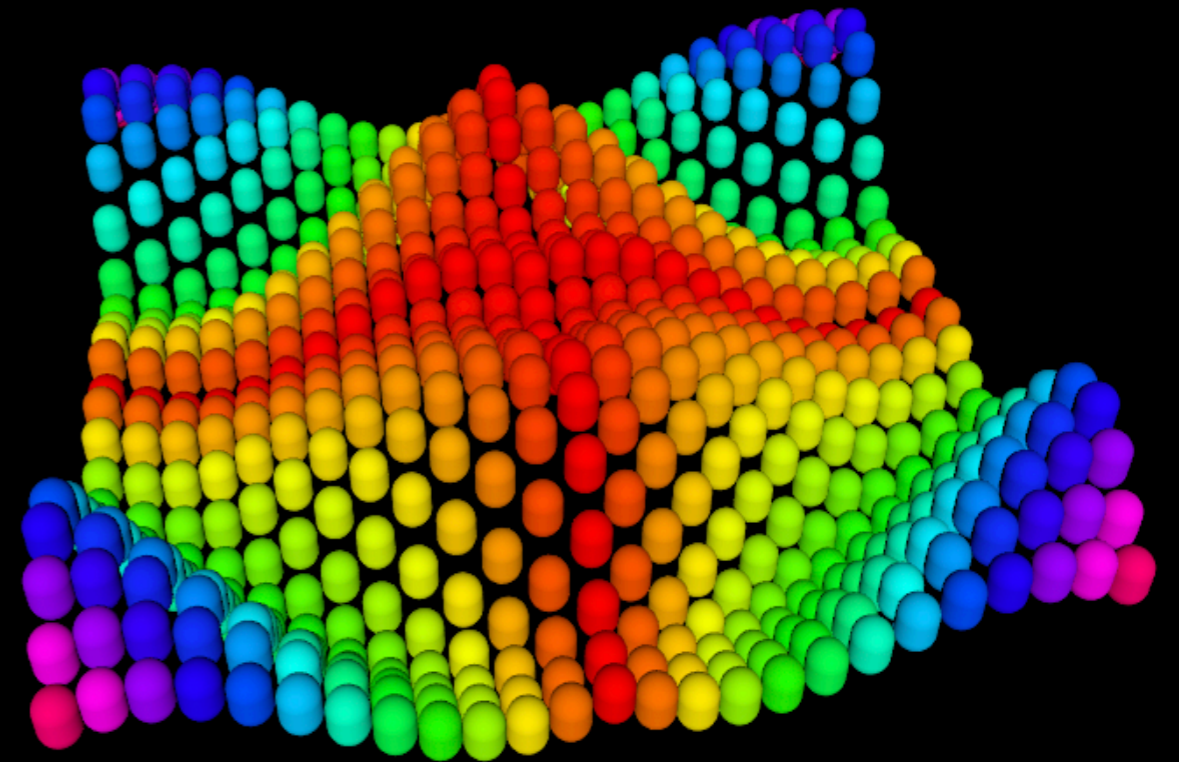


SceneKit



A powerful graphic engine
based on the apple
framework

49 classes
858 entries



SceneKit

- **Control, View and Scene**
- Geometries
- Material
- Lighting system
- Moving

SCNControlMBS

```
MyView = Me.View
```

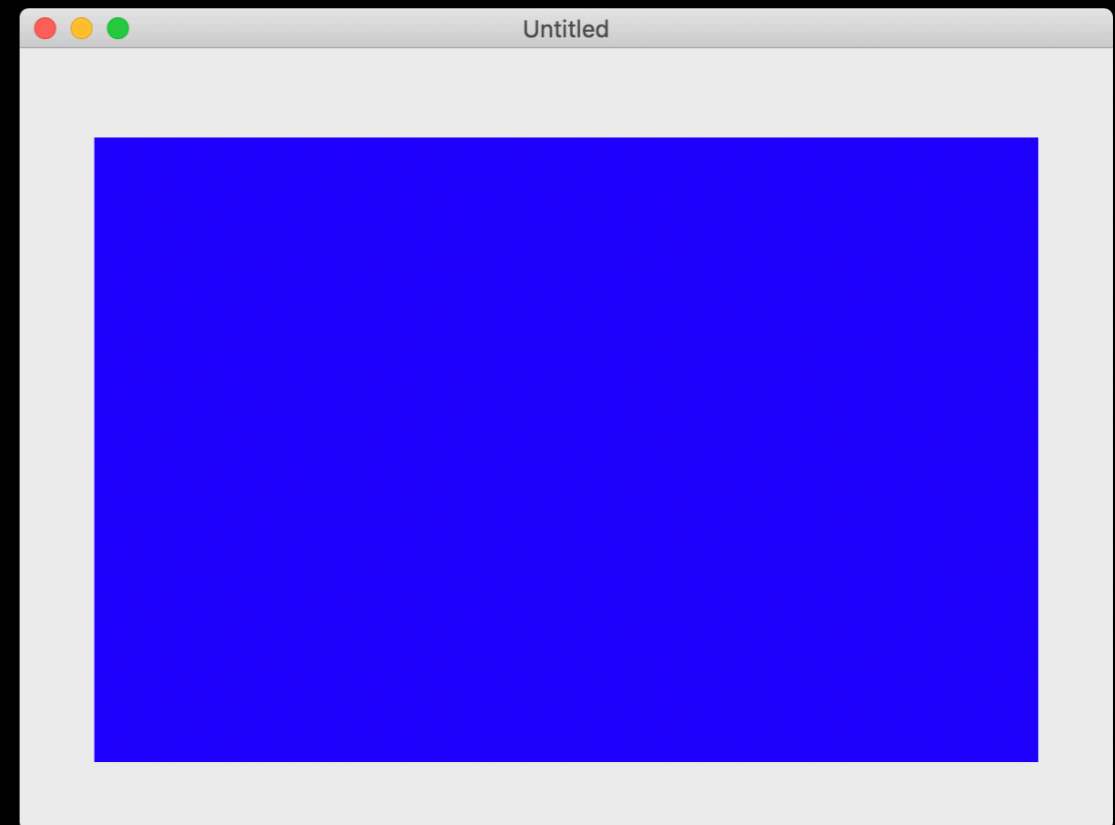
```
MyScene = New SCNSceneMBS
```

```
MyView.scene = MyScene
```

```
myView.autoenablesDefaultLighting = True
```

```
MyView.allowsCameraControl = True
```

```
myView.backgroundColor =  
NSColorMBS.blueColor
```

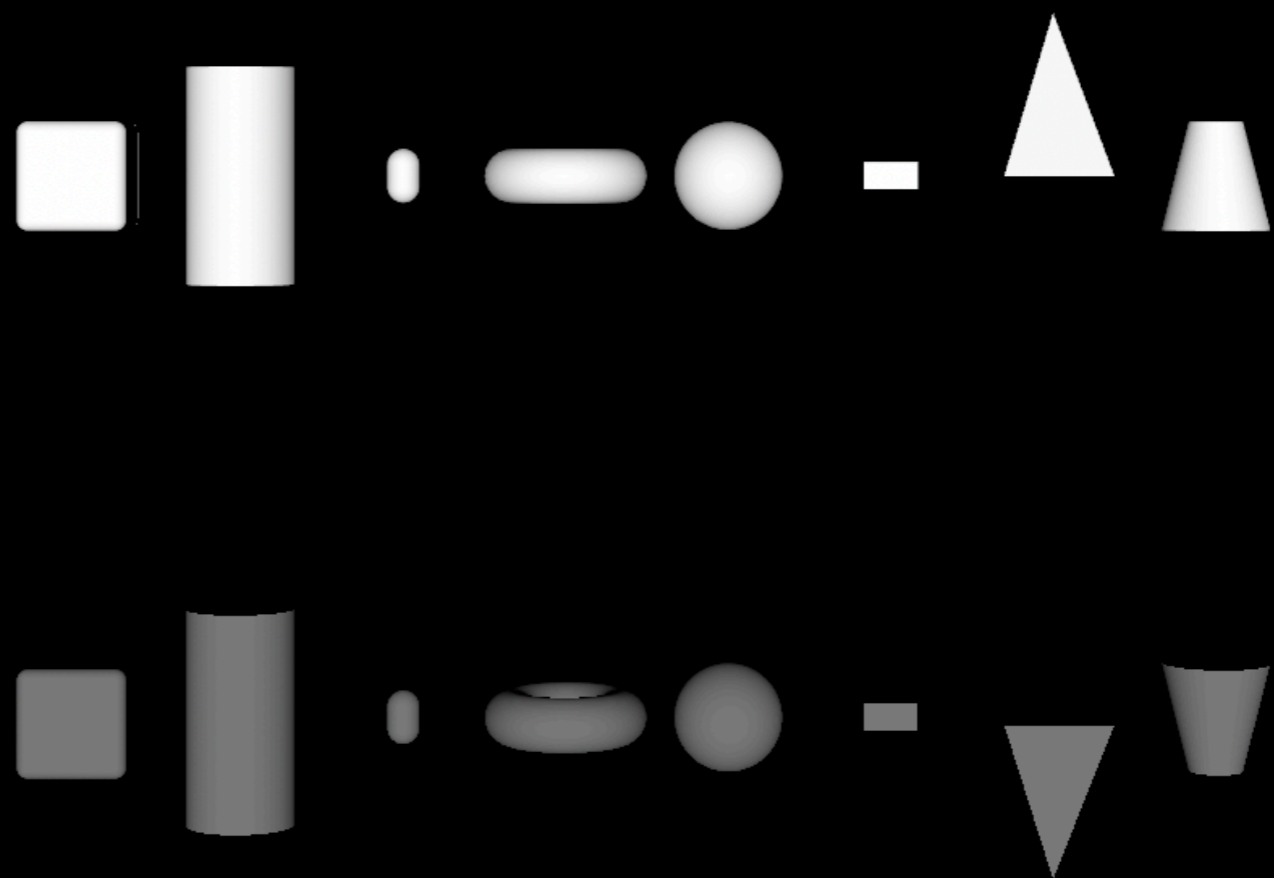


SceneKit

- Control, View and Scene
- **Geometries**
- Material
- Lighting system
- Moving

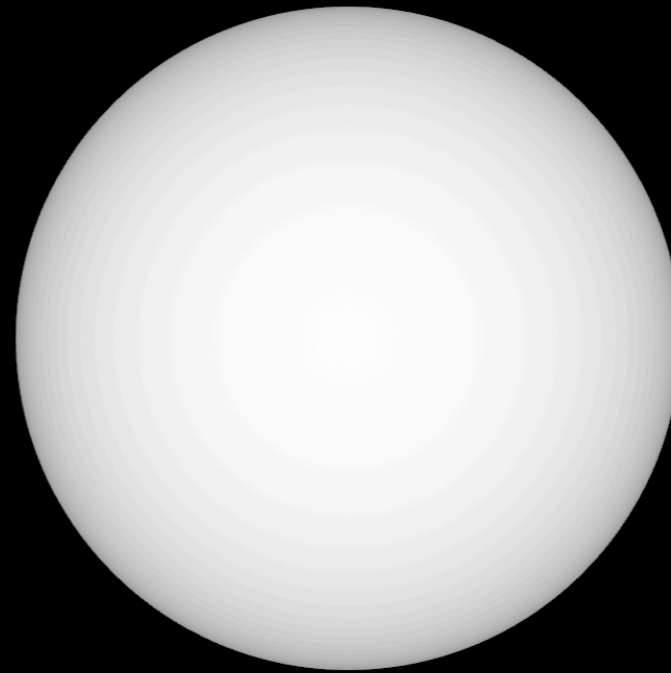
Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Dim radius *As Double* = 1

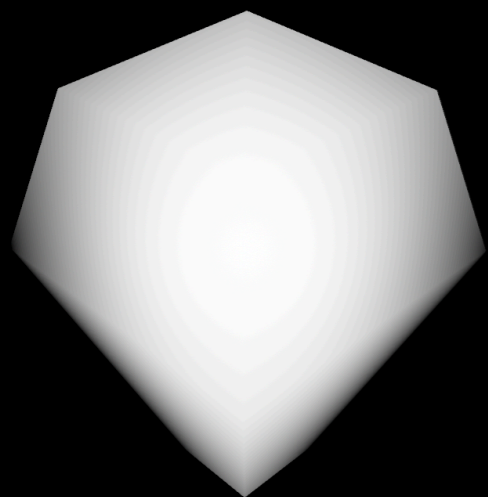
Dim SphereGeometry *As New* SCNSphereMBS (radius)

***SphereGeometry.segmentCount* = 100**

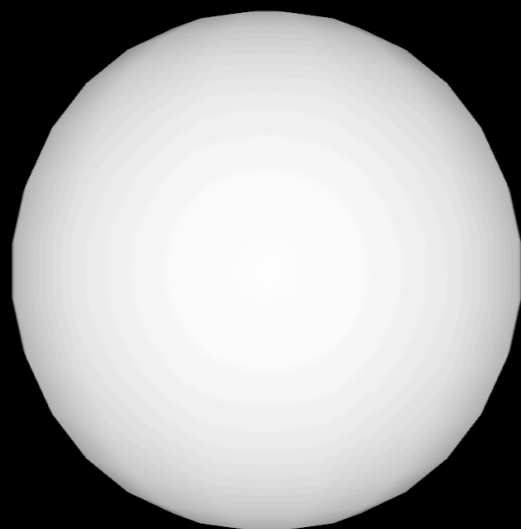
Dim Sphere *As New* SCNNodeMBS (SphereGeometry)

Sphere.Position = *New* SCNVector3MBS(0, 0, 0)

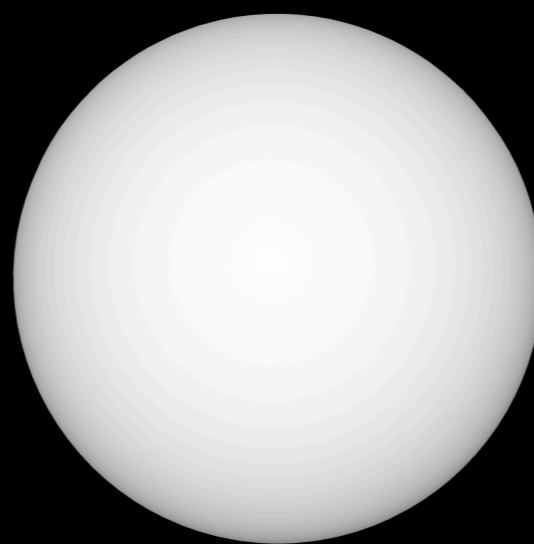
MyScene.rootNode.addChildNode (Sphere)



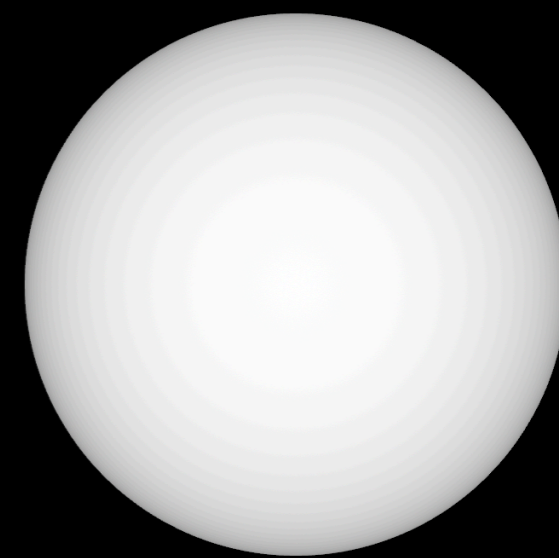
segmentCount = 4



segmentCount = 15



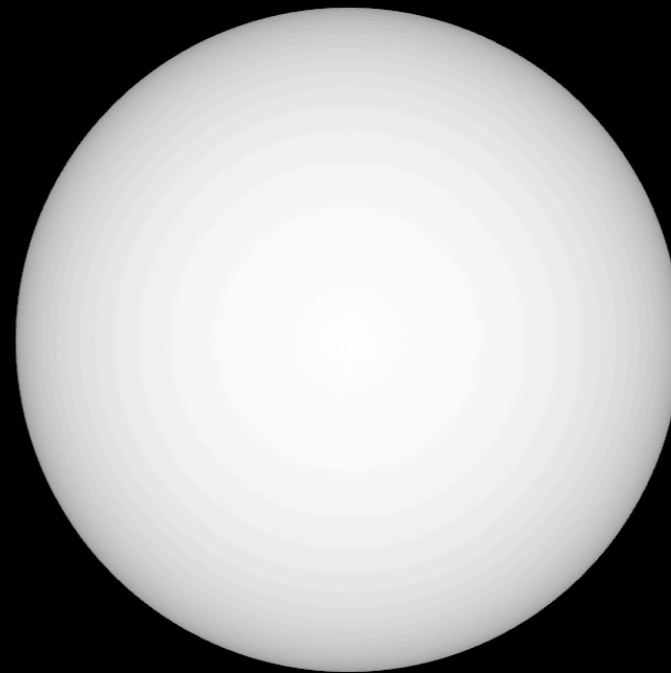
segmentCount = 48



segmentCount = 100

Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



`Dim radius As Double = 1`

`Dim SphereGeometry As New SCNSphereMBS (radius)`

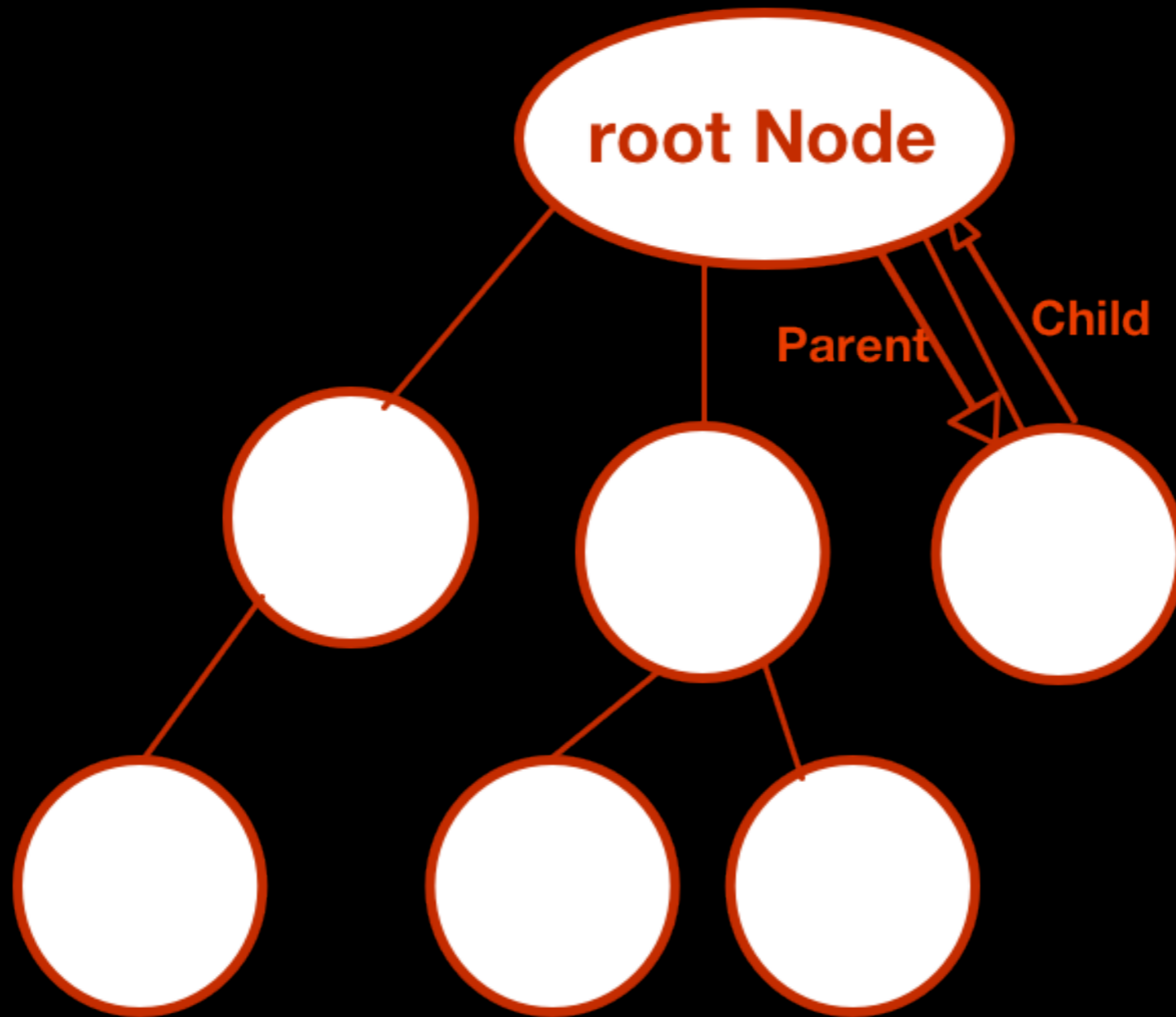
`SphereGeometry.segmentCount = 100`

`Dim Sphere As New SCNNodeMBS (SphereGeometry)`

`Sphere.Position = New SCNVector3MBS(0, 0, 0)`

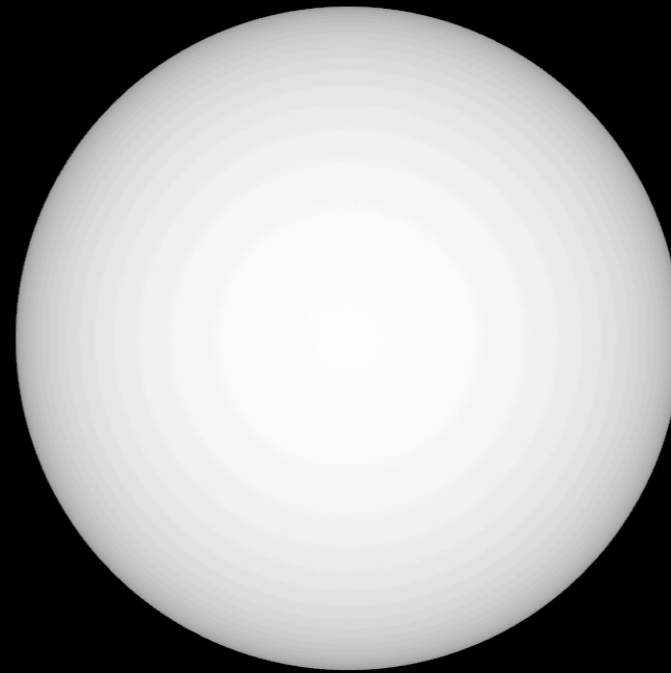
`MyScene.rootNode.addChildNode (Sphere)`

Nodes



Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Dim radius *As Double* = 1

Dim SphereGeometry *As New* SCNSphereMBS (radius)

SphereGeometry.segmentCount = 100

Dim Sphere *As New* SCNNodeMBS (SphereGeometry)

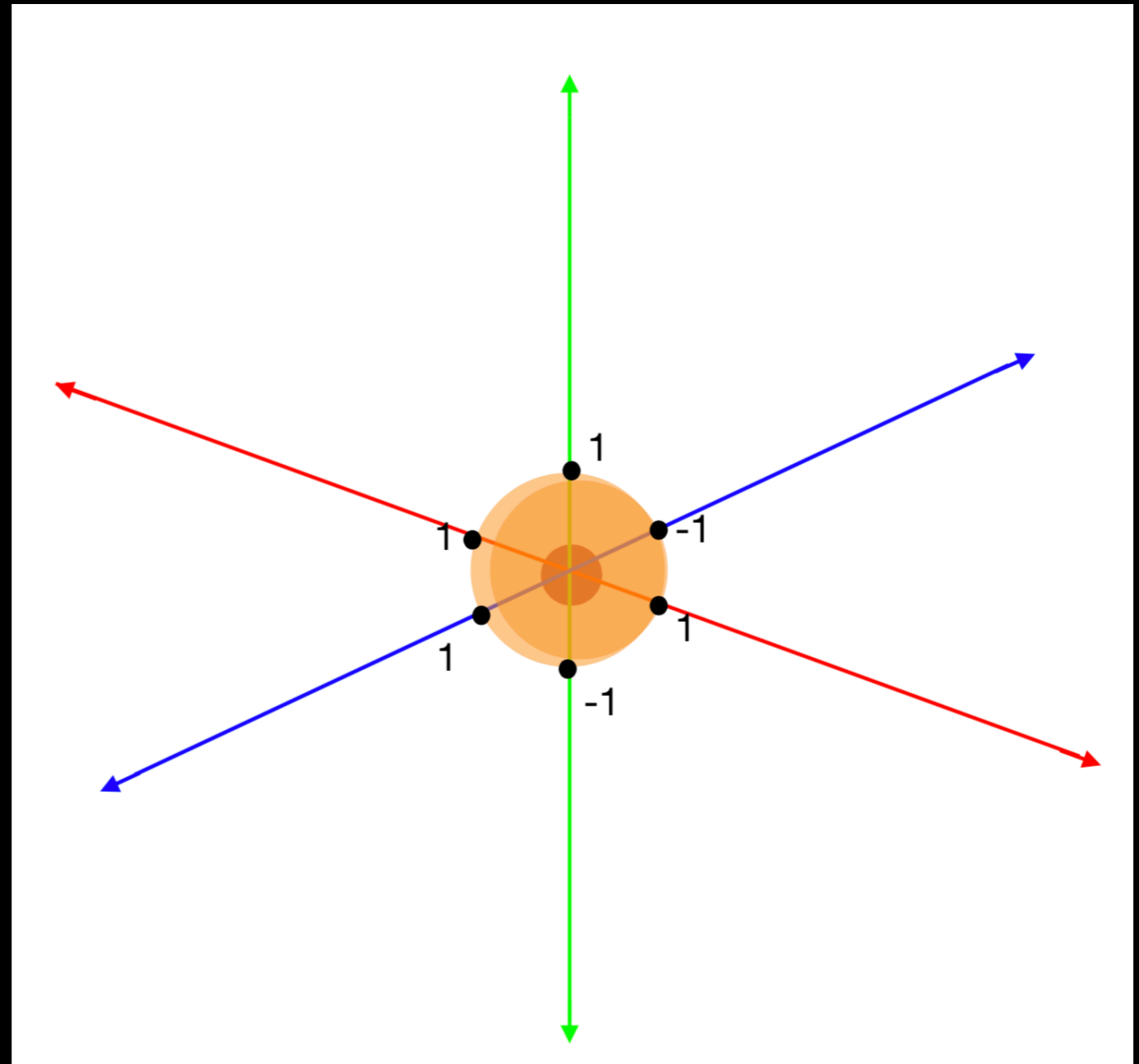
Sphere.Position = New SCNVector3MBS(0, 0, 0)

MyScene.rootNode.addChildNode (Sphere)

Position

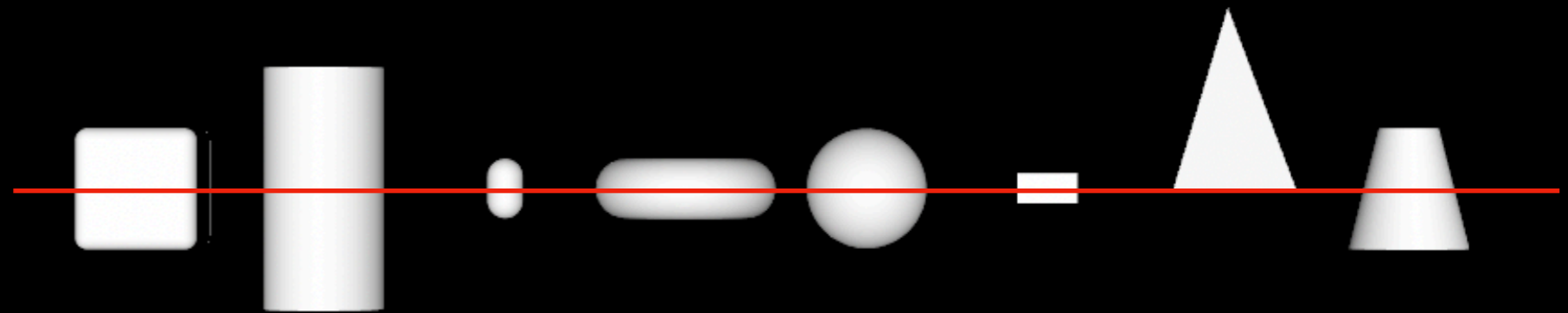
SceneKit uses three-component vectors for a variety of purposes, such as describing node or vertex positions, surface normals, and scale or translation transforms. The different vector components should be interpreted based on the context in which the vector is being used.

Constructor (x as double = 0.0, y as double = 0.0, z as double = 0.0)



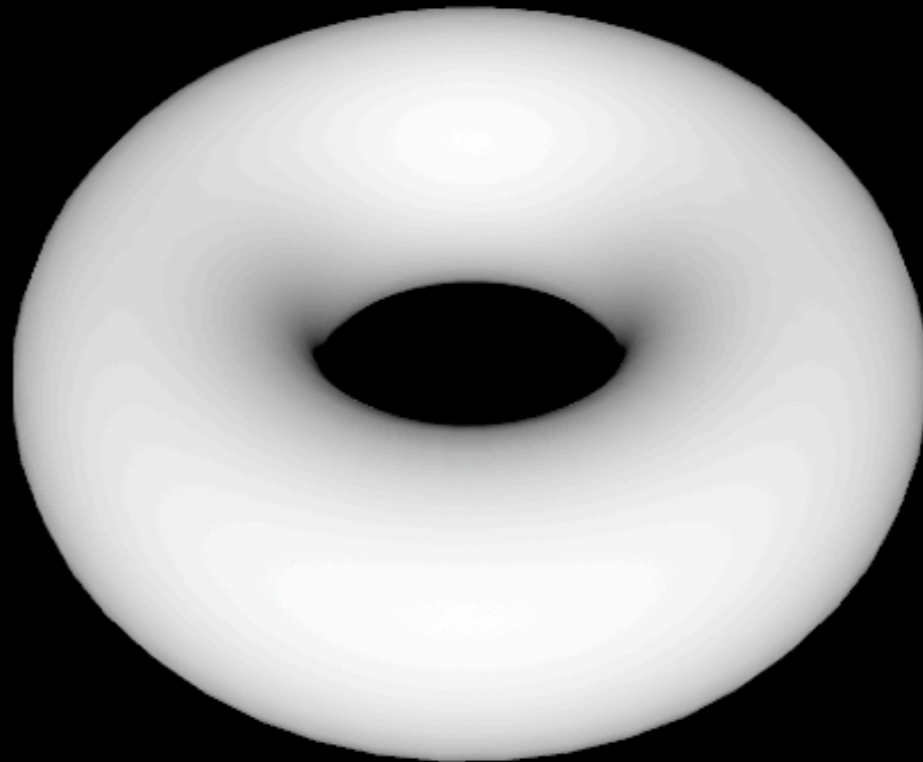
Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Basic forms

- Sphere
- **Torus**
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



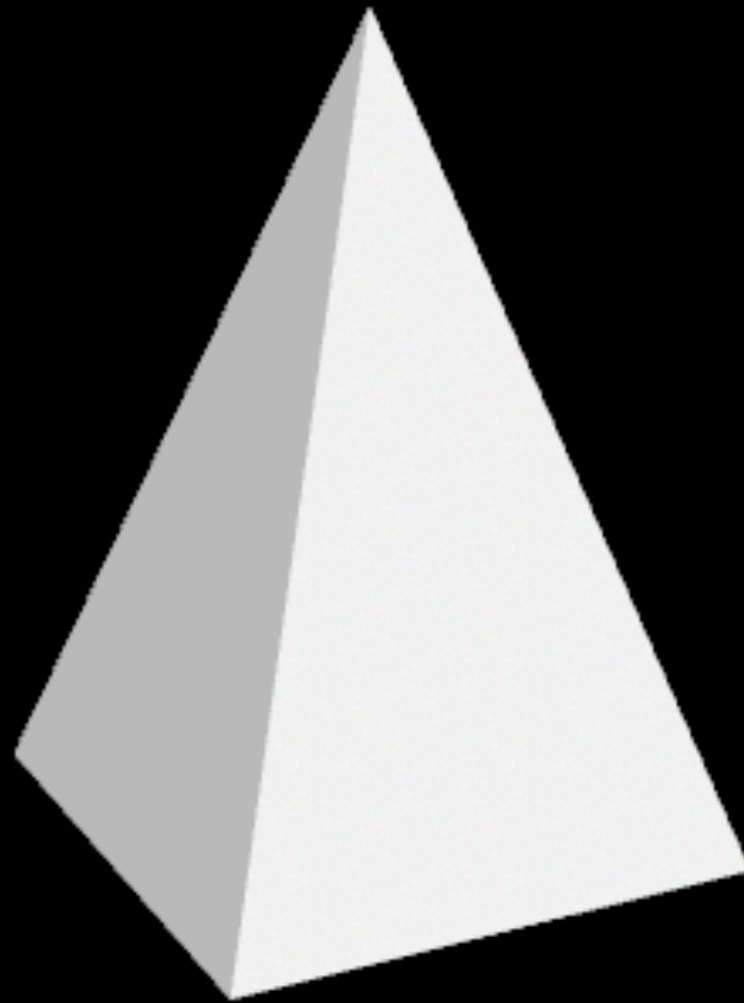
Basic forms

- Sphere
- Torus
- **Plane**
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



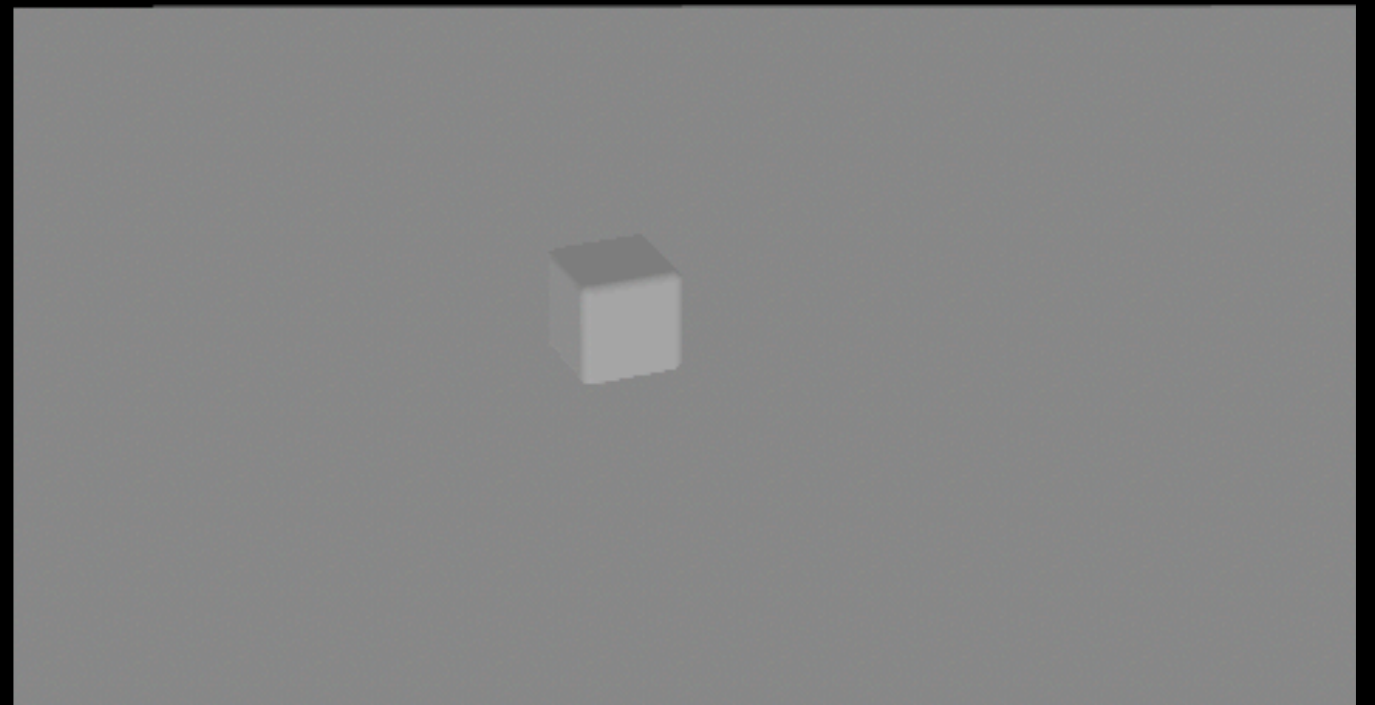
Basic forms

- Sphere
- Torus
- Plane
- **Pyramid**
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- **Cylinder**
- Cone
- Capsula
- Box
- Tube



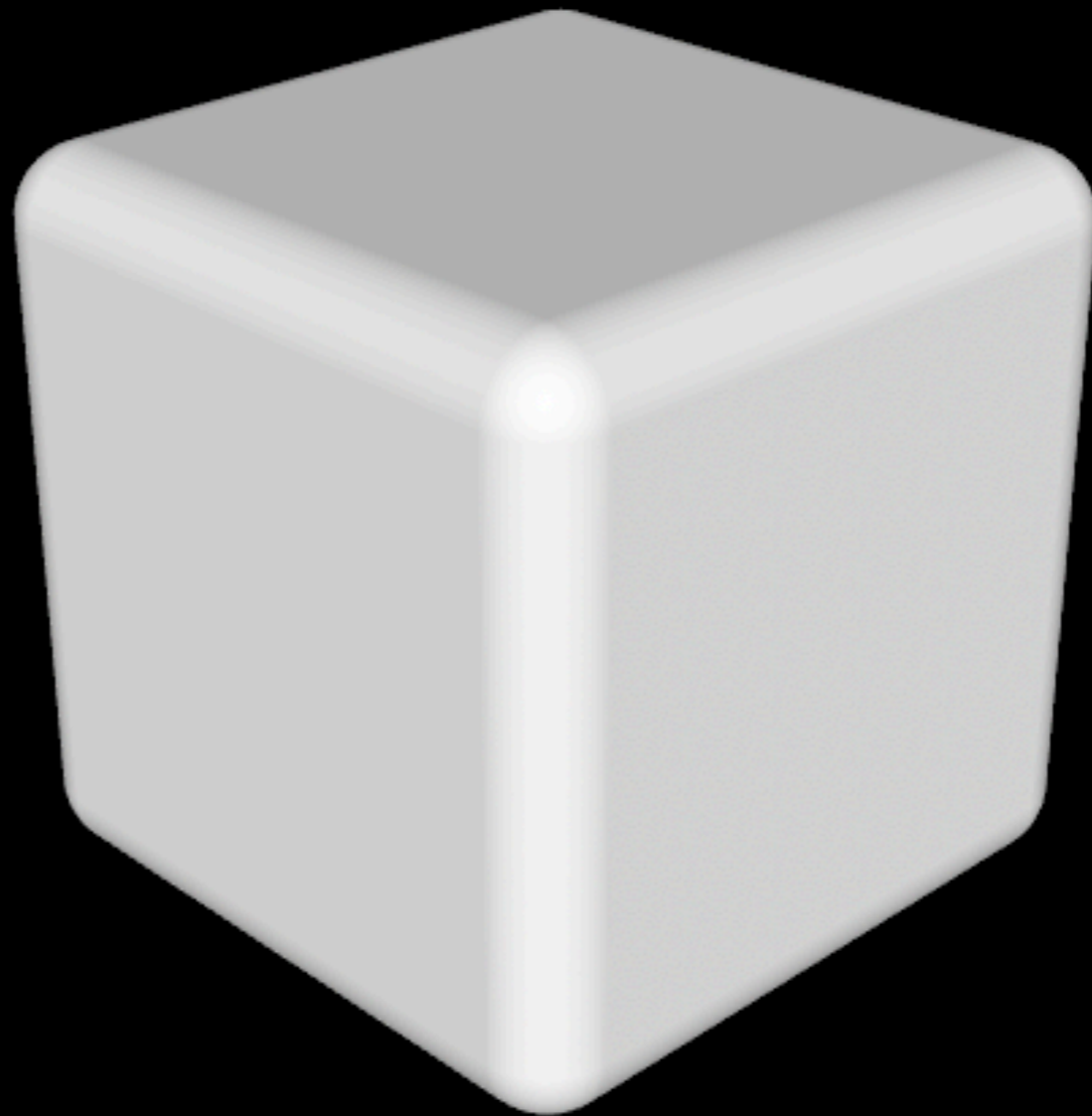
Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



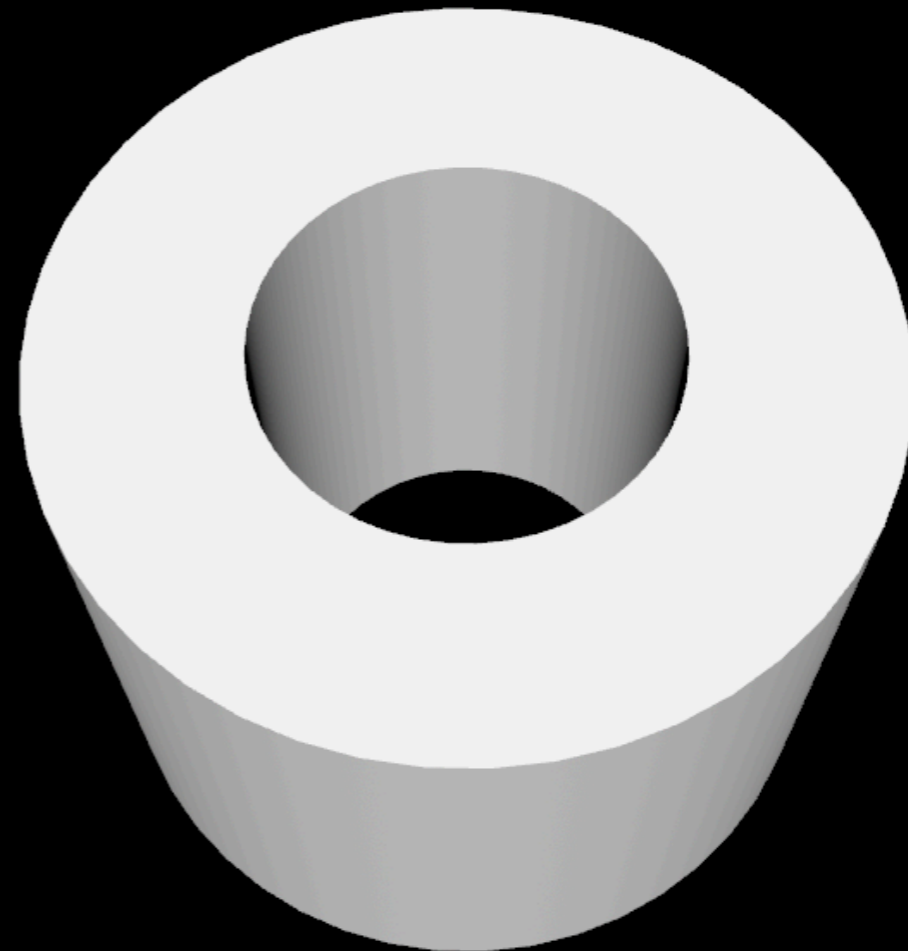
Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- **Box**
- Tube



Basic forms

- Sphere
- Torus
- Plane
- Pyramid
- Floor
- Cylinder
- Cone
- Capsula
- Box
- Tube



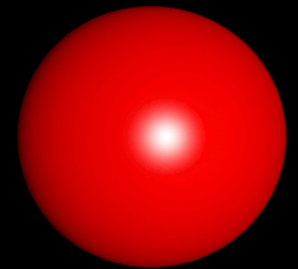
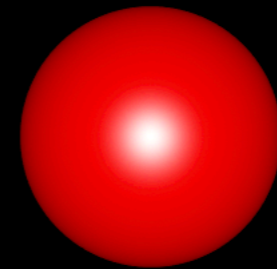
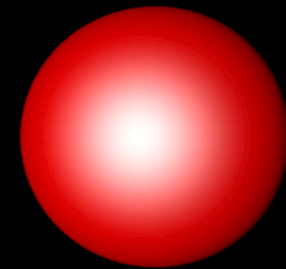
Text

```
Dim Textstring As String = "Xojo is Cool!"  
Dim Textextrusion As Double = 10  
  
Dim TextGeometry As New  
SCNTextMBS(Textstring, Textextrusion)  
TextGeometry.chamferRadius = 0.9  
TextGeometry.chamferSegmentCount = 10  
TextGeometry.flatness = 0.1  
Dim Texts As New SCNNodeMBS(TextGeometry)  
MyScene.rootNode.addChildNode(Texts)
```



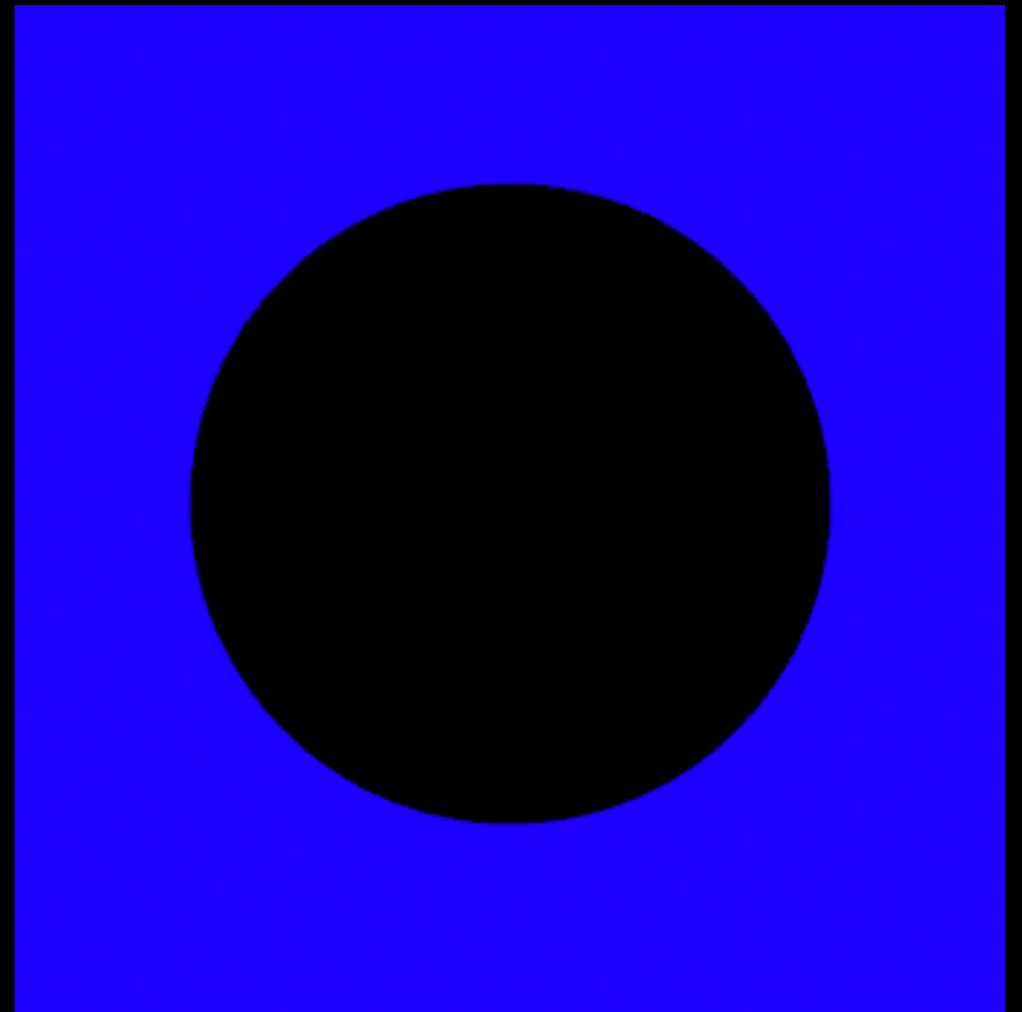
SceneKit

- Control, View and Scene
- Geometries
- **Material**
- Lighting System
- Moving



Color your life

```
NodeOneGeometry.firstMaterial  
.diffuse.contents =  
NSColorMBS.blackColor
```



Color your life

- **blueColor**
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

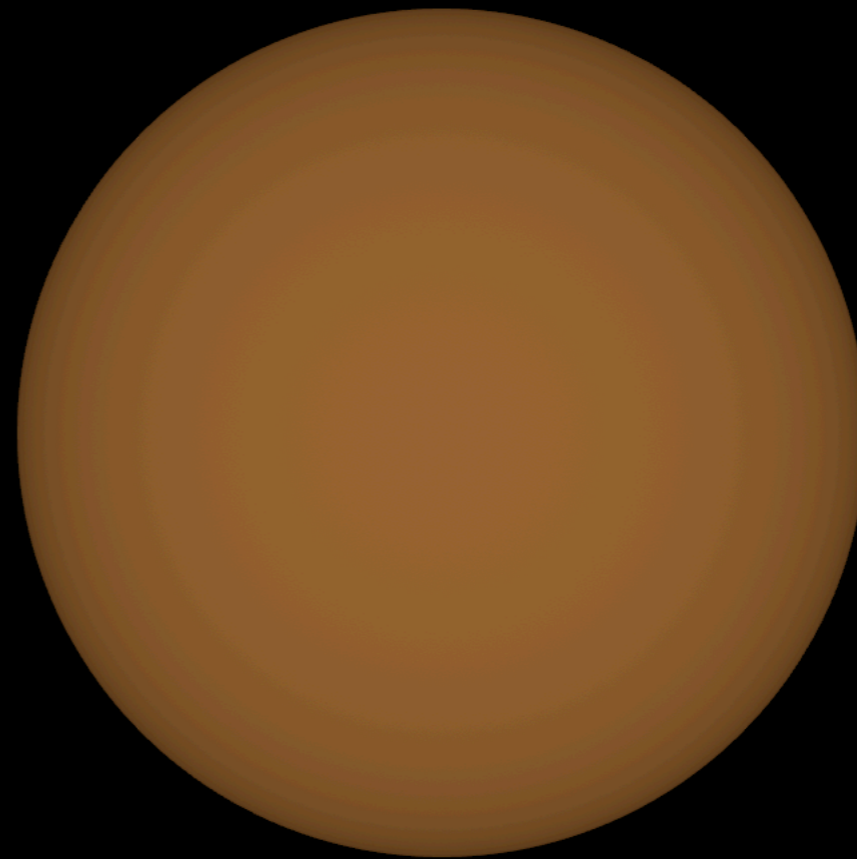
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- **brownColor**
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

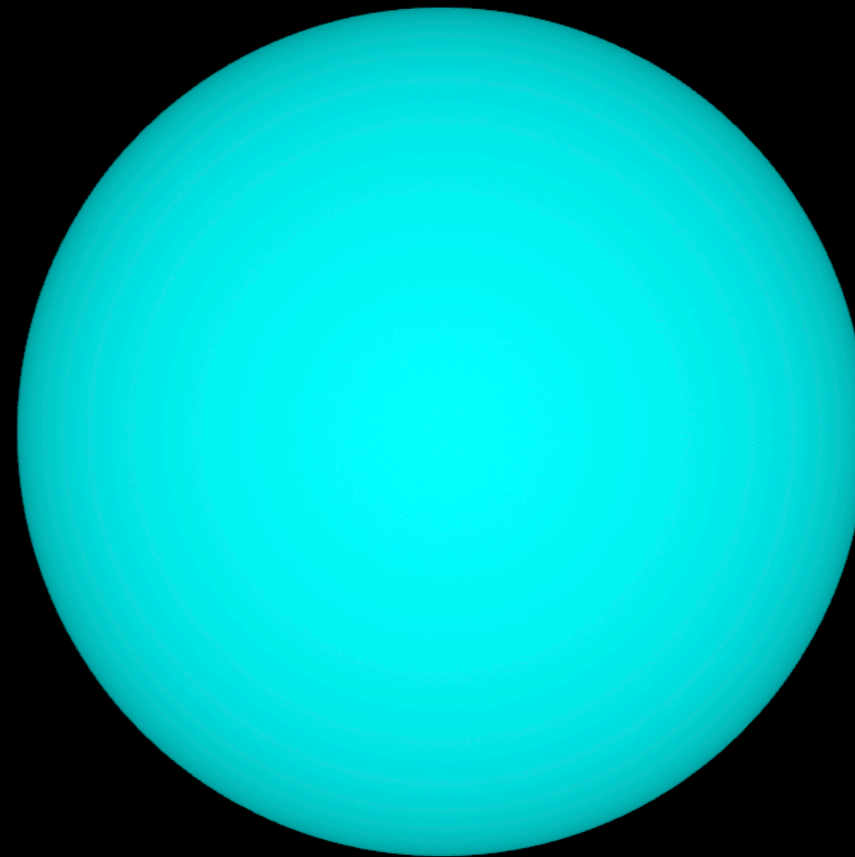
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- **cyanColor**
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

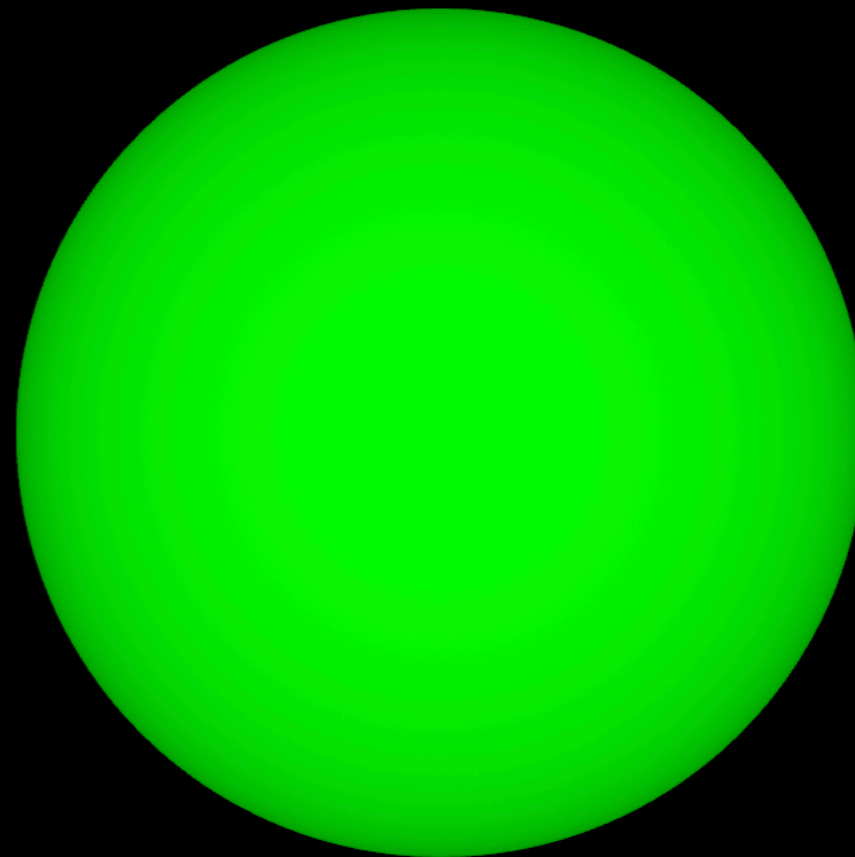
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- **greenColor**
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

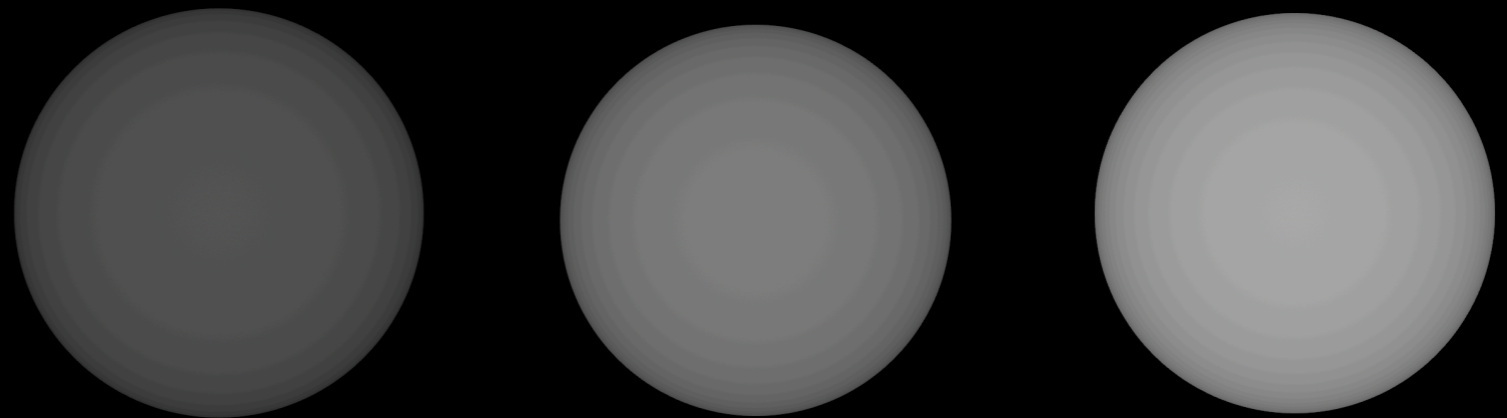
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- **darkGrayColor**
- **GrayColor**
- **lightGrayColor**
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

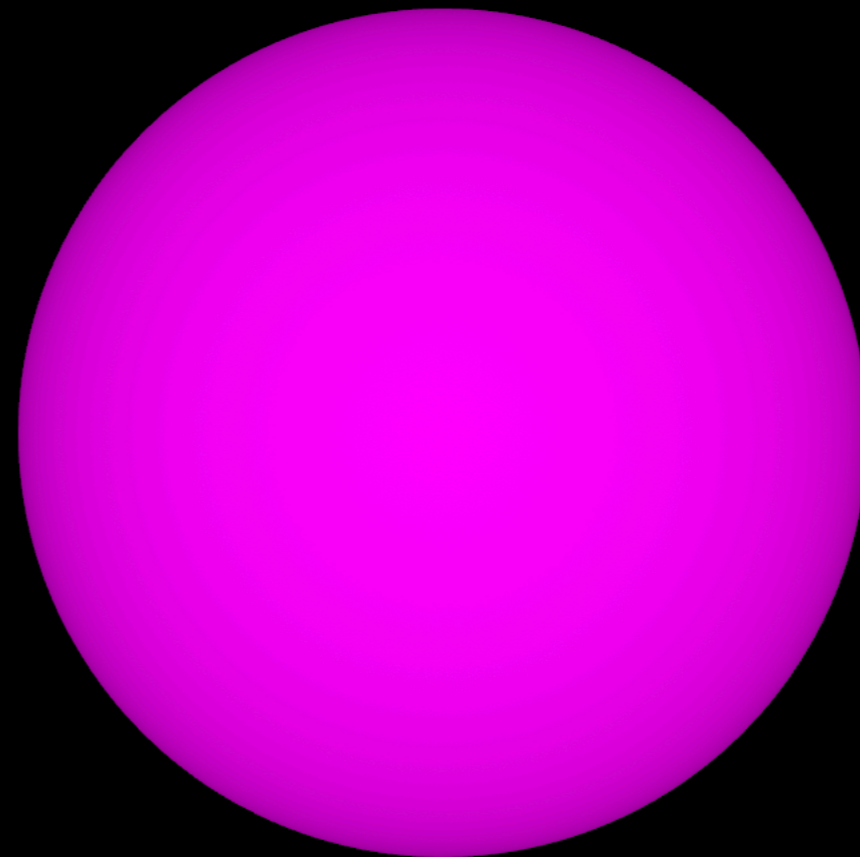
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- **magentaColor**
- orangeColor
- purpleColor
- redColor
- witheColor
- yellowColor

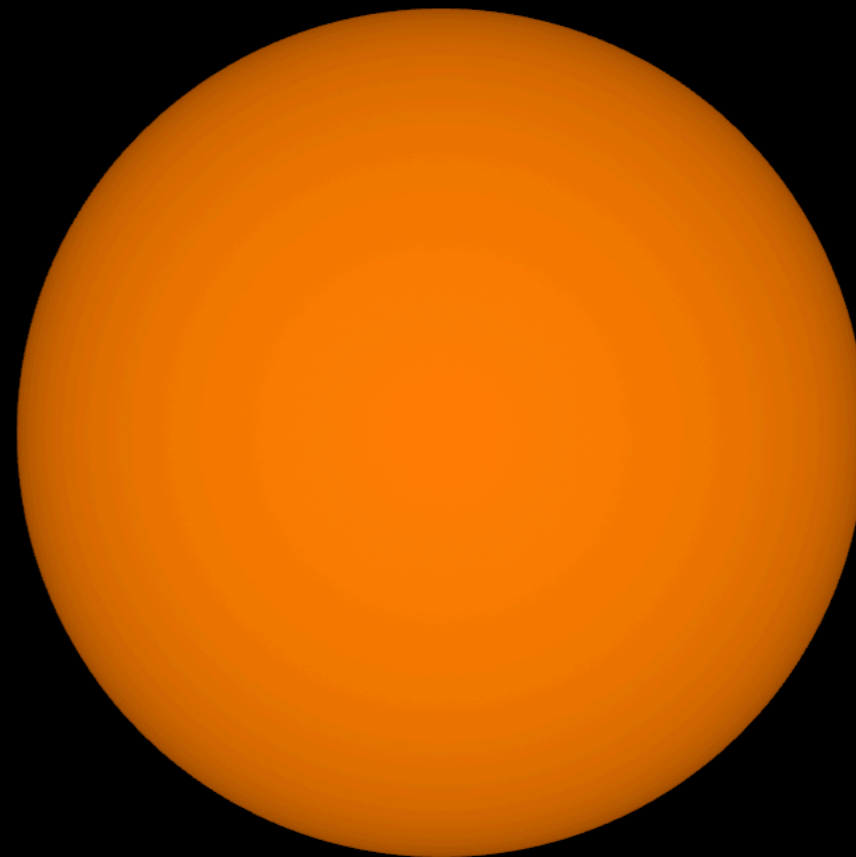
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- **orangeColor**
- purpleColor
- redColor
- witheColor
- yellowColor

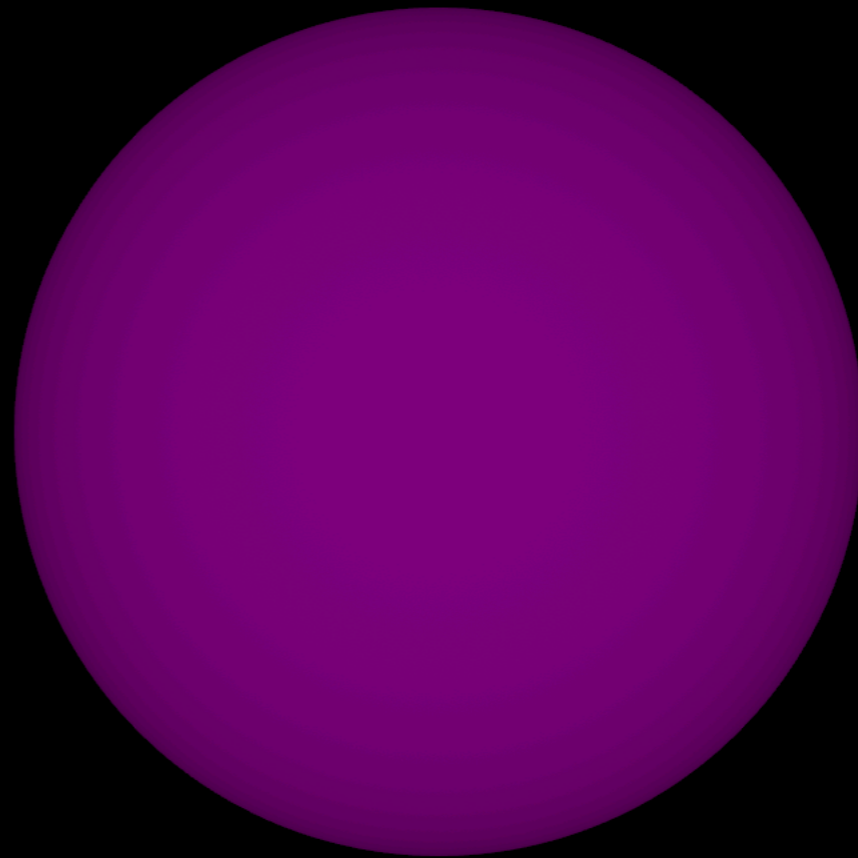
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- **purpleColor**
- redColor
- witheColor
- yellowColor

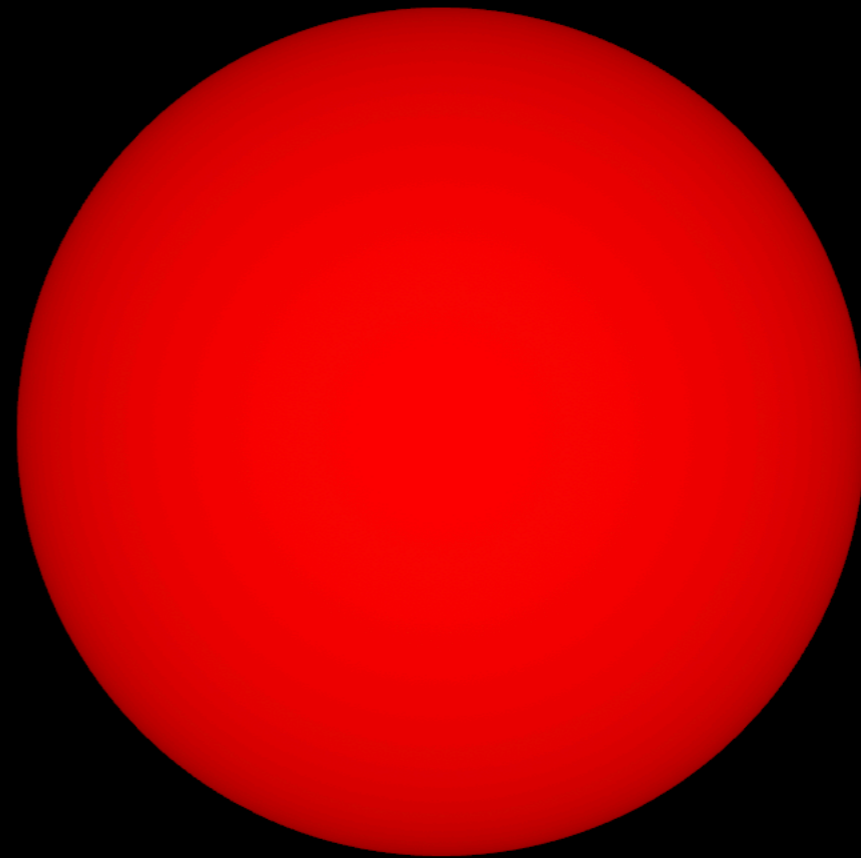
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- **redColor**
- witheColor
- yellowColor

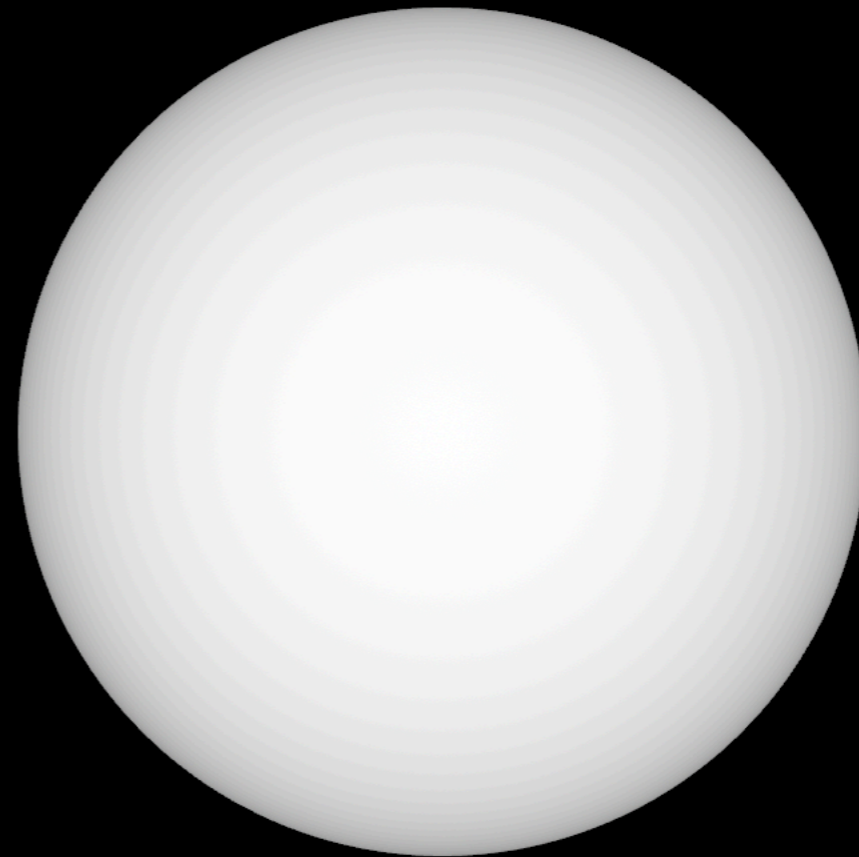
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- **witheColor**
- yellowColor

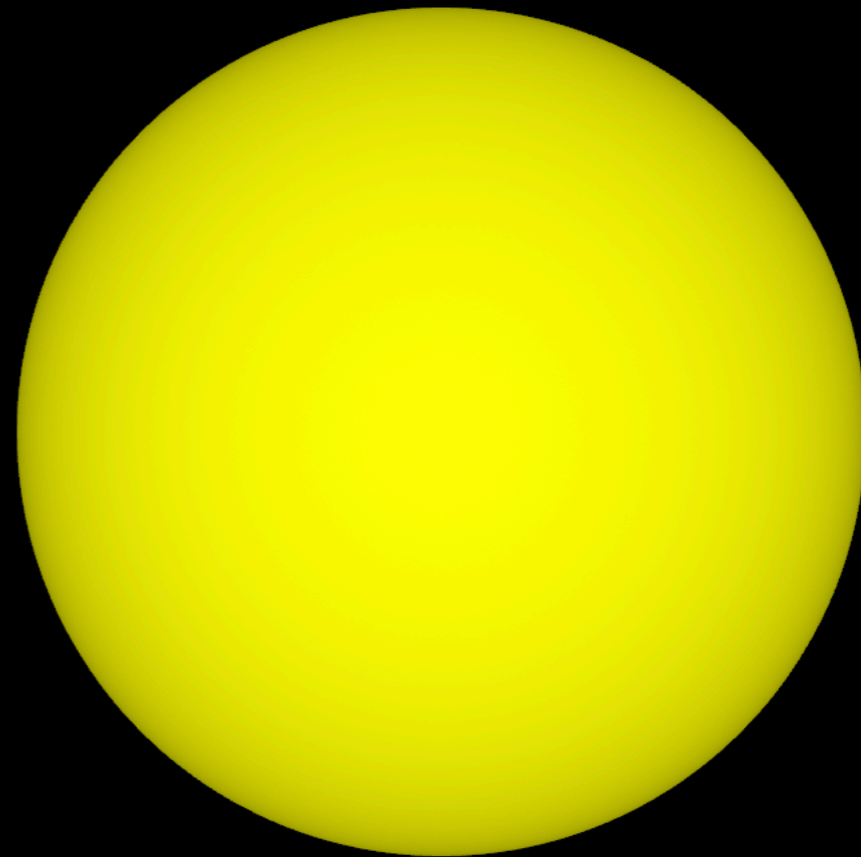
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your life

- blueColor
- brownColor
- cyanColor
- greenColor
- darkGrayColor
- GrayColor
- lightGrayColor
- magentaColor
- orangeColor
- purpleColor
- redColor
- witheColor
- **yellowColor**

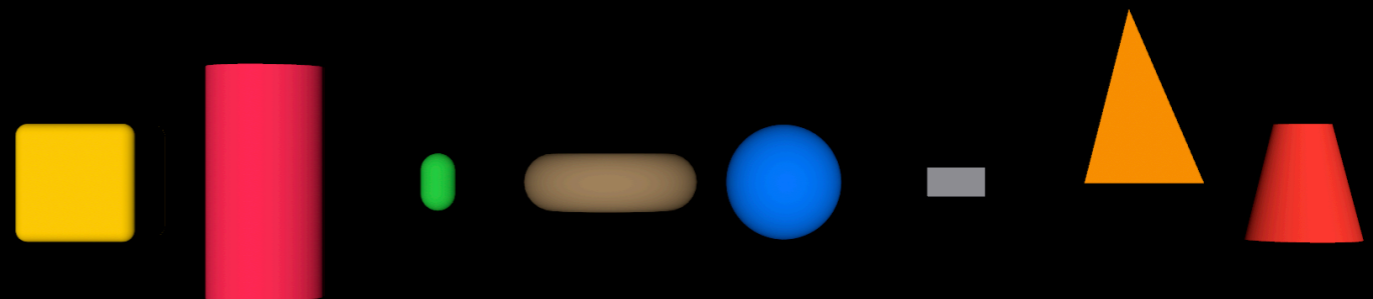
```
NodeOneGeometry.firstMaterial.  
diffuse.contents =  
NSColorMBS....
```



Color your live

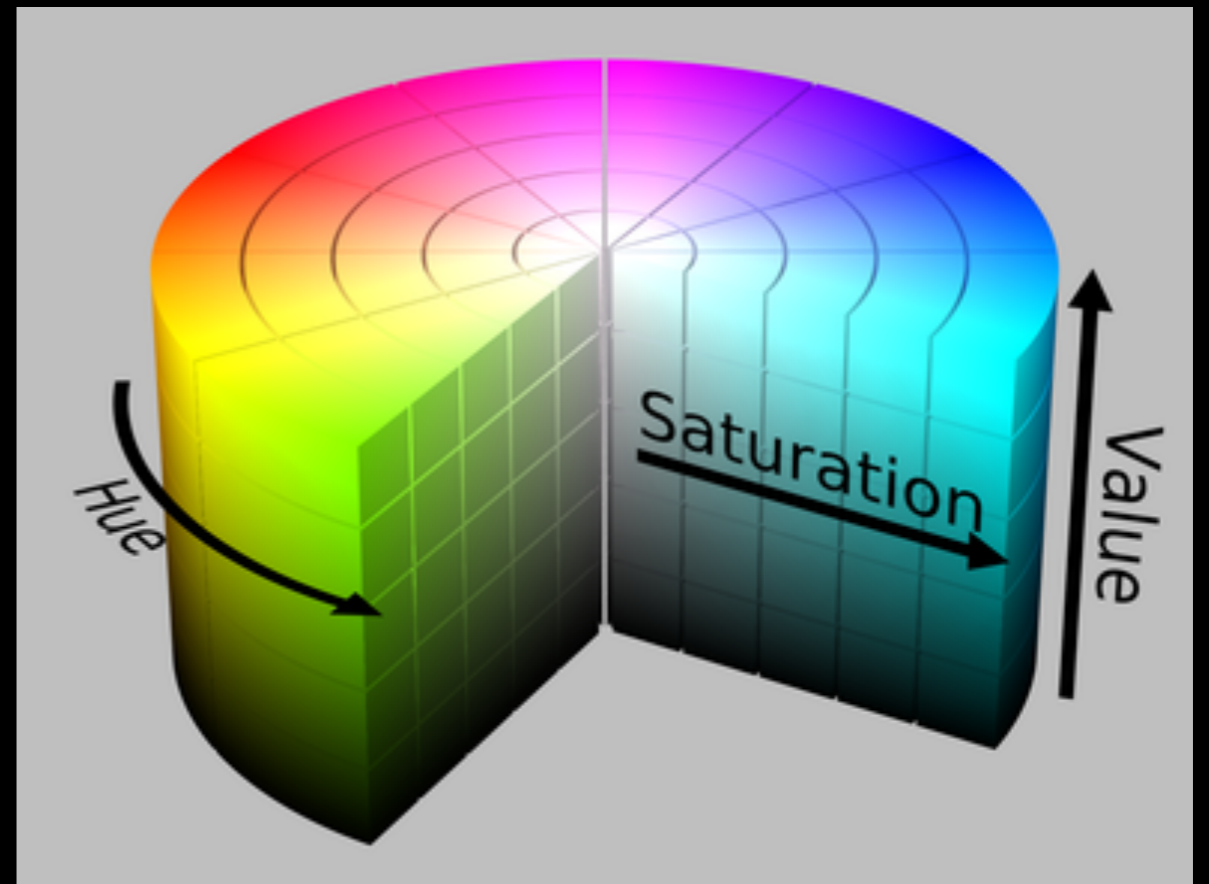
System Color

- `systemBlueColor`
- `systemBrownColor`
- `systemGrayColor`
- `systemGreenColor`
- `systemIndigoColor`
- `systemOrangeColor`
- `systemPinkColor`
- `systemPurpleColor`
- `systemRedColor`
- `systemTealColor`
- `systemYellowColor`



HSV Colorspace

`NSColorMBS.colorWithHSV(`
hue as double,
saturation as double,
brightness as double,
alpha as double=1.0)
as `NSColorMBS`



RGB Colorspace

`NSColorMBS.colorWithSRGB(`

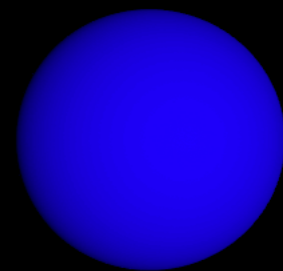
`red as Double,`

`green as Double,`

`blue as Double,`

`alpha as Double=1.0)`

`as NSColorMBS`

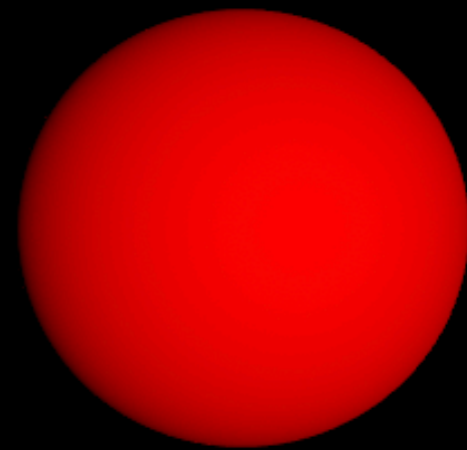
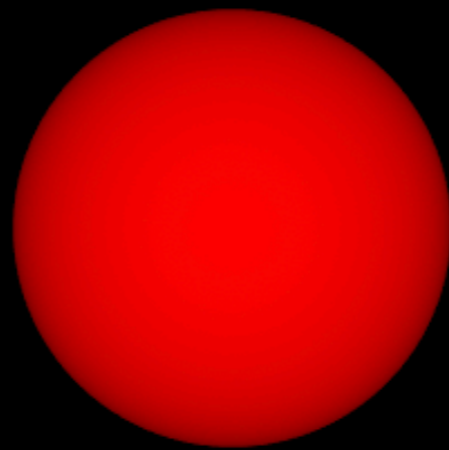


Map a photo on your geometry

```
PlaneGeometry.firstMaterial.  
diffuse.contents = pic
```



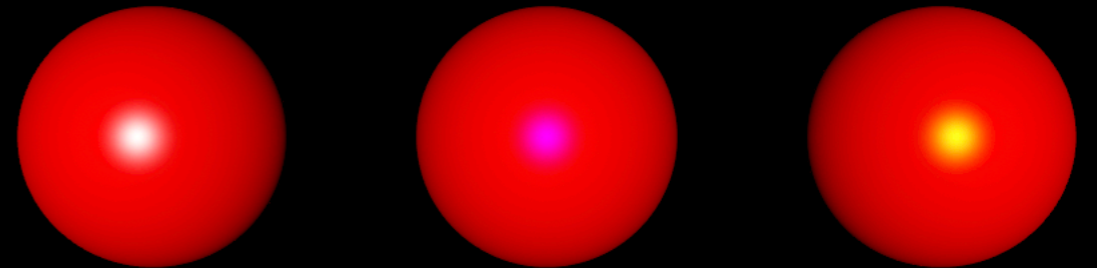
Properties of a material



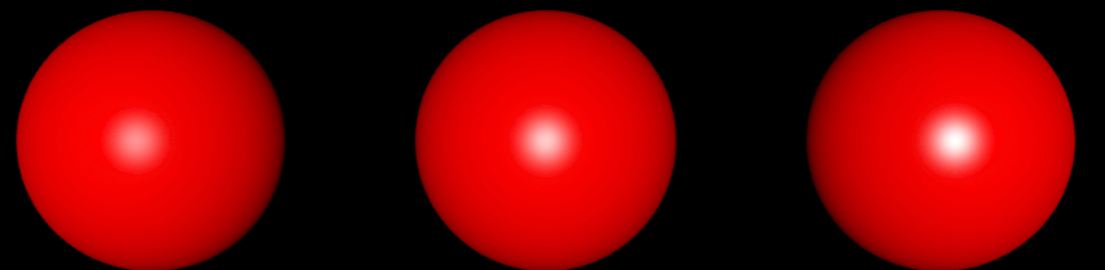
Properties of a material

specular

SphereGeometry.
firstMaterial.specular.
contents=
NSColorMBS.whiteColor



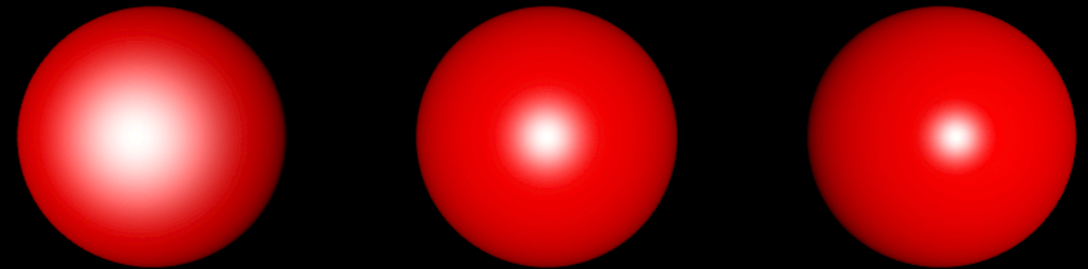
SphereGeometry.
firstMaterial.specular.
intensity = 0.9



Properties of a material

shininess

SphereGeometry.
firstMaterial.shininess
= 0.5



Properties of a material

reflectivity

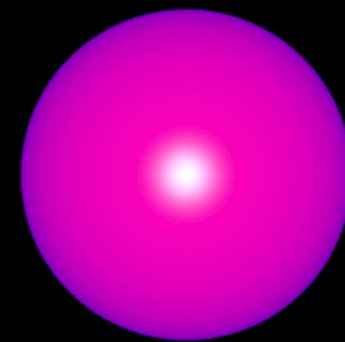
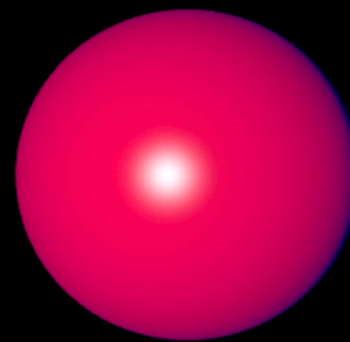
SphereGeometry.

firstMaterial.

reflective.contents =

NSColorMBS.

blueColor



SphereGeometry.

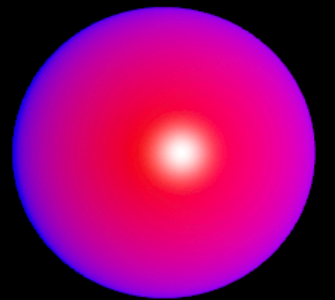
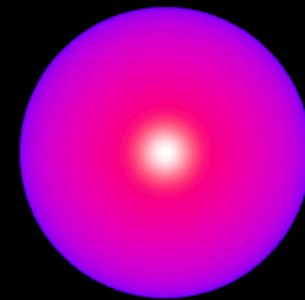
firstMaterial.

reflective.intensity = 1

Properties of a material

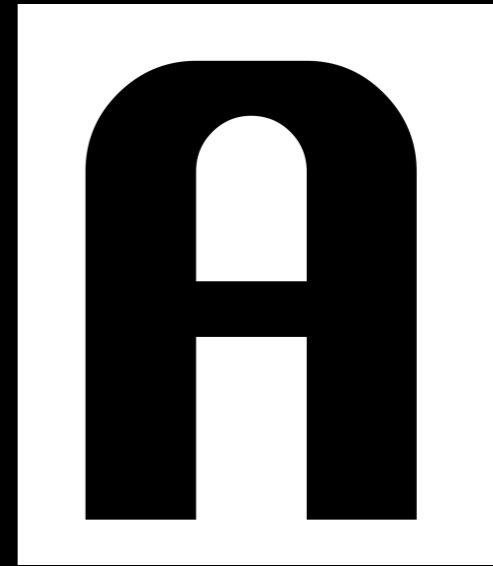
fresnel

SphereGeometry.
firstMaterial.
fresnelExponent = 1



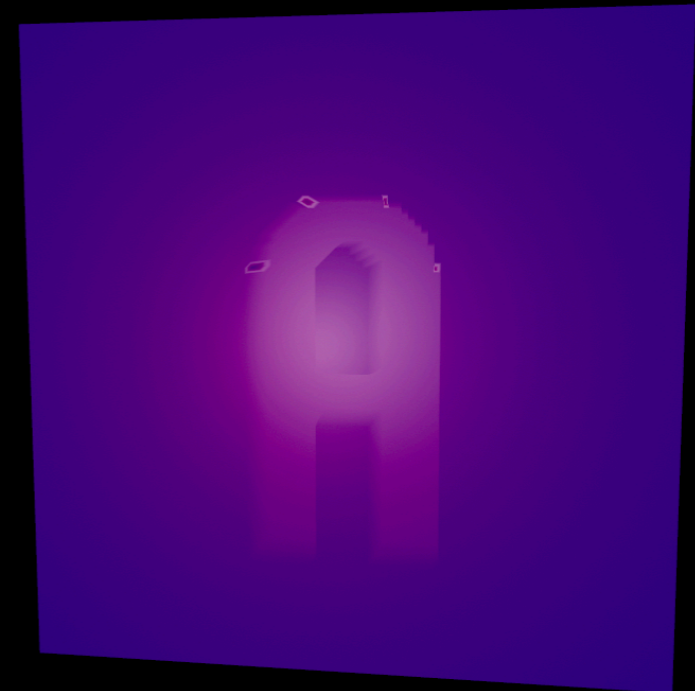
Properties of a material

displacement



SphereGeometryR.firstMaterial
.displacement.contents = A

SphereGeometryR.firstMaterial
.displacement.intensity = 2



Turn on the
light



~~myView.auto.enableDefaultLighting = True~~

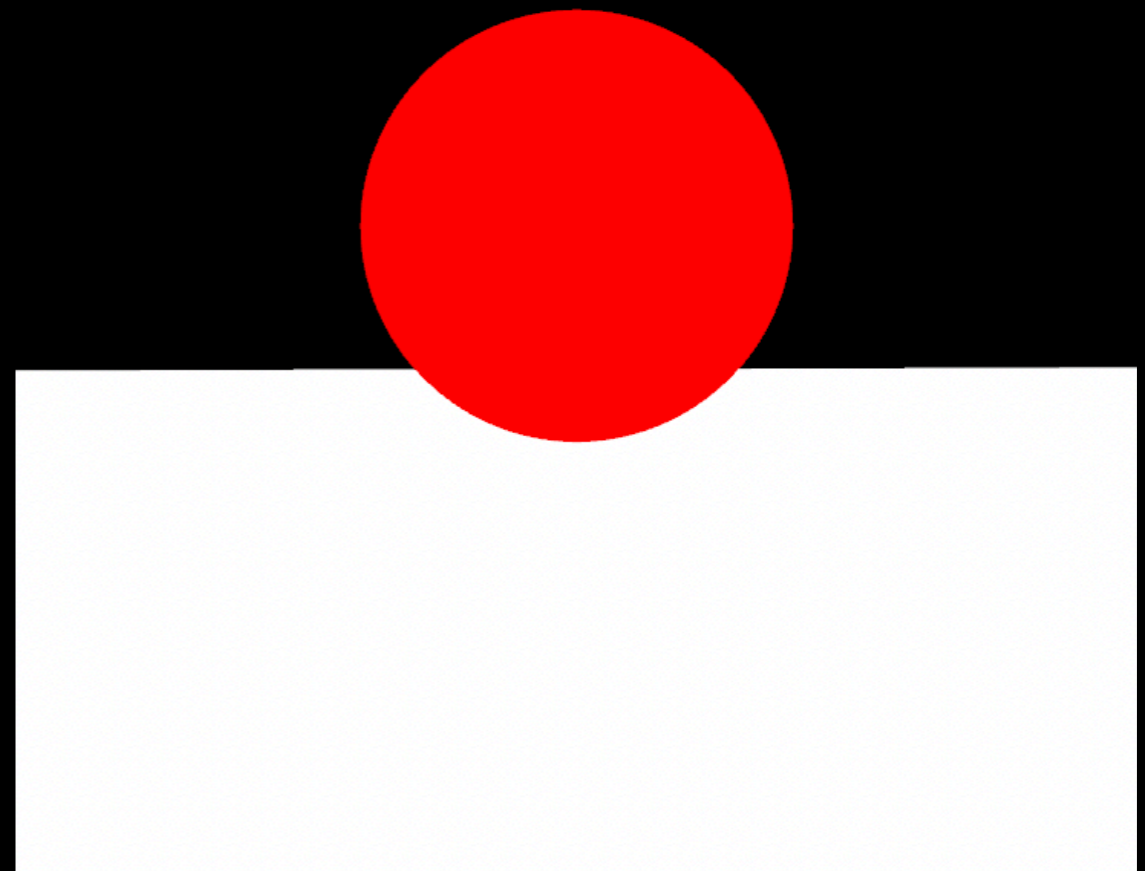
Lighting system Type

Ambient
Directional
IES
Omni
Spot



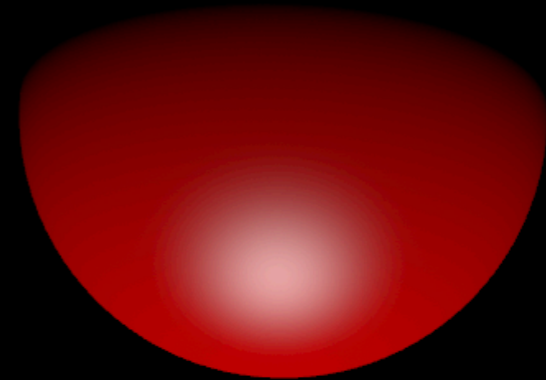
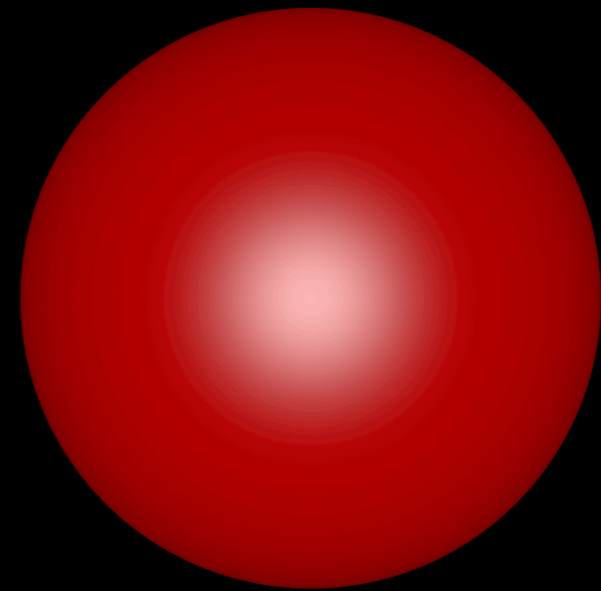
Ambient light

Effects all objects light up in the same intensity,
non directional lighting



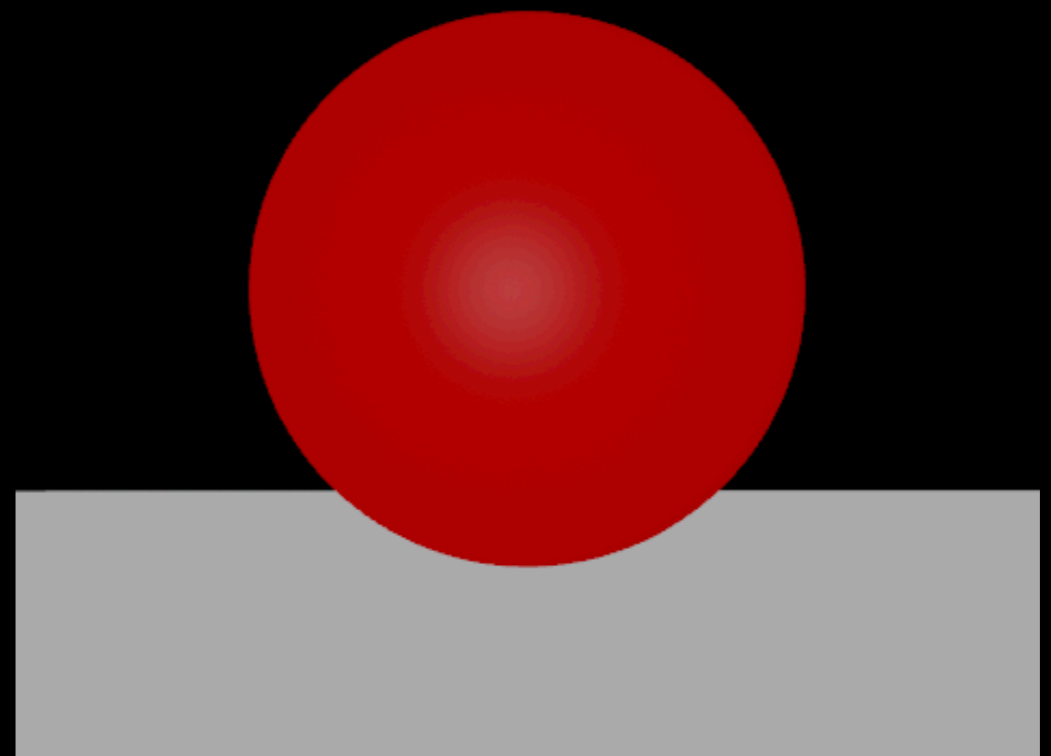
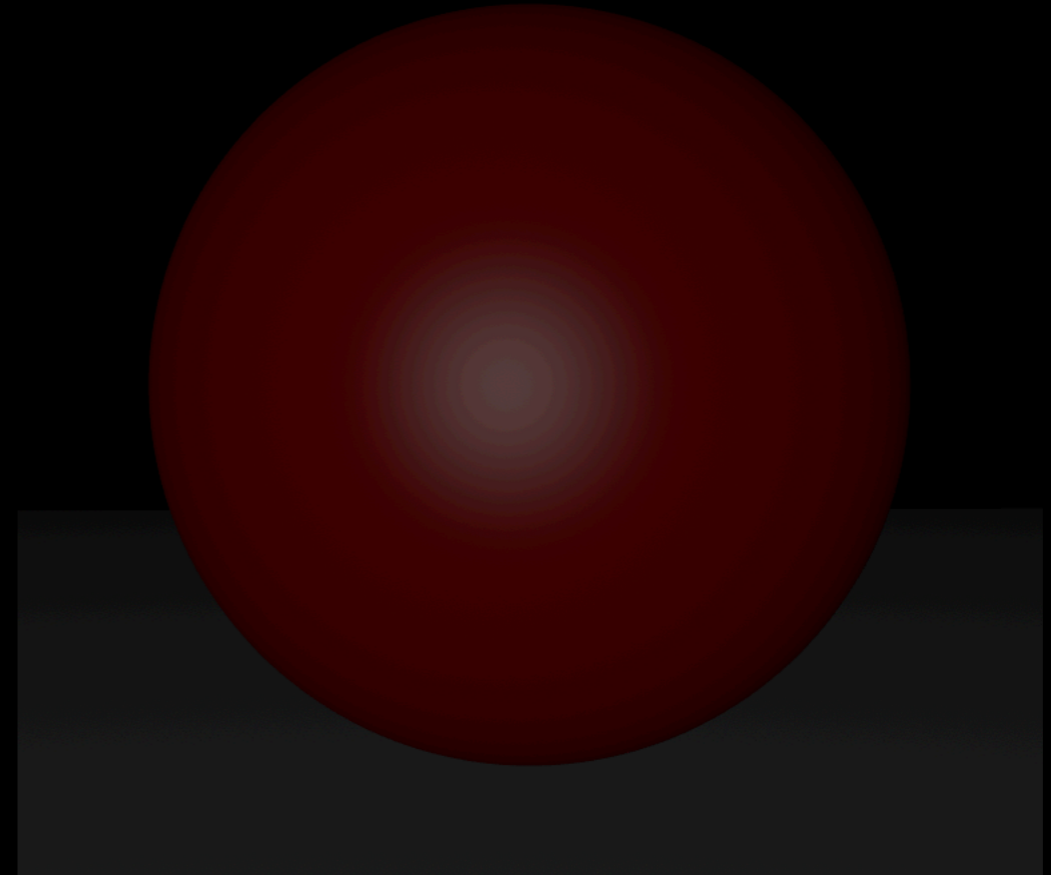
Directional light

Illuminates all Objects
from a given direction



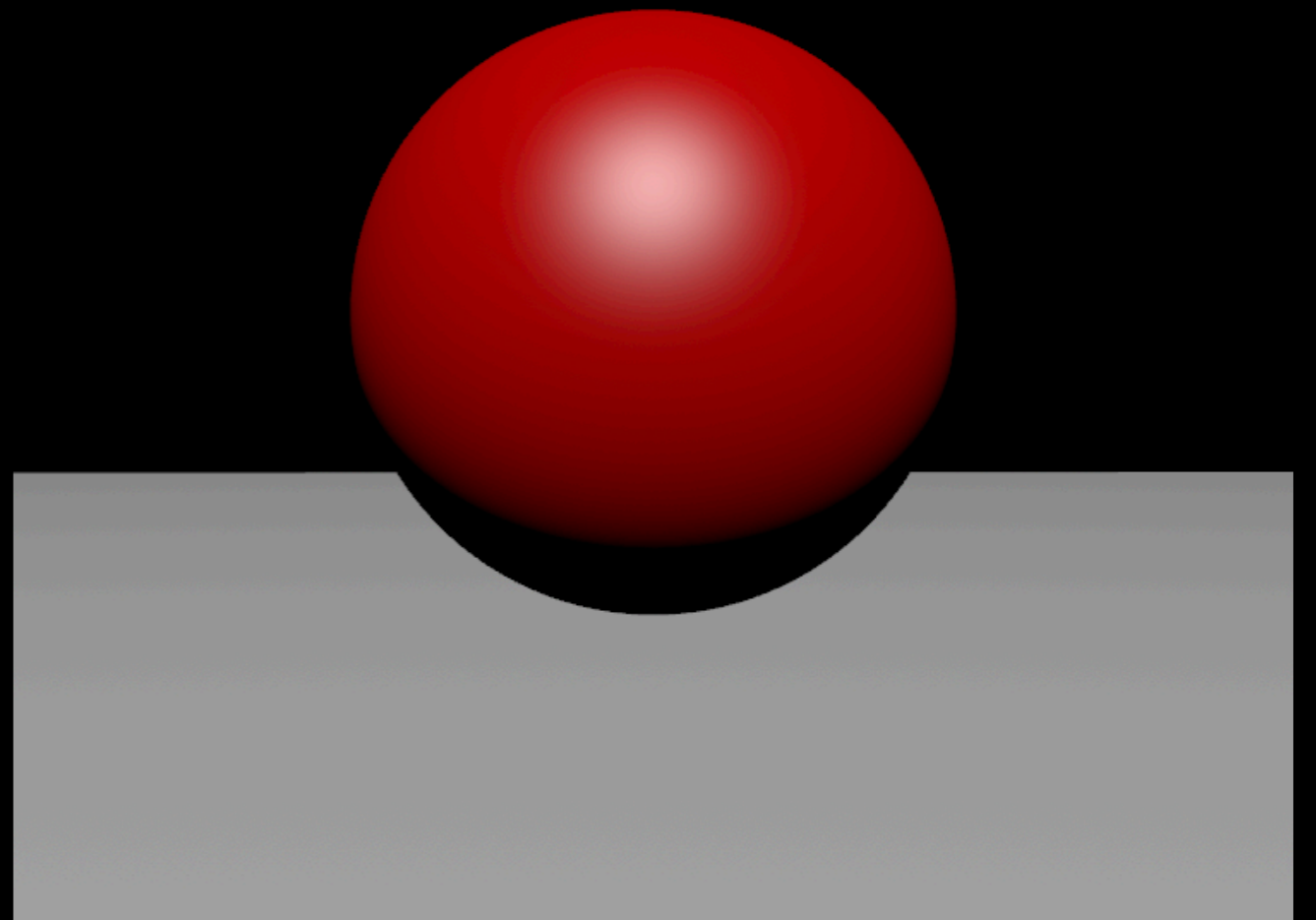
IES light

The cone of light is pear-shaped



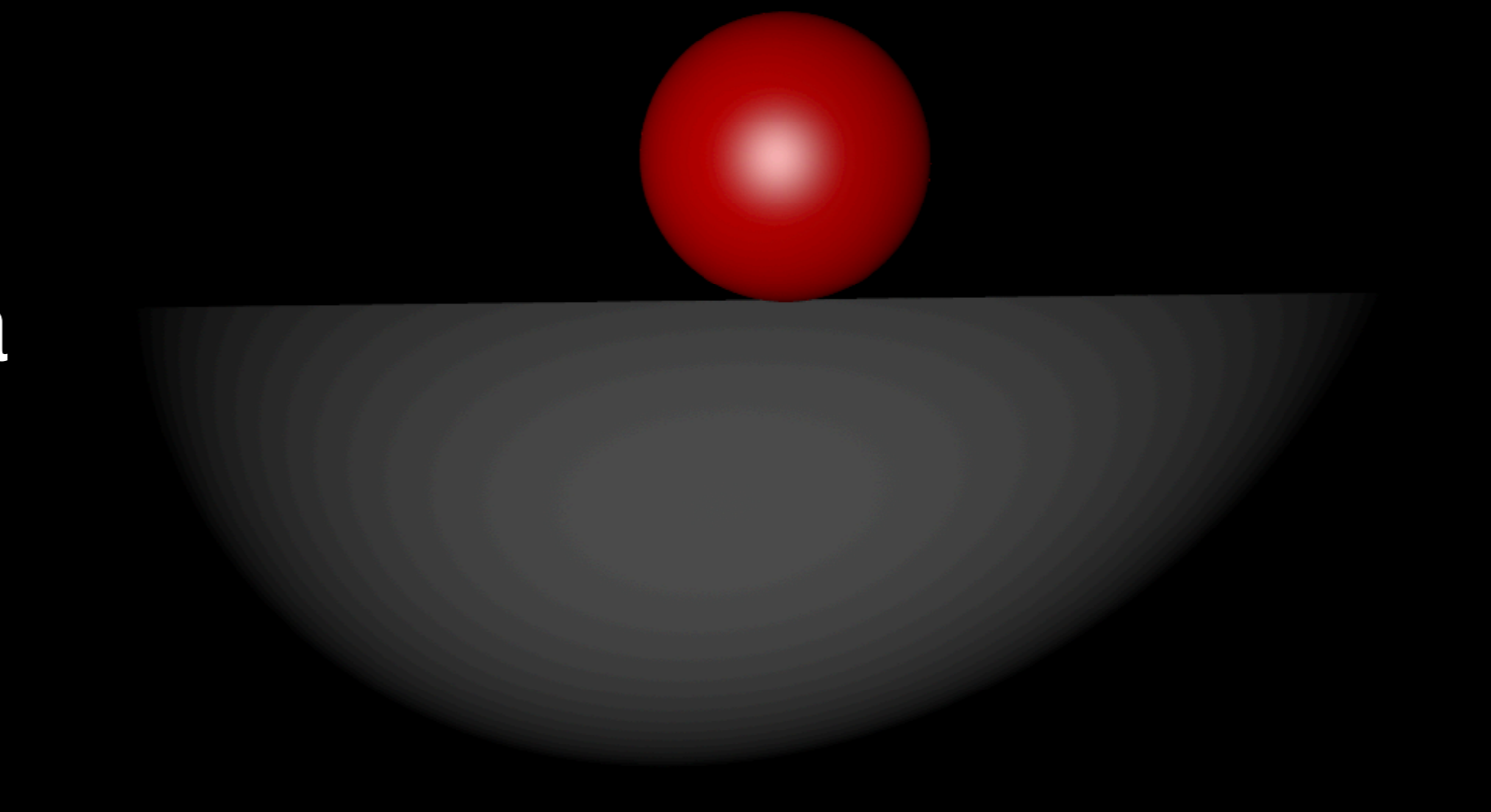
Omni light

Comes from a Point and
shines in all directions

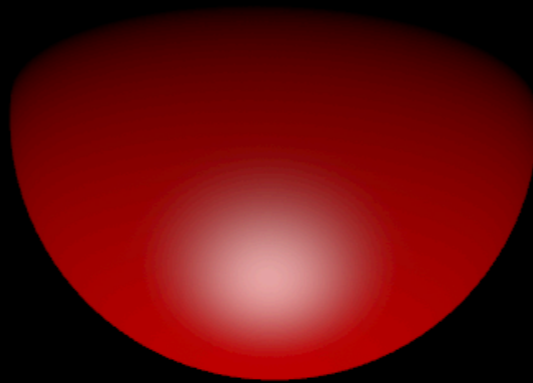


Spot light

Like a spot, light spreads outward in a cone from a defined point

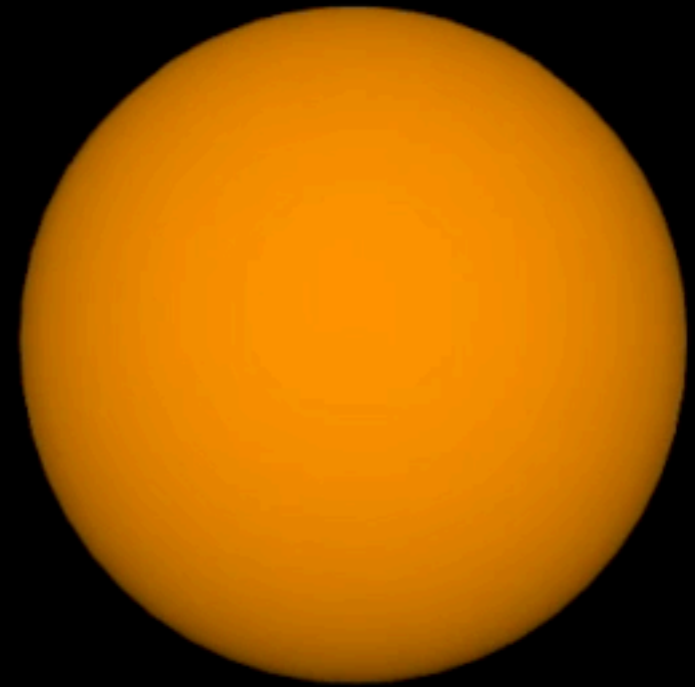


Implement the light



```
Dim directional As New SCNNodeMBS
directional.light = New SCNLightMBS
directional.Light.Type =
SCNLightMBS.SCNLightTypeDirectional
directional.light.Color =
NSColorMBS.colorWithWhite(0.75, 1)
MyScene.rootNode.addChildNode(directional)
```

Move it!

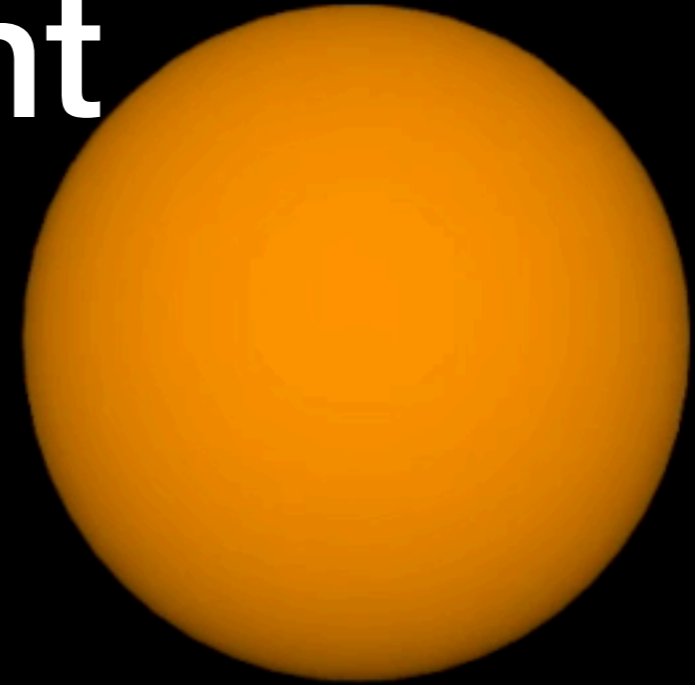


Two types of movements



- From point to an other point
- Rotation

Move from a point to an other point



- `SCNActionMBS.moveby`
(x, y, z, duration)
- `SCNActionMBS.moveto`
(SCNVector3MBS, duration)

Rotation

- Rotation around its own axis
- Rotation around an other object



Rotation around its own axis

SCNActionMBS.rotateBy
(x angle, y angle,
z angle, duration)



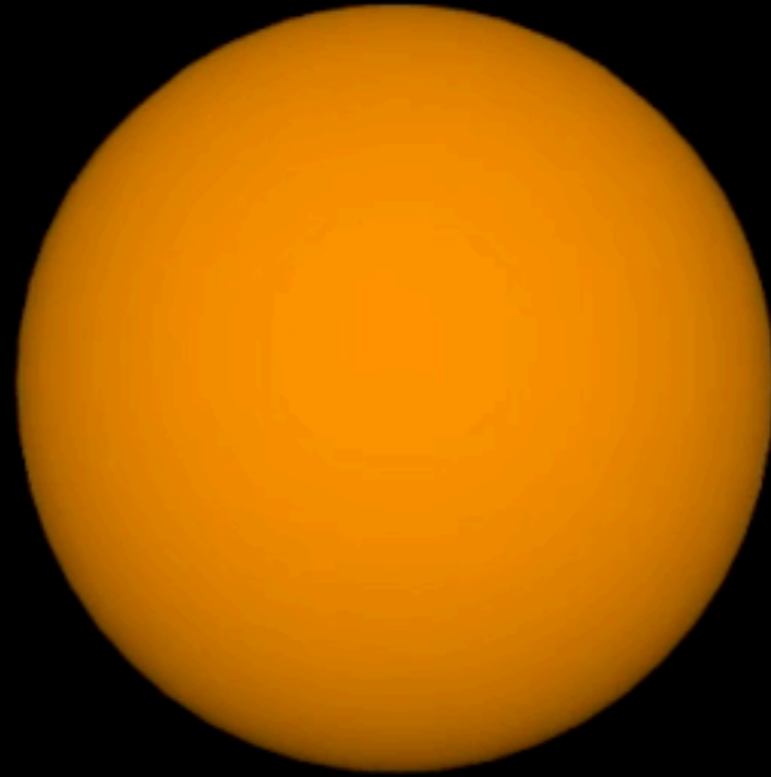
Rotation around an other object



The sphere is a parent
of the pyramid

```
sphere.addChildNode(pyramid)
```

How you implement moving



How you implement moving

```
Dim moveUp As SCNActionMBS = SCNActionMBS.moveby(0,  
1, 0.0, 0.5)
```

```
Dim moveDown As SCNActionMBS =  
SCNActionMBS.moveby(0, -1, 0, 0.5)
```

```
Dim sequence As SCNActionMBS =  
SCNActionMBS.sequence(Array (moveUp, moveDown))
```

```
Dim repeat As SCNActionMBS =  
SCNActionMBS.repeatActionForever(sequence)
```

```
Sphere.runAction(repeat)
```

Demo

Q&A

**Thank you for your
attention**



