GPU Programming for Video Games

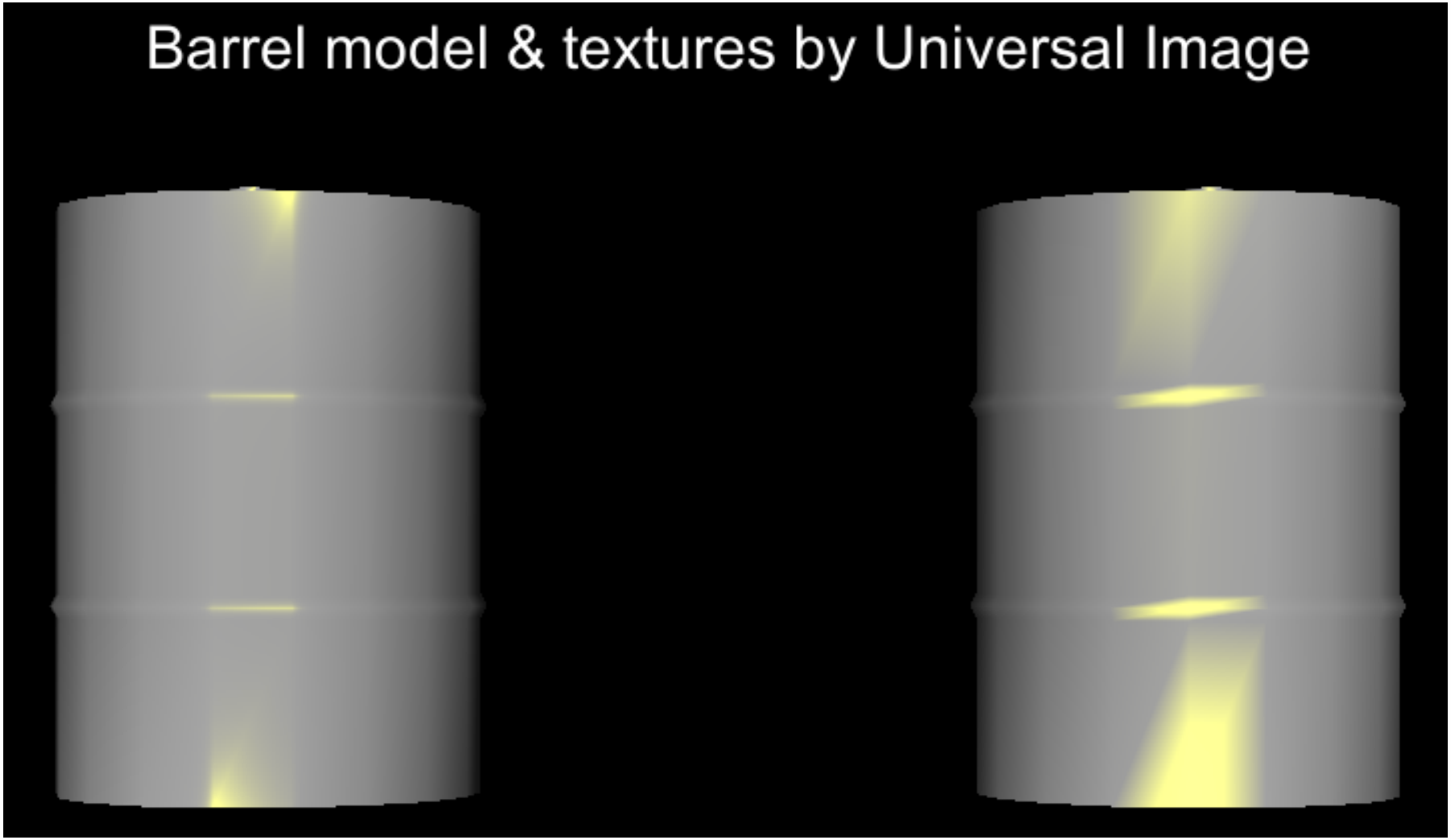
Materials in Unity

Prof. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

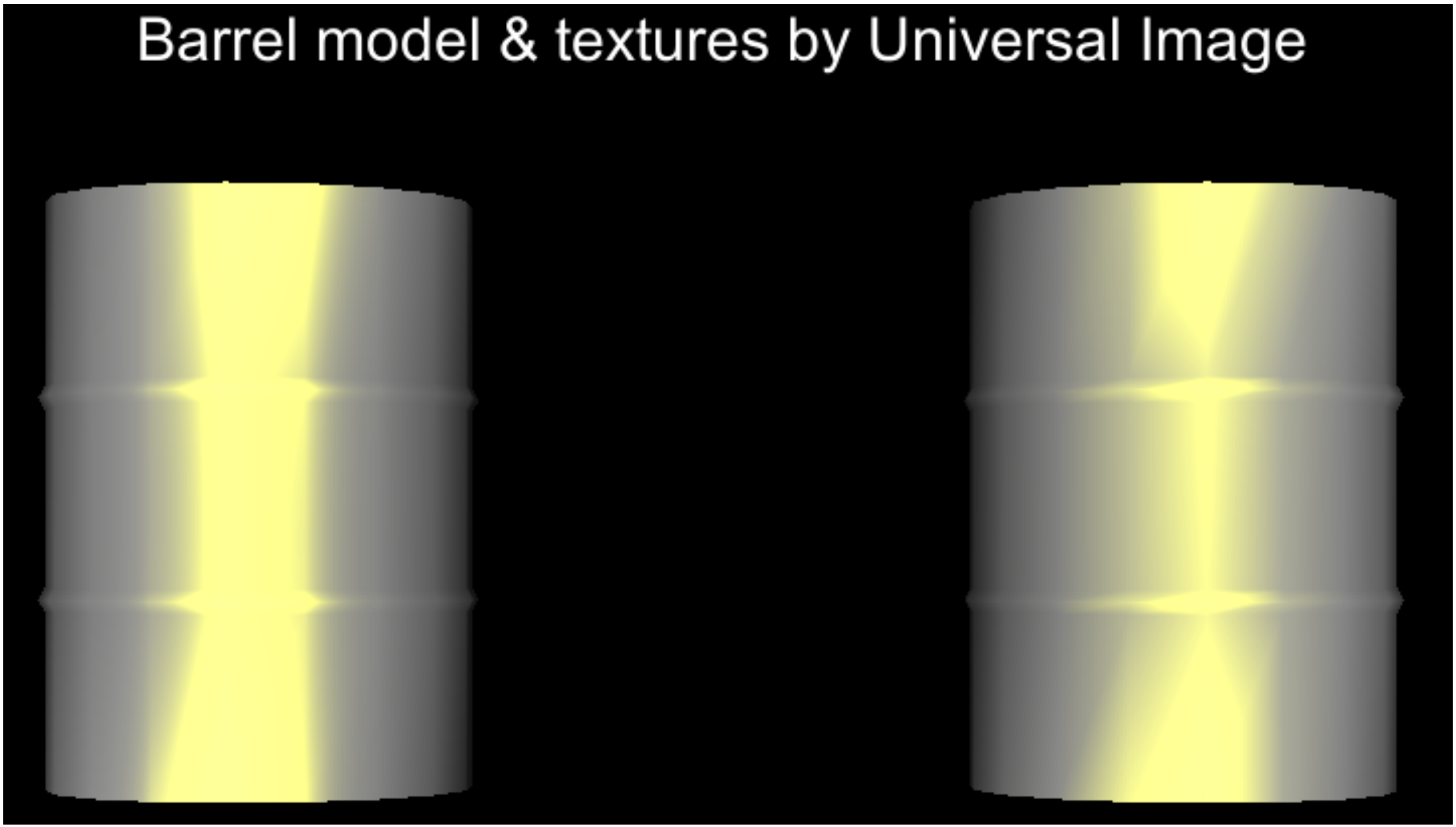
Georgia Institute of Technology

Per-vertex vs. per-pixel lighting (1)

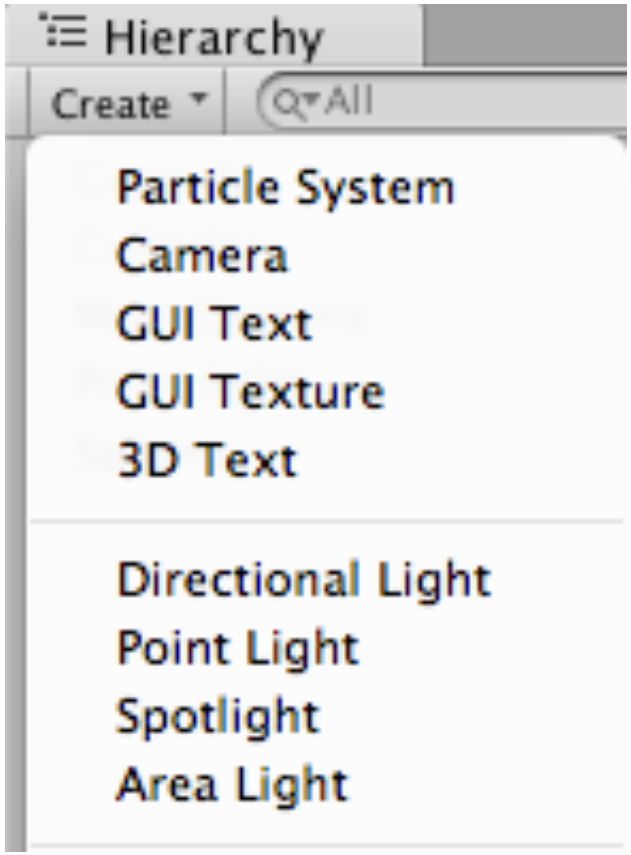
Per-vertex vs. per-pixel lighting (2)

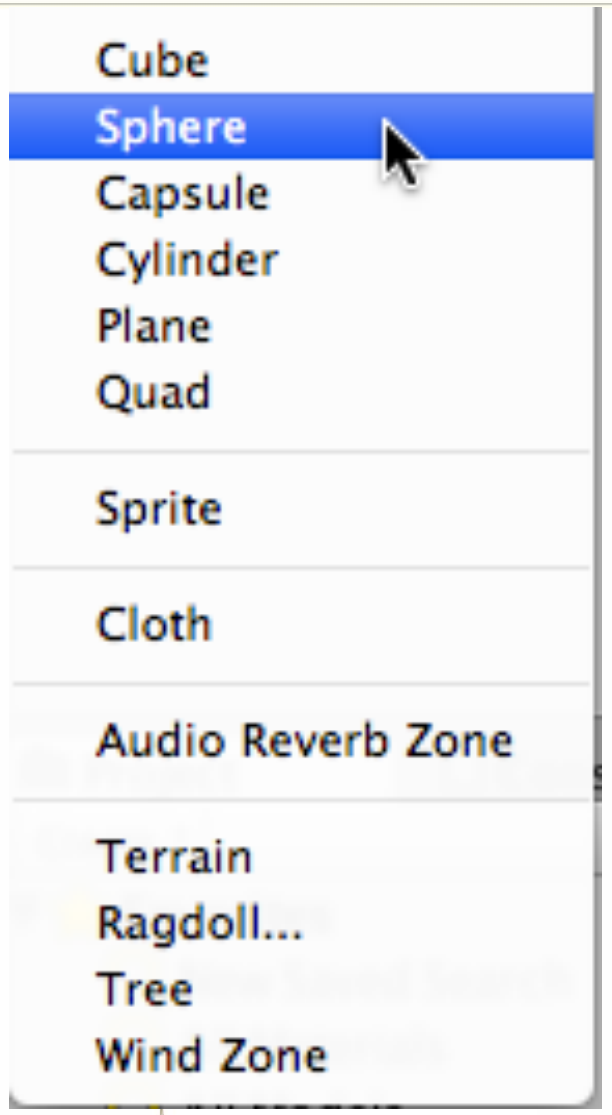


Per-vertex vs. per-pixel lighting (3)

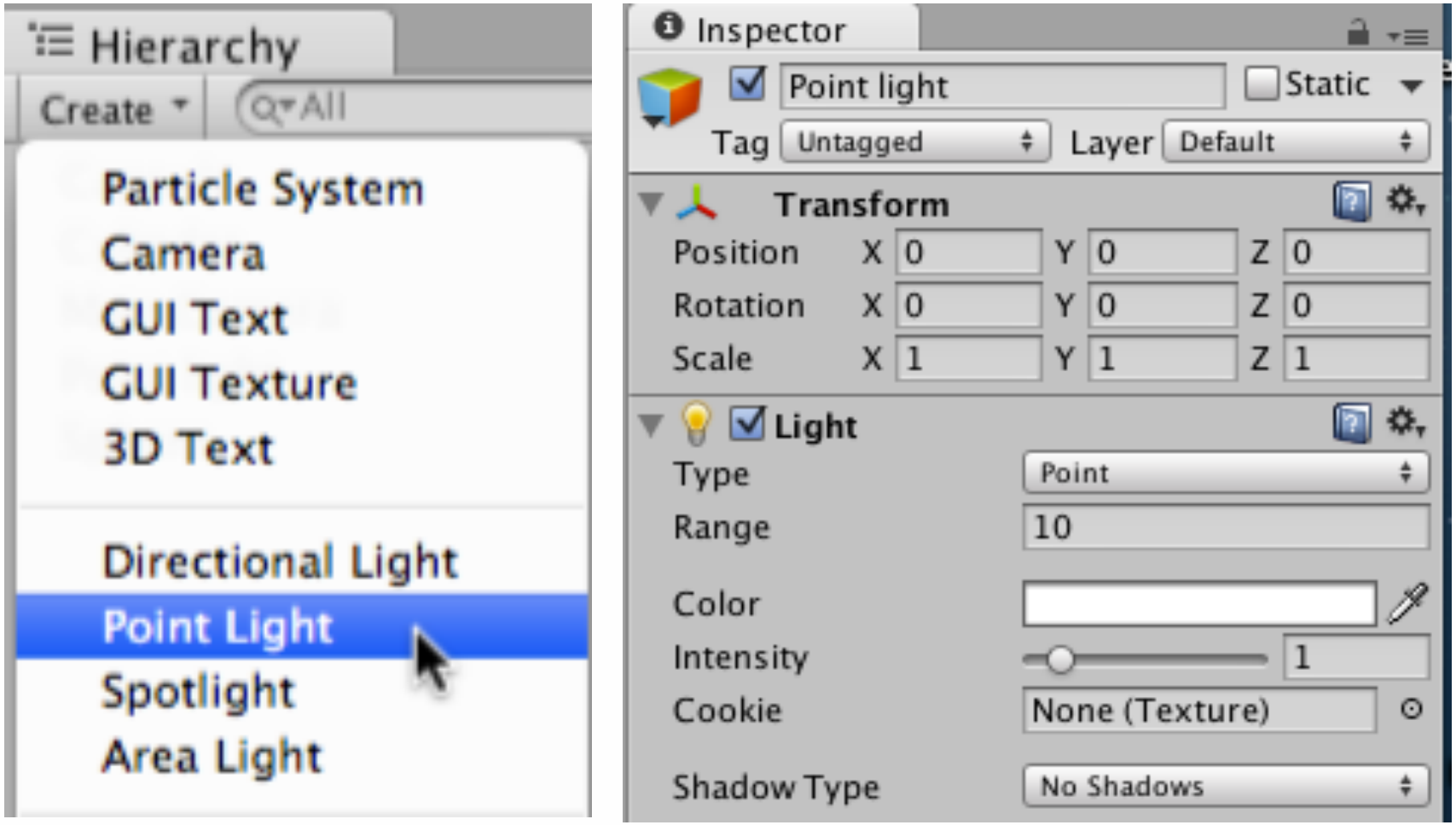


Adding some test geometry

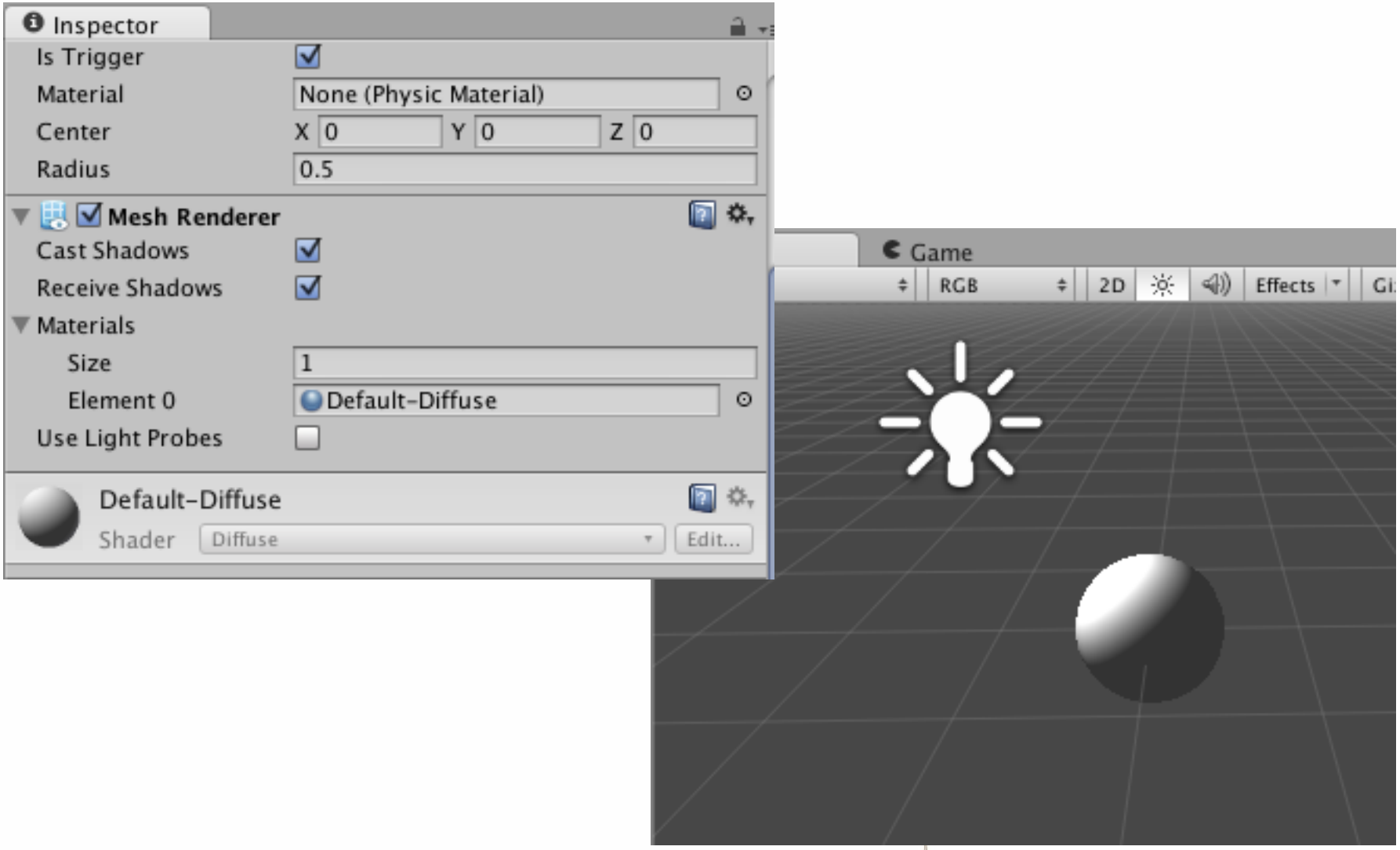




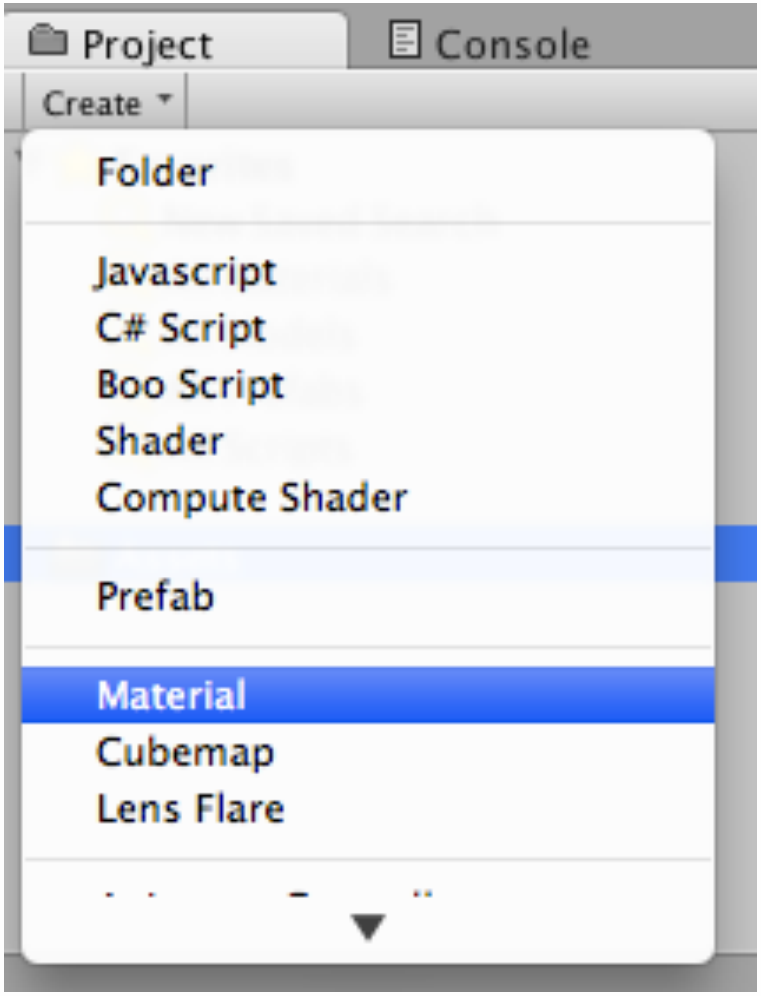
Adding a light source



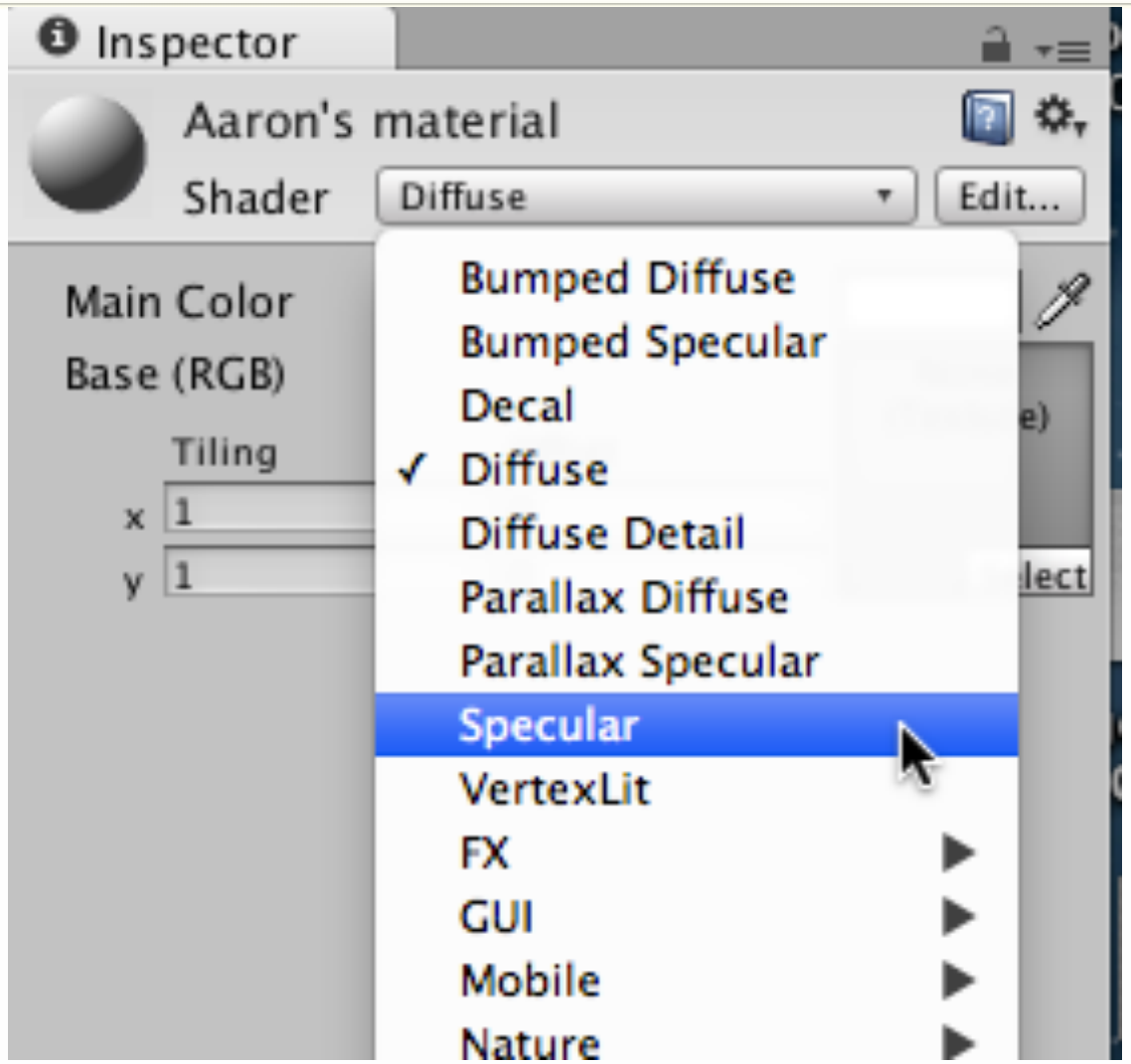
Boring sphere



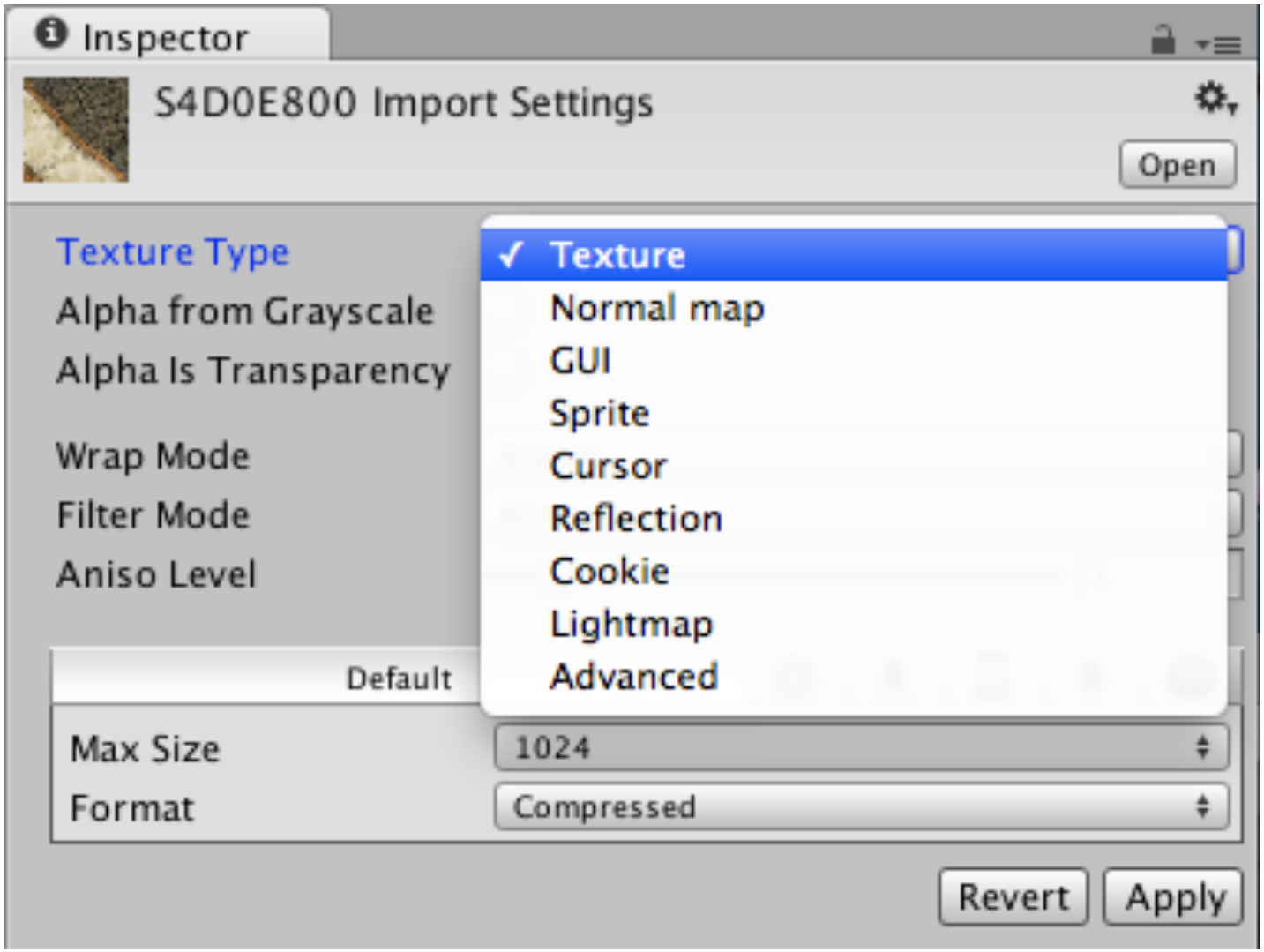
Making a new material



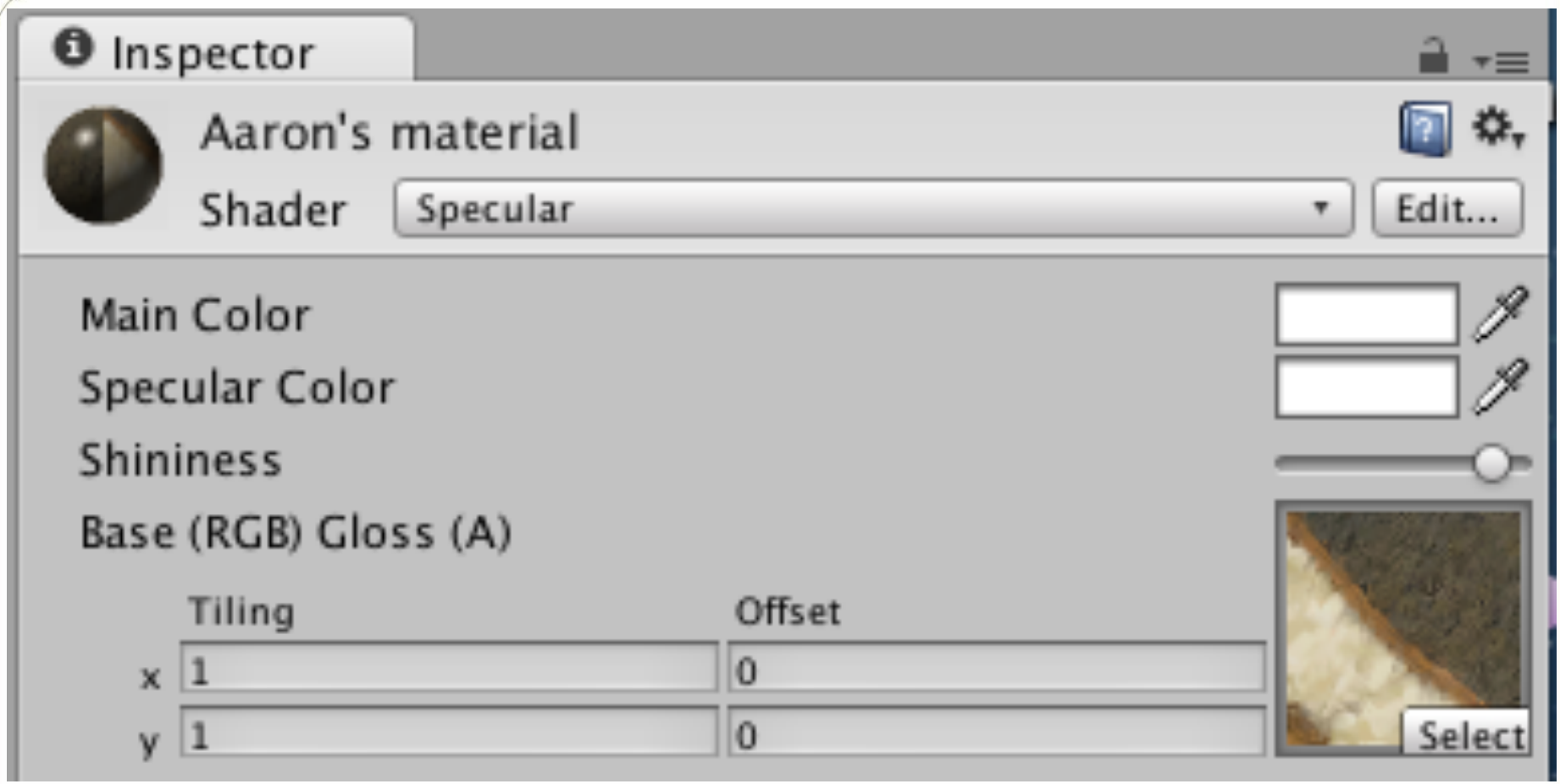
Choosing a shader



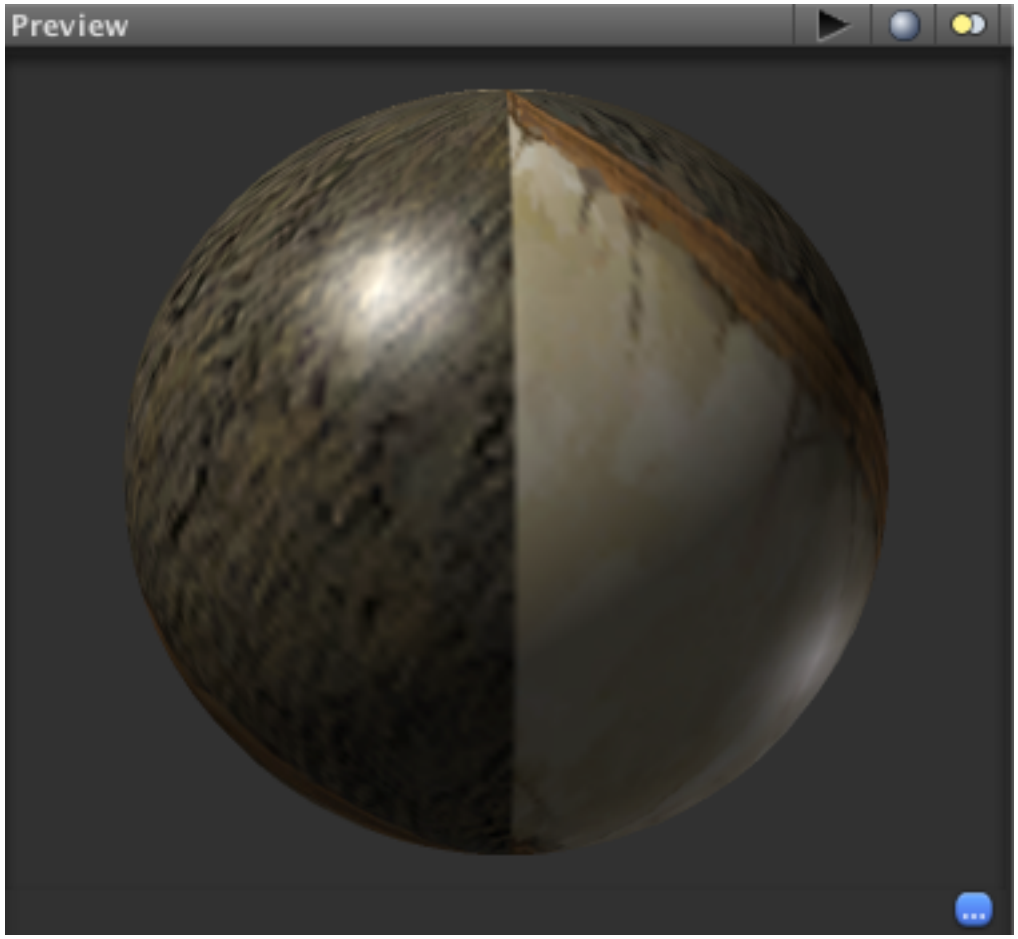
Importing an RGB(A) texture



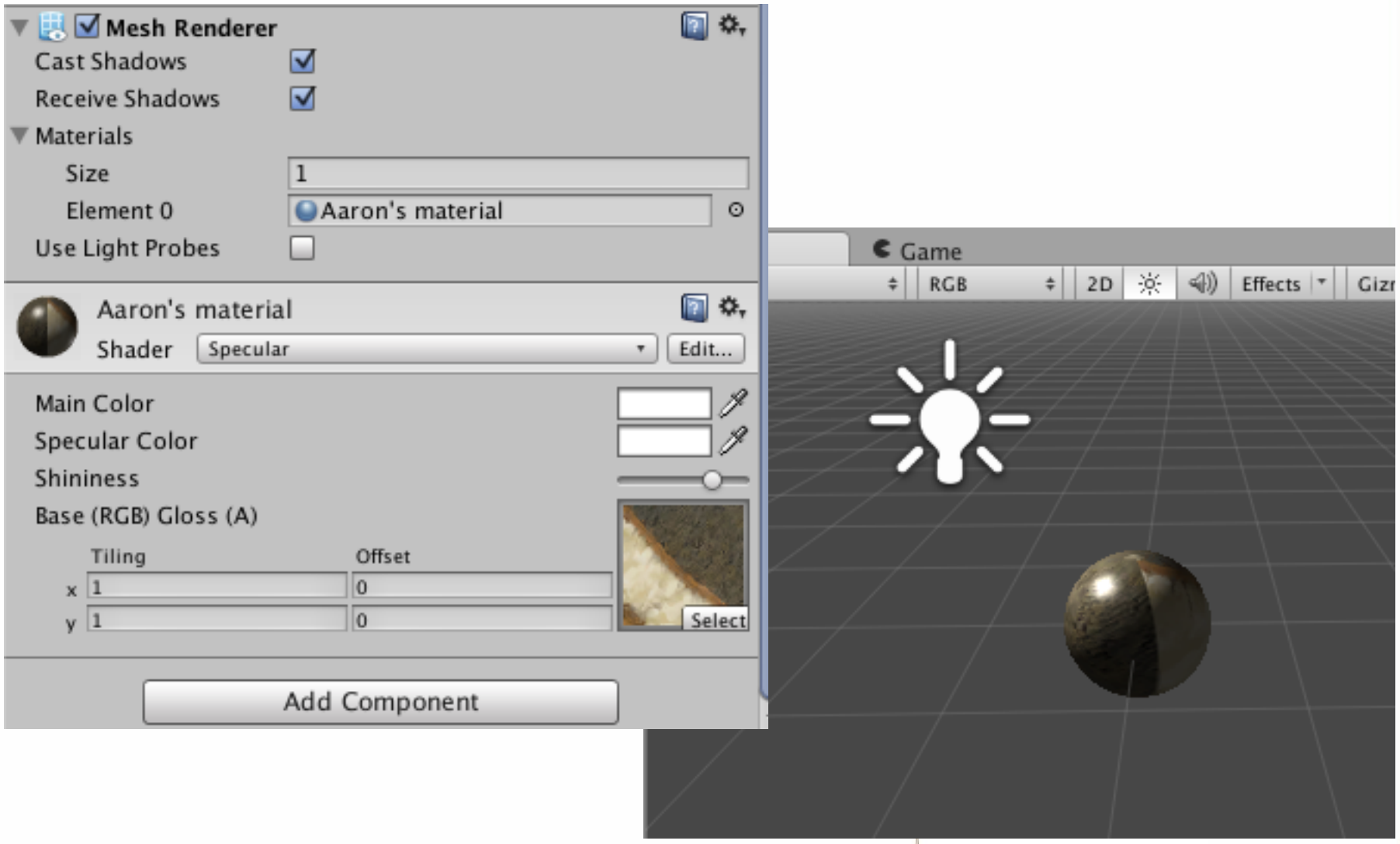
Apply RGB texture to material



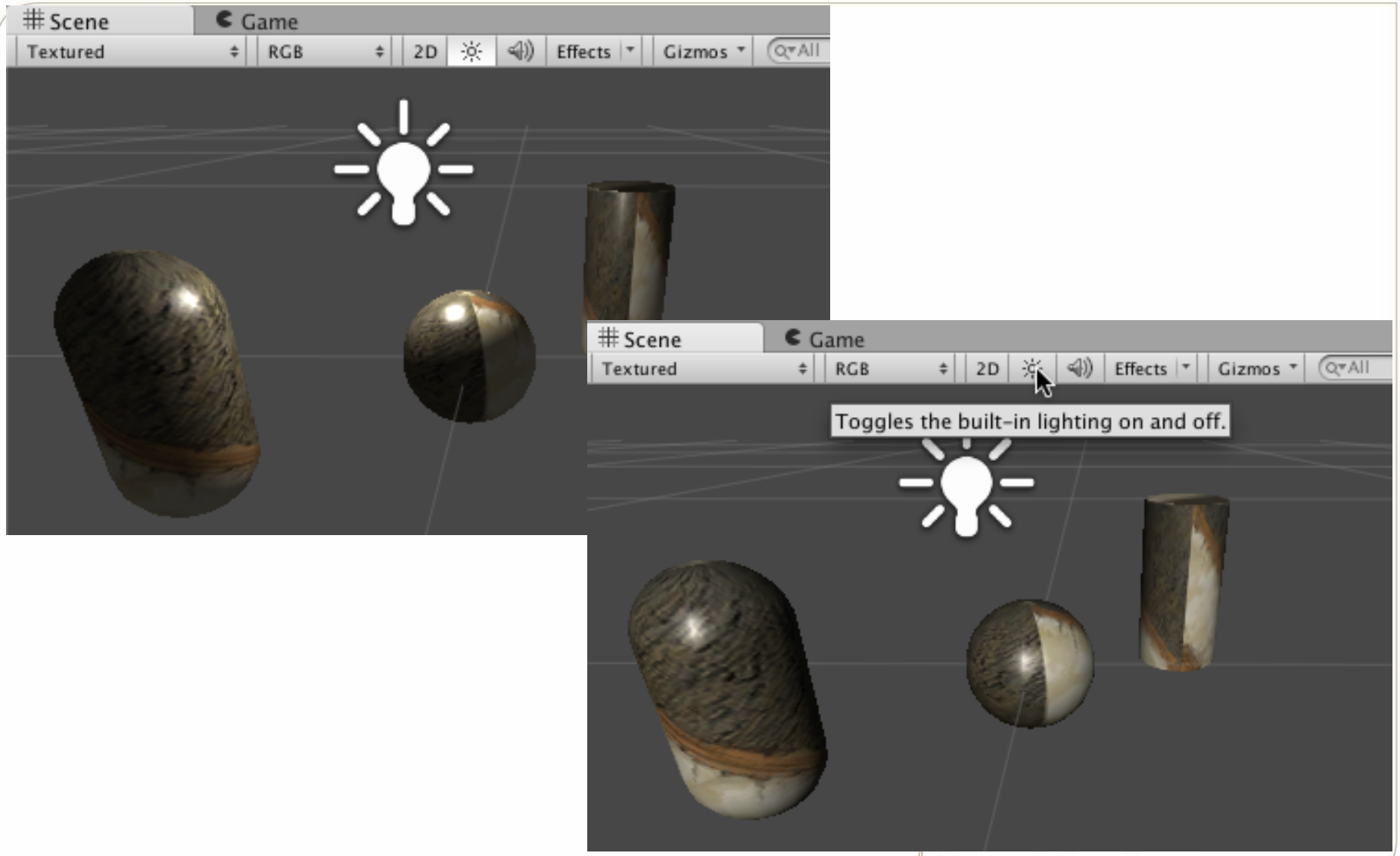
Material inspector display



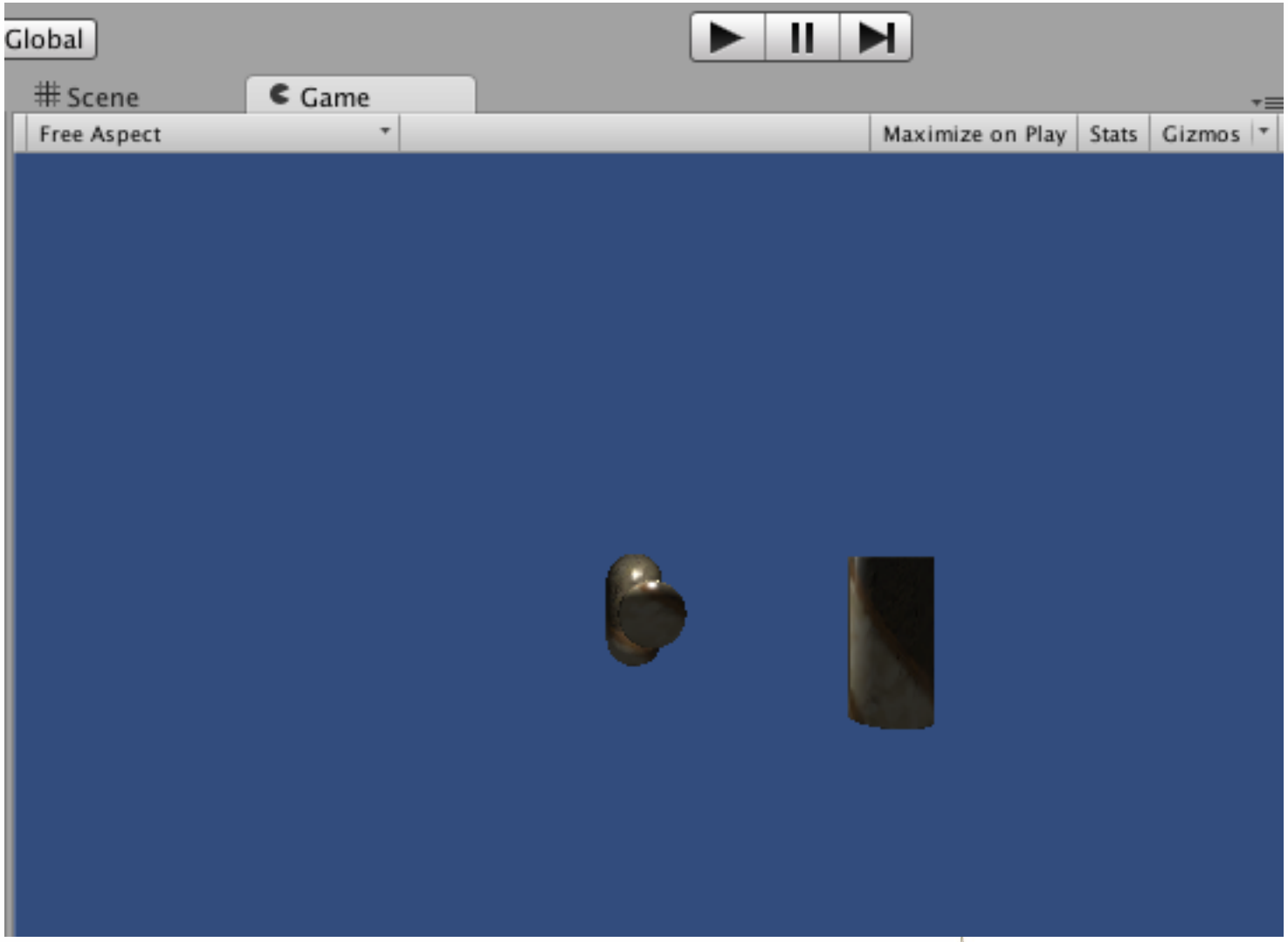
Apply material to object



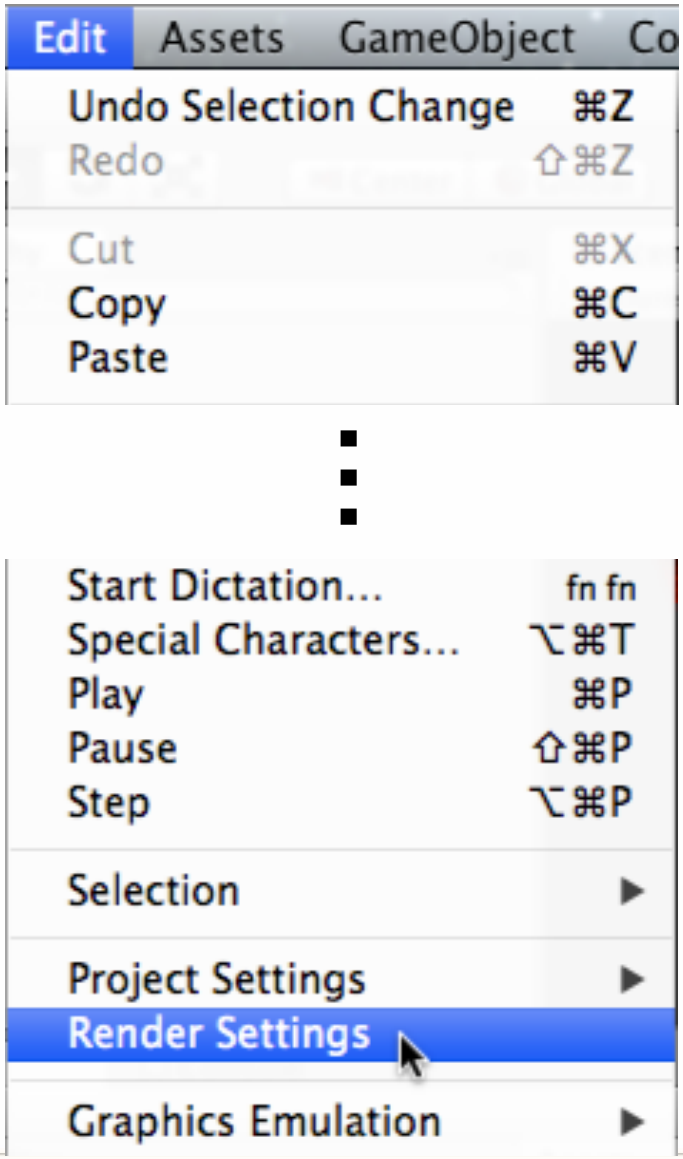
Turn off built-in lighting

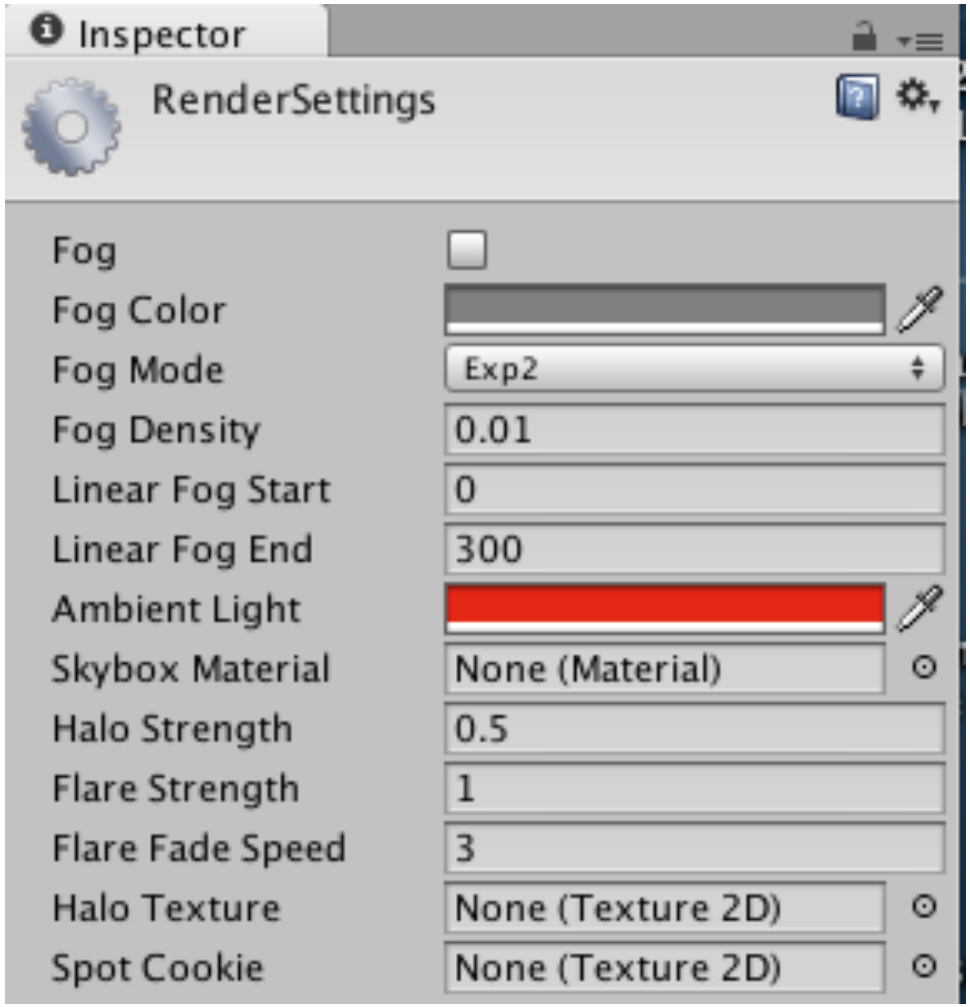


Game view


Global
Scene
Game
Free Aspect
Maximize on Play
Stats
Gizmos

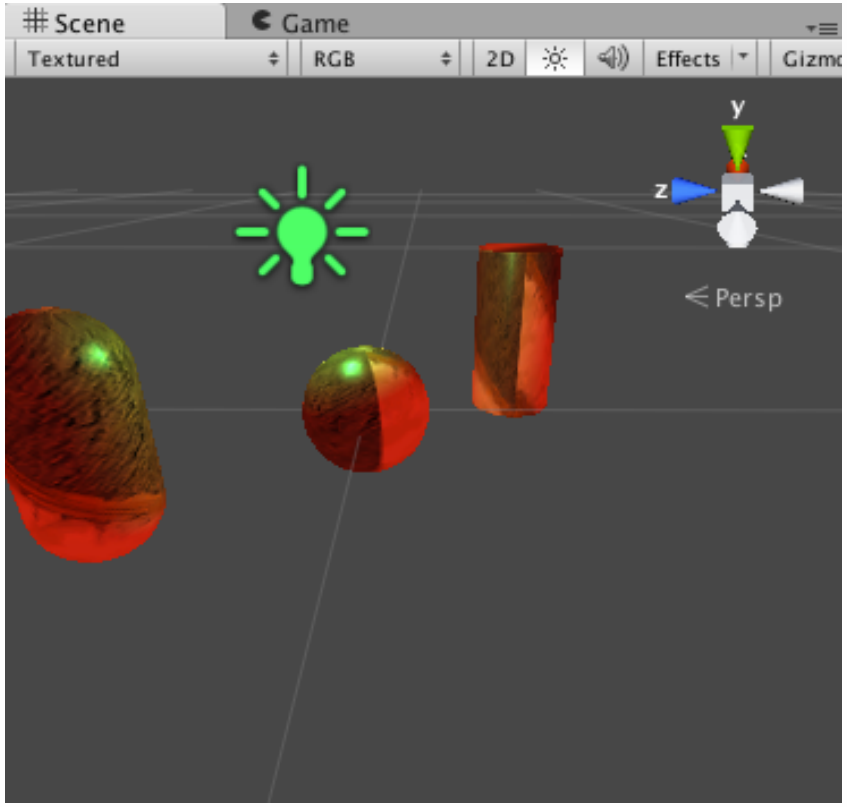
Ambient light

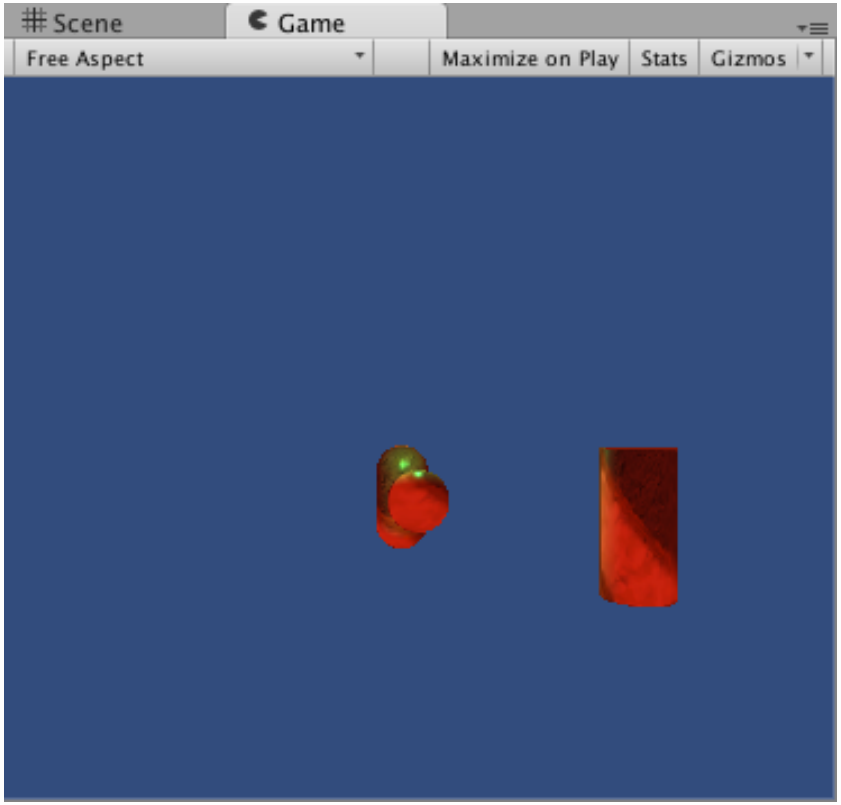




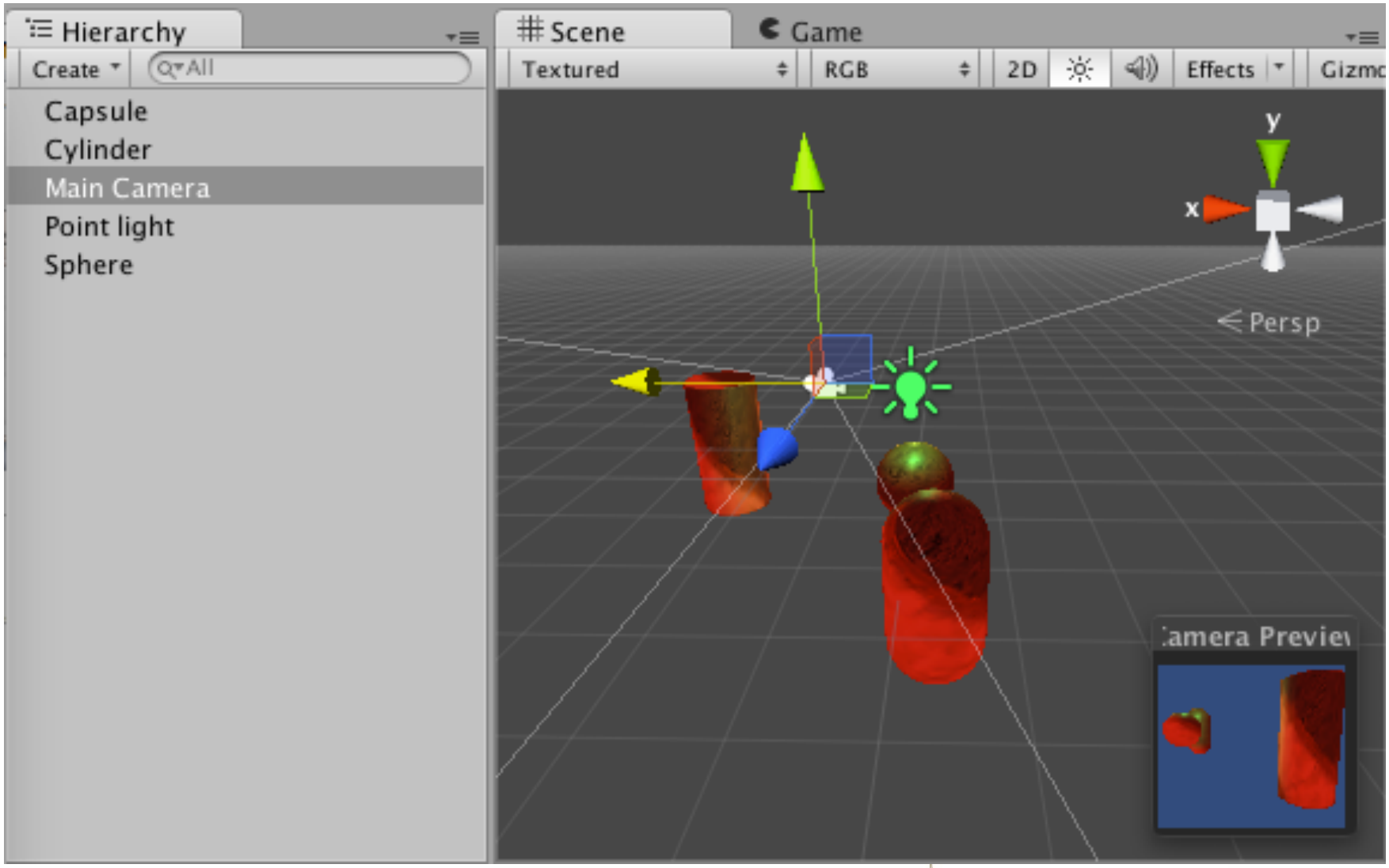
Georgia Institute of Technology

Comparing scene vs. game view

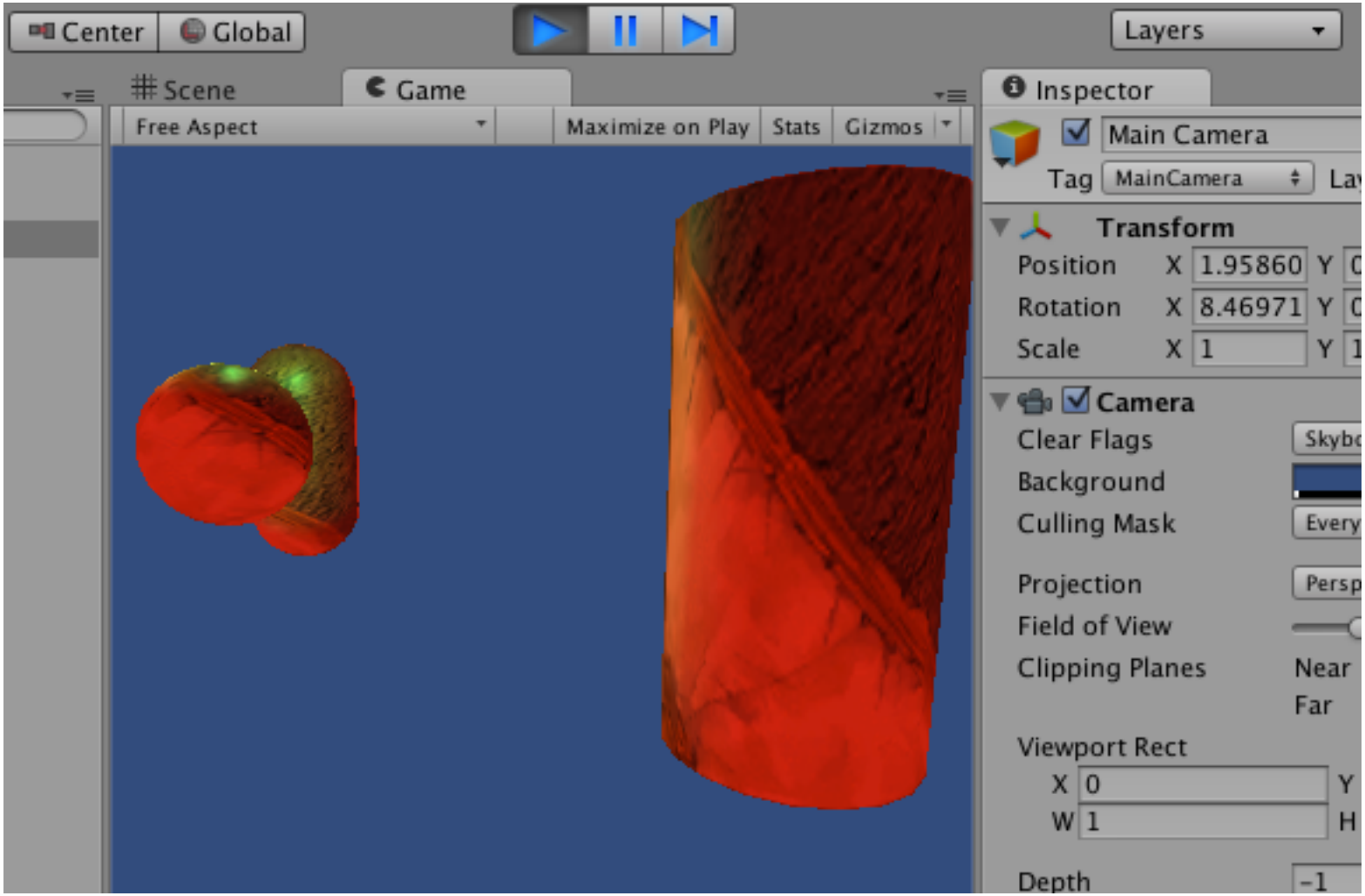




Adjusting the camera



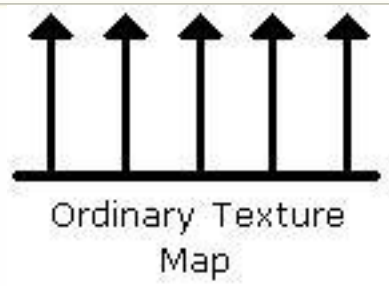
Change settings while game runs!

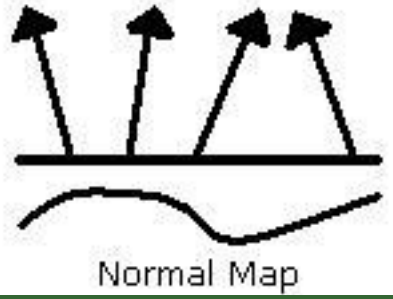


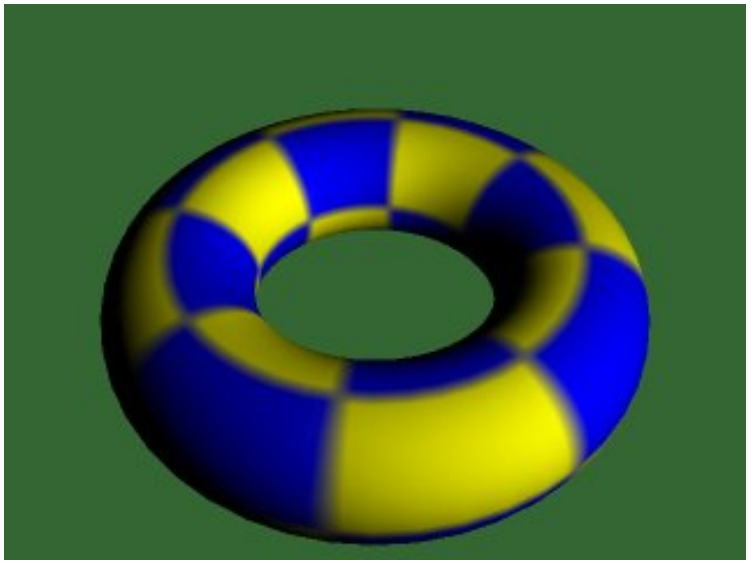
(note these changes are not saved)

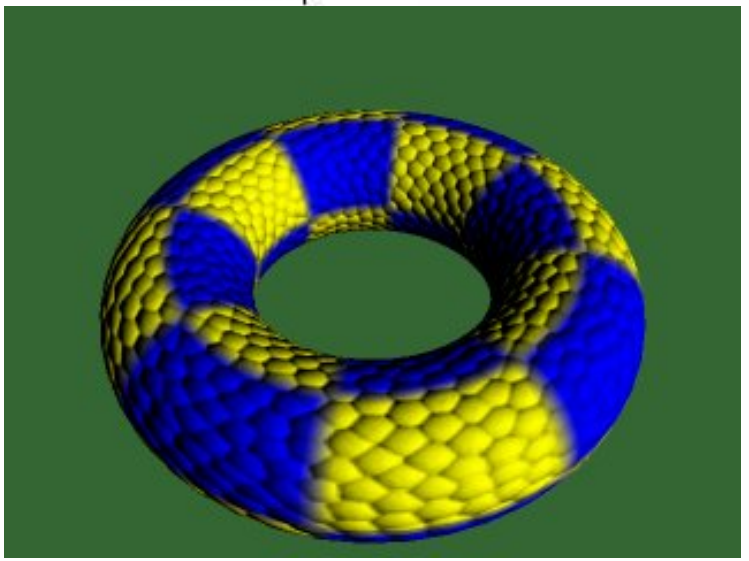
Georgia Institute of Technology

Normal mapping









Drawing from Søren Dreijer, "Bump Mapping Using Cg (3rd Edition),"
www.blacksmith-studios.dk/projects/downloads/bumpmapping_using_cg.php

Images from Paul Baker, "Simple Bumpmapping,"
www.paulsprojects.net/tutorials/simplebump/simplebump.html

Storing normals in textures

- Textures don't have to store color; we can store other things as well, like normals
  - Use r, g, b components to store, x, y, z of normal
- Problem: Textures take [0,1] values; normals need [-1,1] values
- Easy solution: "Range Compression"

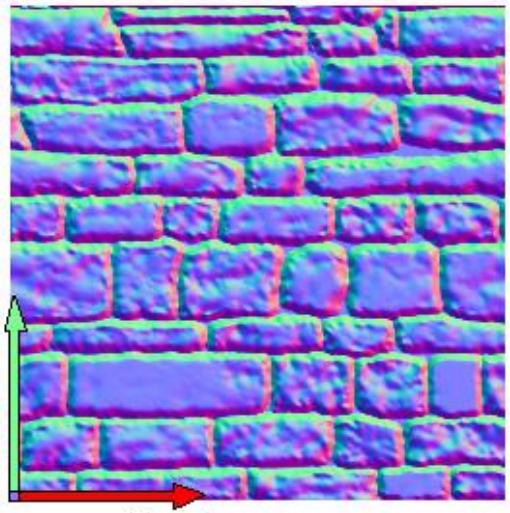
```
colorComponent = 0.5 * normalComponent + 0.5;

normalComponent = 2 * (colorComponent - 0.5);
```

From "The Cg Tutorial," p. 202

Normal mapping example

Normal map

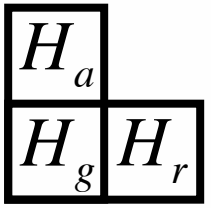




From Søren Dreijer, "Bump Mapping Using Cg (3rd Edition),"
www.blacksmith-studios.dk/projects/downloads/bumpmapping_using_cg.php

Creating normal map from height field

- Height field H(u,v)

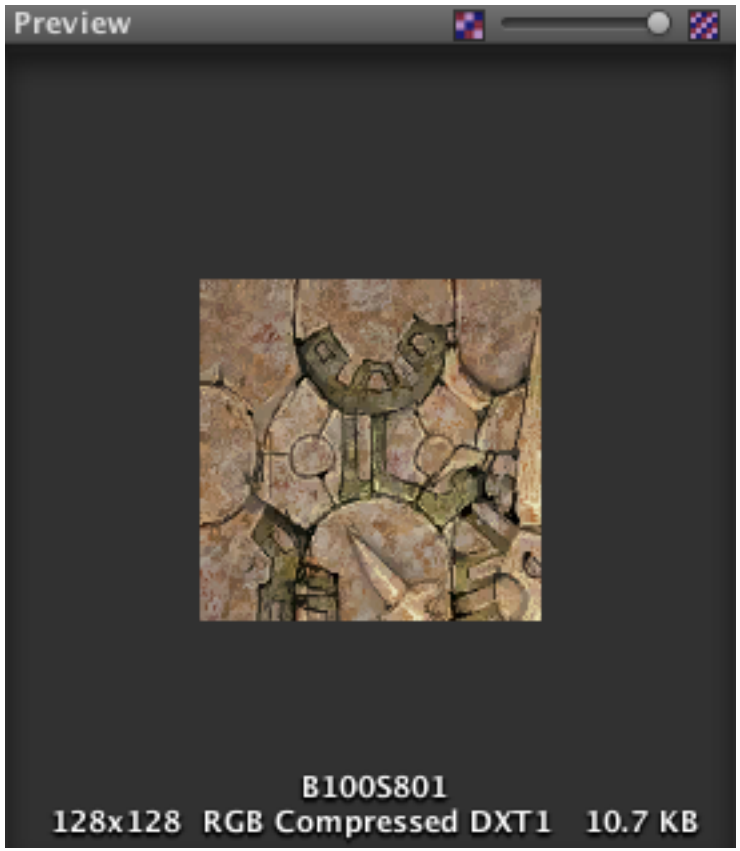
$$normal = \frac{\left(H_g - H_r, H_g - H_a, 1\right)}{\left\|\left(H_g - H_r, H_g - H_a, 1\right)\right\|}$$

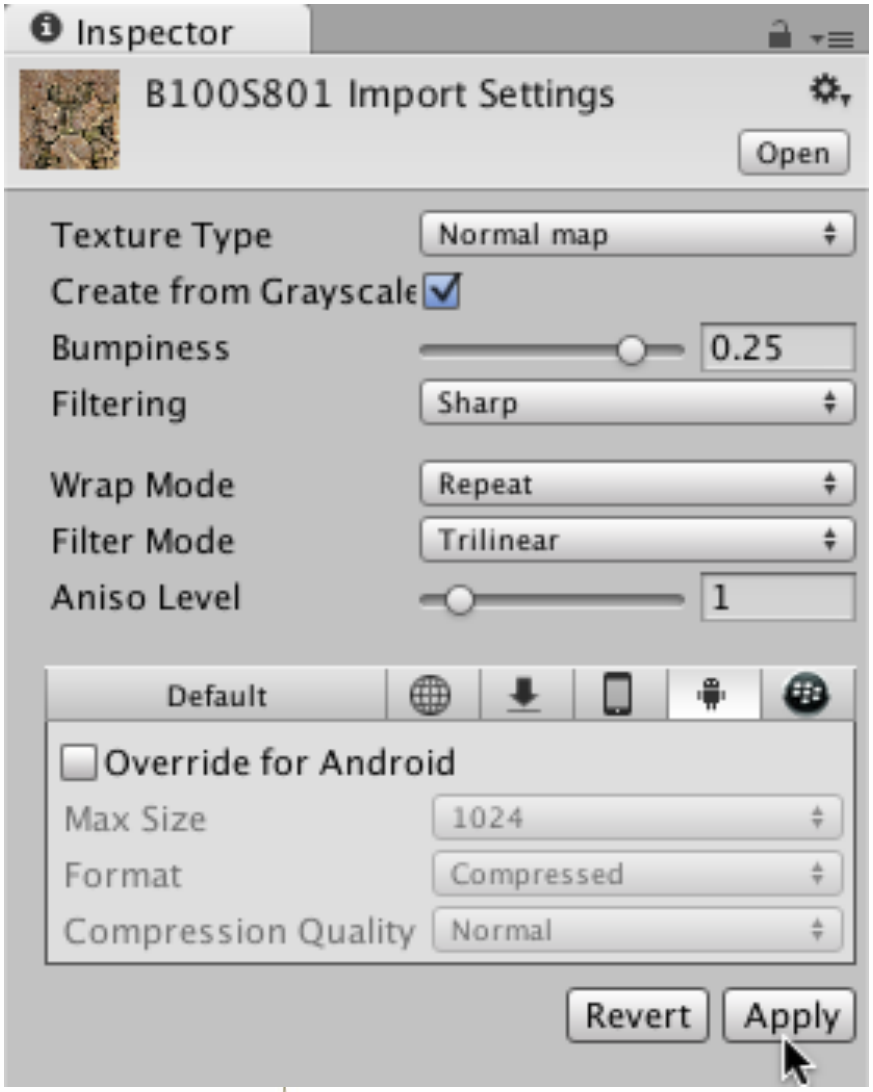


- In flat regions, normal is (0,0,1), i.e. pointing "up"

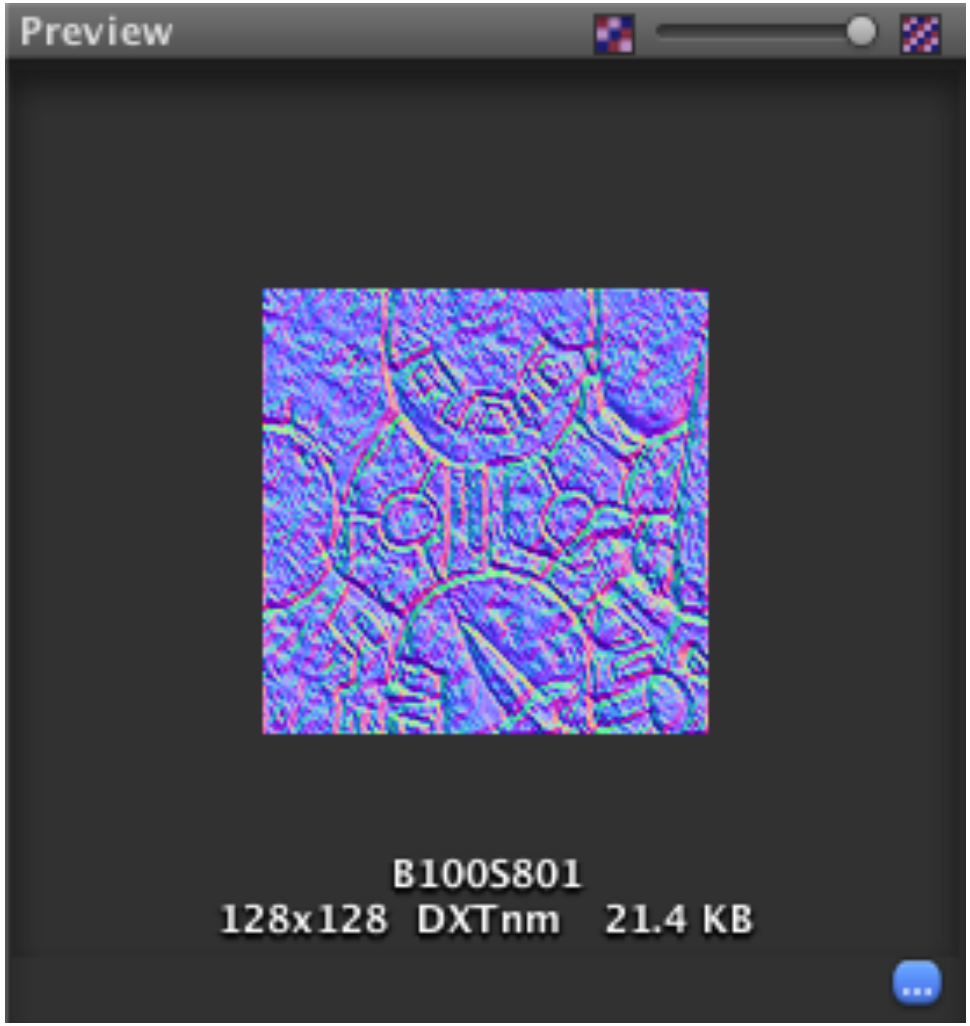
From "The Cg Tutorial," p. 203

Create a normal map

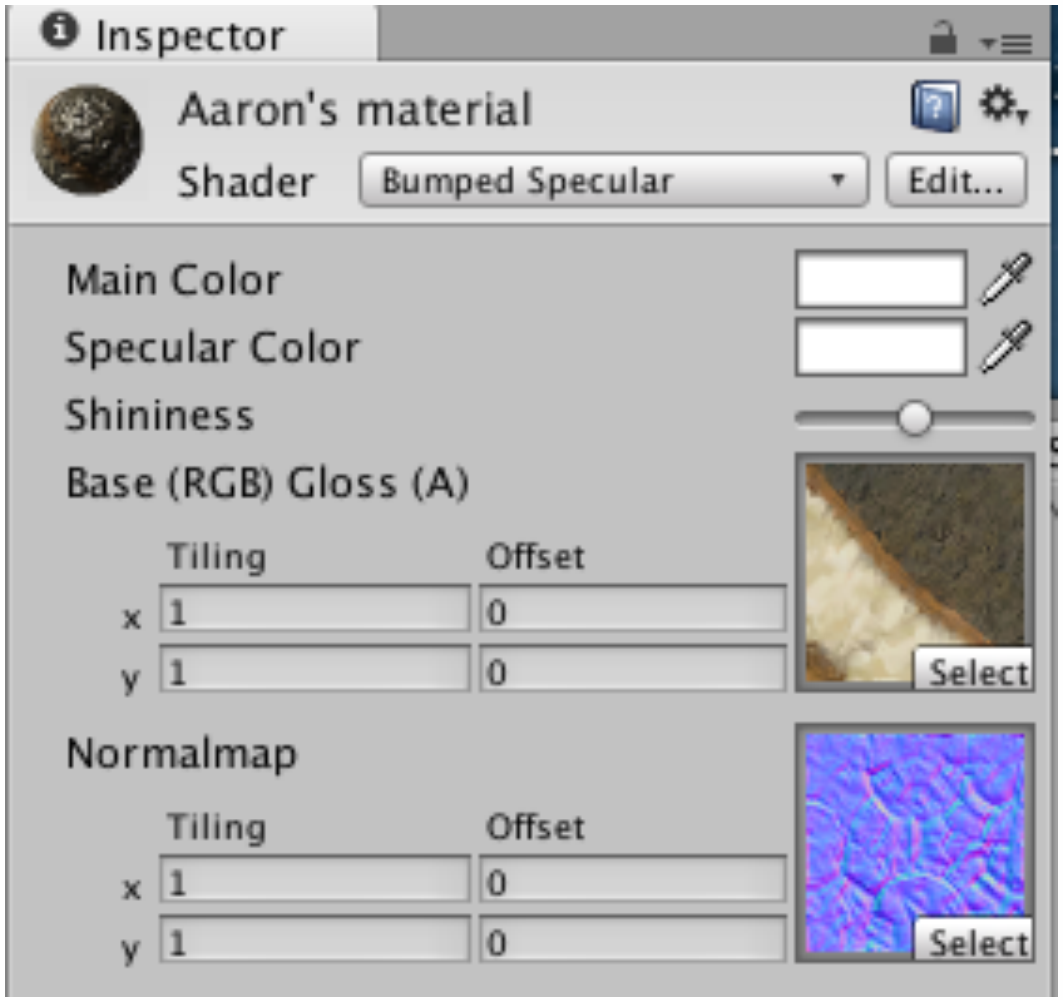


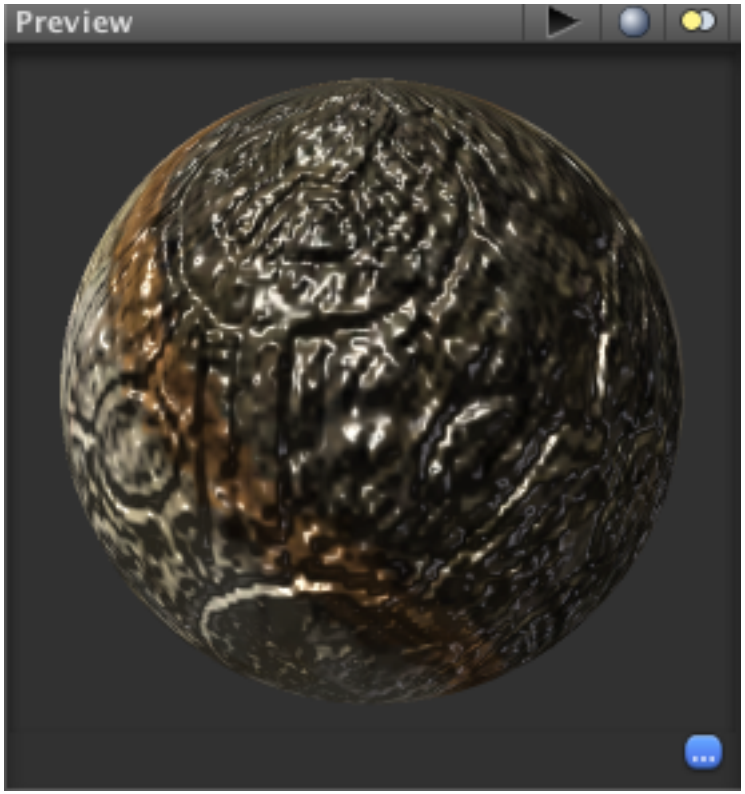


Normal map displayed as RGB

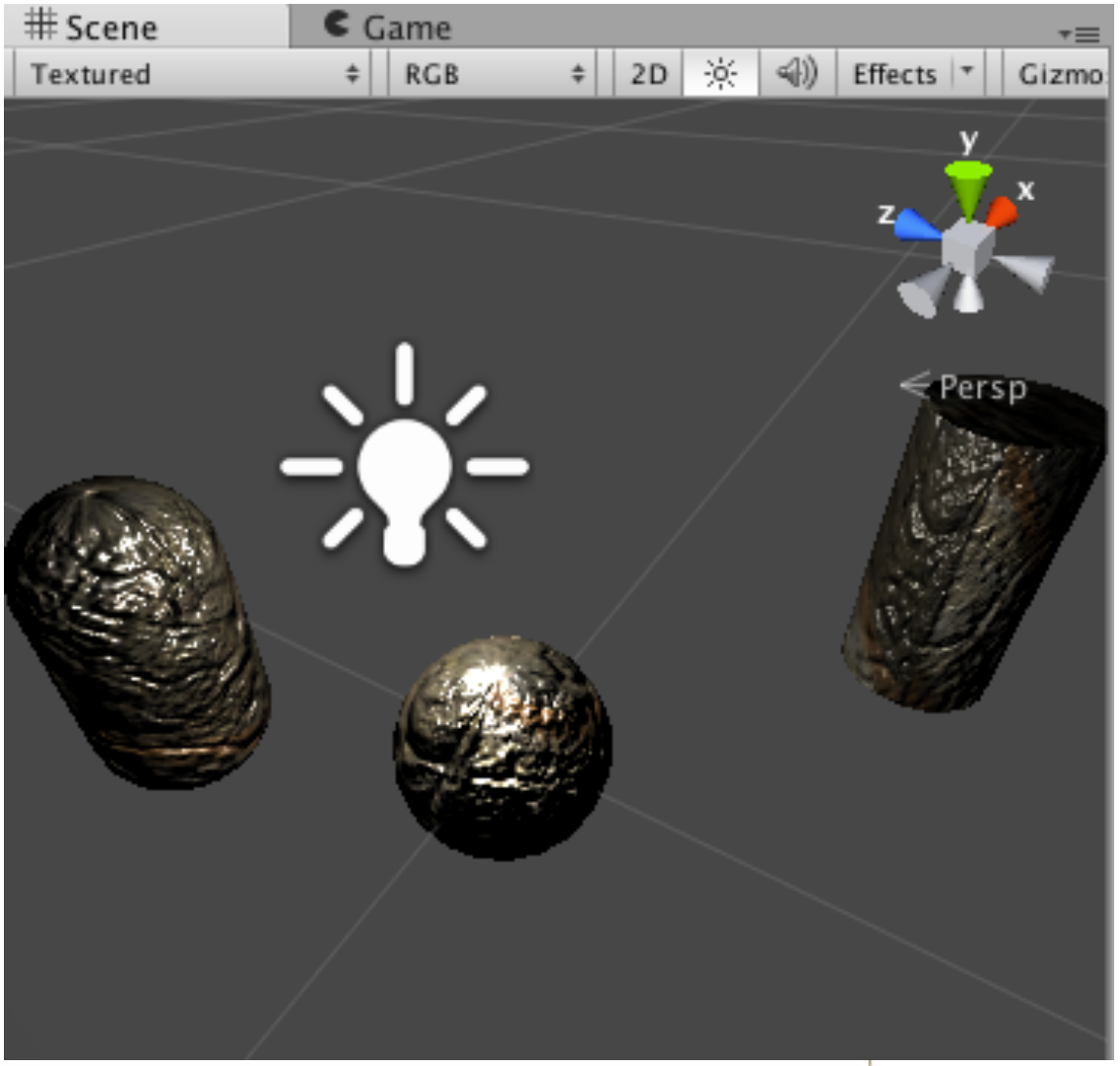


Apply normal map to material

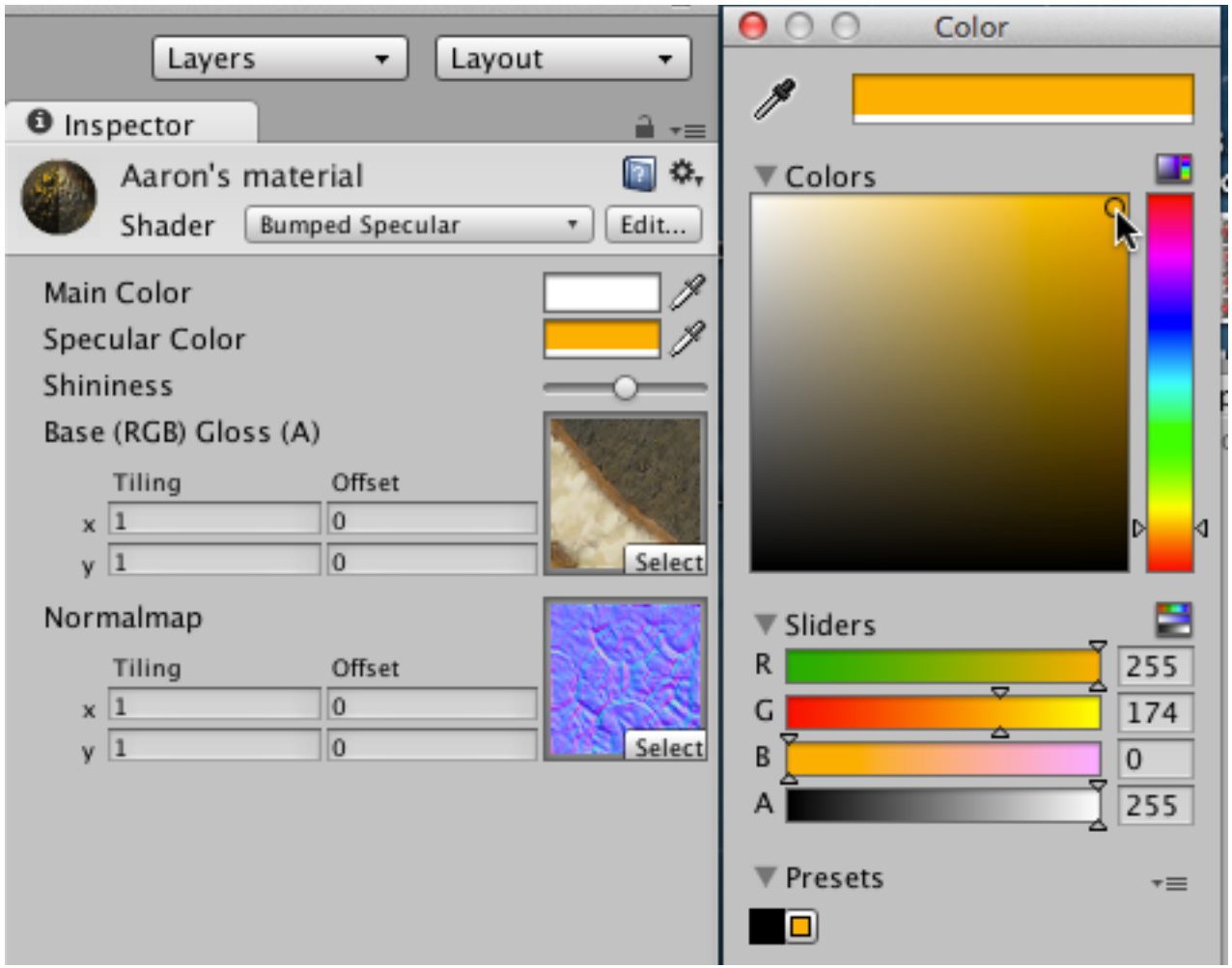




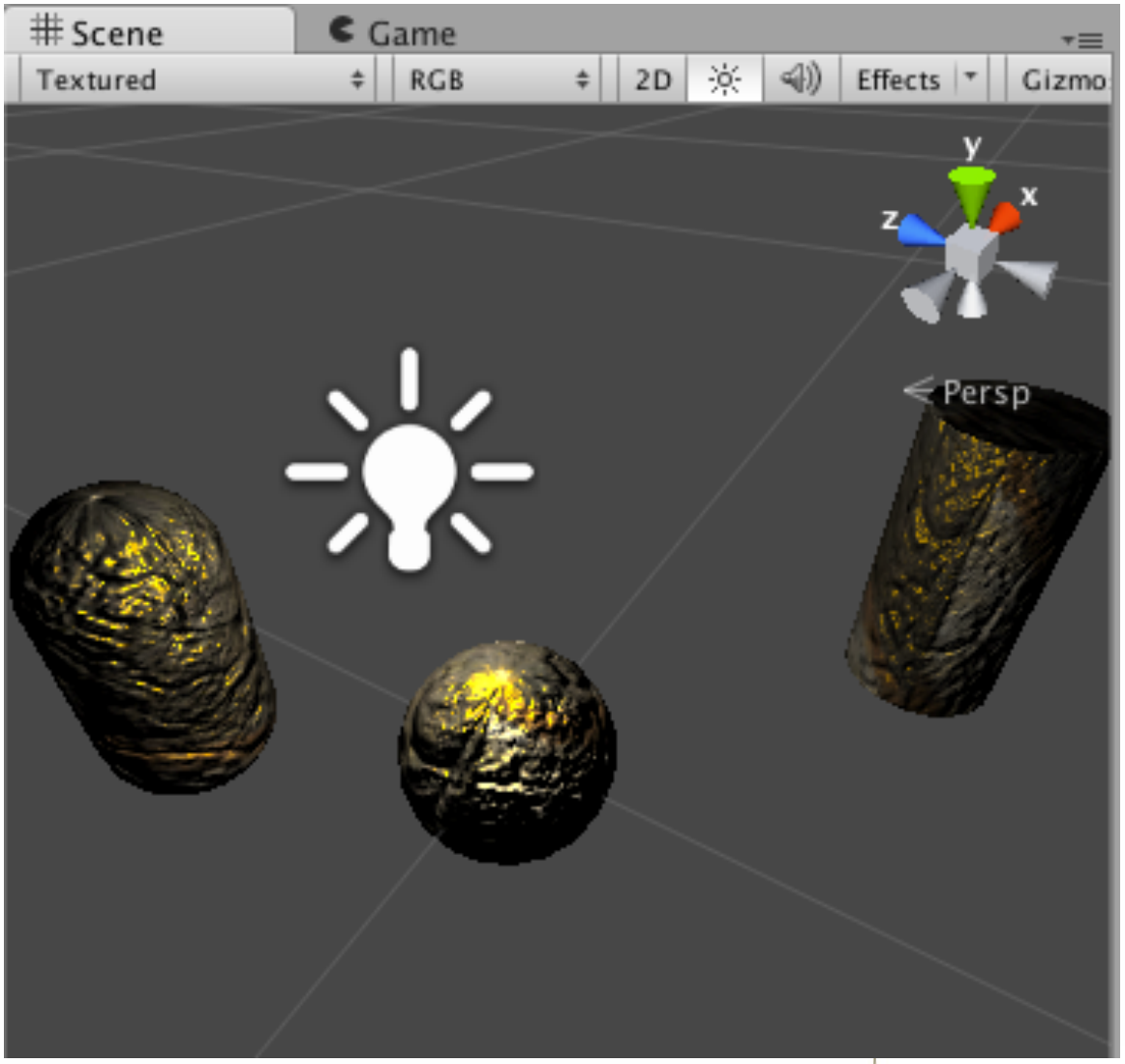
Normal mapping in action



Coloring the specular highlights…



…makes them stand out



Every video game needs barrels!

Barrel

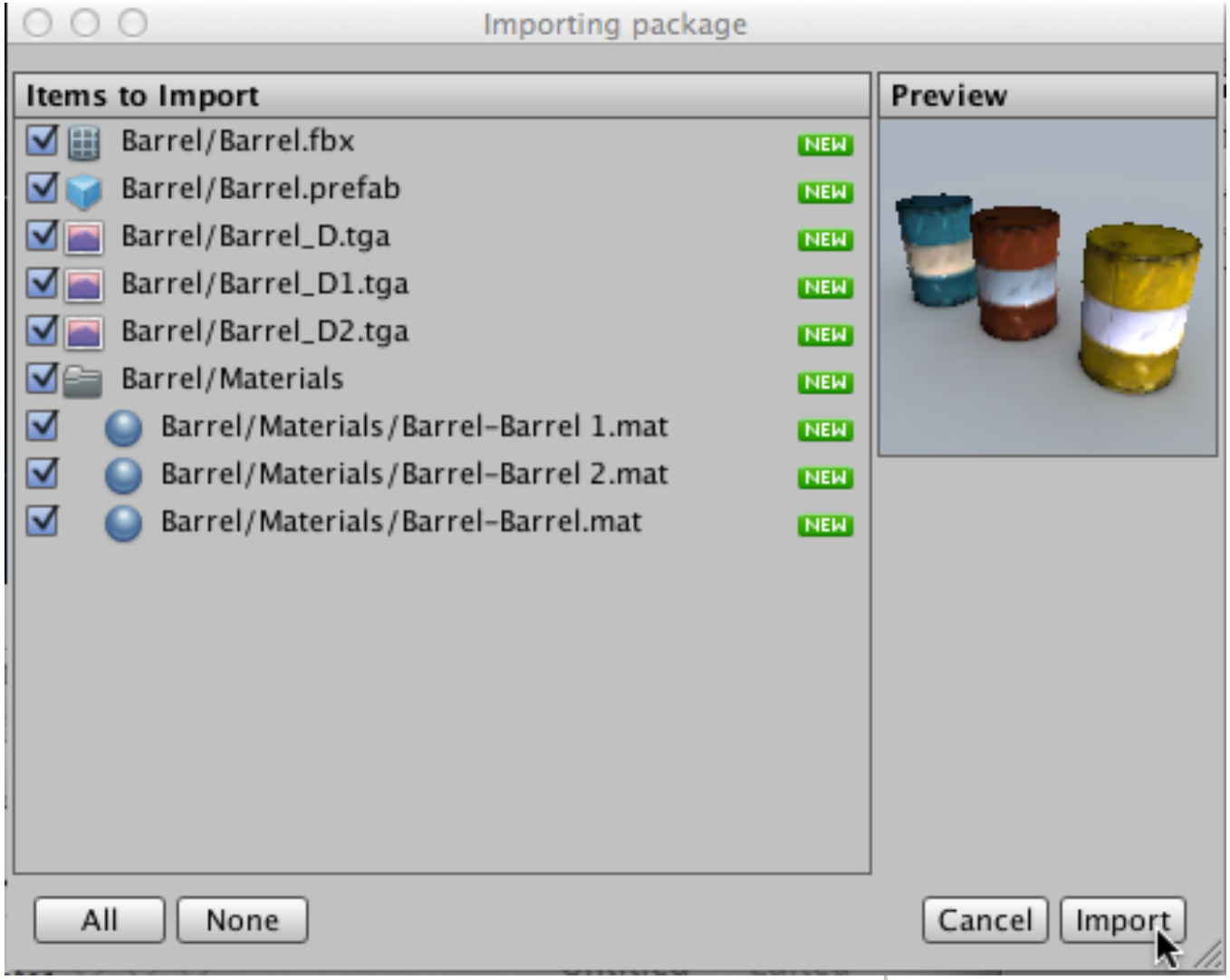
Category: 3D Models/Props/Industrial
Publisher: Universal Image
Rating: ★★★★★ (45)
Price: Free

Download

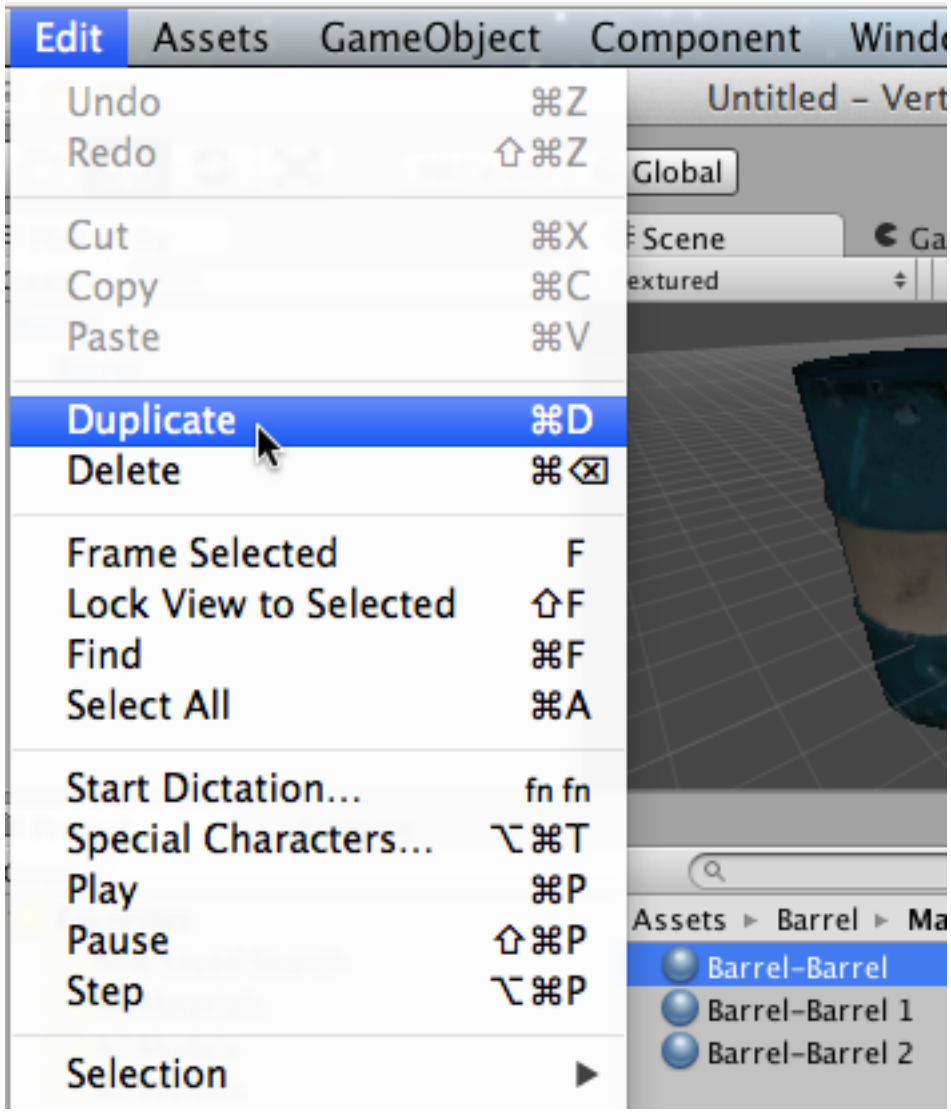
This packages contains a single barrel model with three different color maps, which is optimized for the Unity game engine.

Georgia Institute of Technology

Importing package from the asset store



Materials are global!



- Editing a material *anywhere* causes every instance of that material to change *everywhere*
- Make duplicate materials as needed

Georgia Institute of Technology

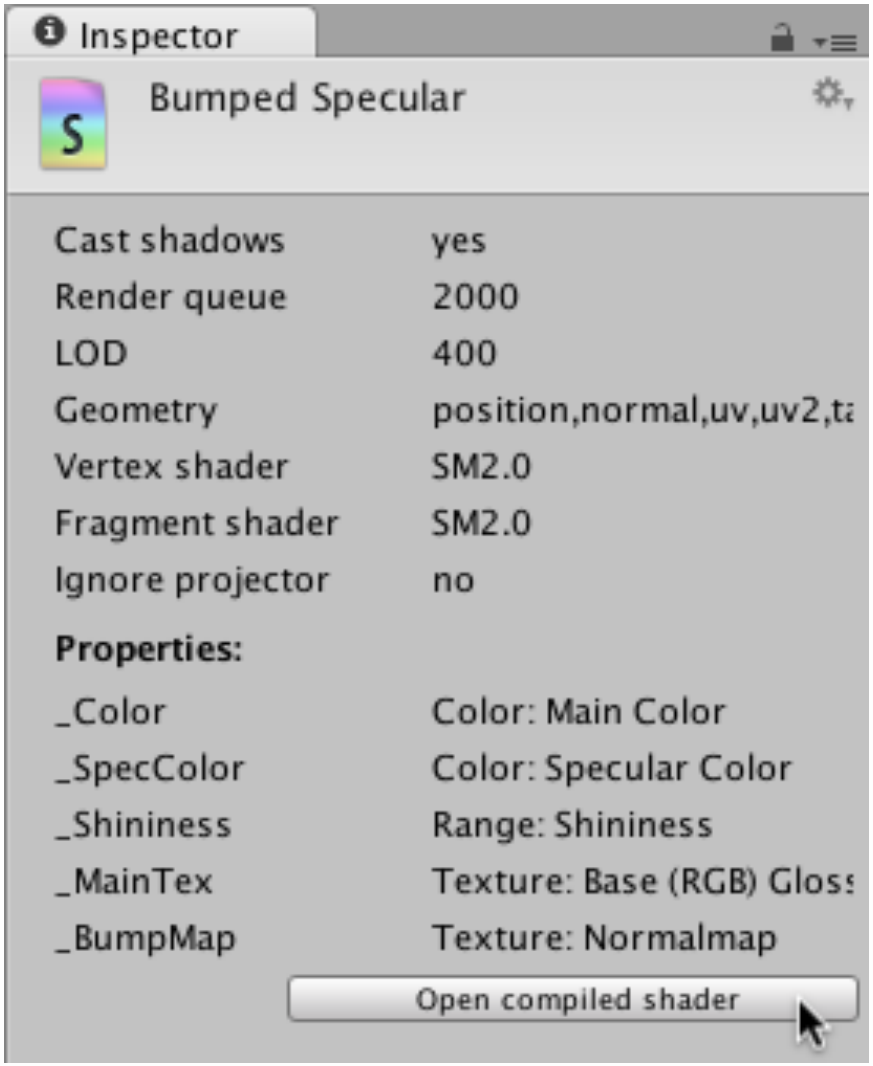
C# Script (Setup)

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class RotateObject : MonoBehaviour {
5
6     public float rotateSpeed;
7     public bool rotateX;
8     public bool rotateY;
9     public bool rotateZ;
10
11    private GUIStyle guiStyle;
12
13    // Use this for initialization
14    void Start () {
15        rotateX = false;
16        rotateY = false;
17        rotateZ = false;
18        rotateSpeed = 30f;
19
20        guiStyle = new GUIStyle ();
21        guiStyle.alignment = TextAnchor.MiddleCenter;
22        guiStyle.normal.textColor = Color.white;
23        guiStyle.fontSize = 30;
24    }
```

C# Script (Updates)

```
26      void OnGUI () {
27          Rect textArea = new Rect (100, 20, Screen.width - 200, 50);
28          GUI.Label (textArea, "Barrel model & textures by Universal Image", guiStyle);
29      }
30
31      // Update is called once per frame
32      void Update () {
33          if (Input.GetKeyDown(KeyCode.Alpha0)) rotateSpeed = 0f;
34          if (Input.GetKeyDown(KeyCode.Alpha1)) rotateSpeed = 10f;
35          if (Input.GetKeyDown(KeyCode.Alpha2)) rotateSpeed = 20f;
36          if (Input.GetKeyDown(KeyCode.Alpha3)) rotateSpeed = 30f;
37          if (Input.GetKeyDown(KeyCode.Alpha4)) rotateSpeed = 40f;
38          if (Input.GetKeyDown(KeyCode.Alpha5)) rotateSpeed = 50f;
39
40          if (Input.GetKeyDown(KeyCode.X)) rotateX = !rotateX;
41          if (Input.GetKeyDown(KeyCode.Y)) rotateY = !rotateY;
42          if (Input.GetKeyDown(KeyCode.Z)) rotateZ = !rotateZ;
43
44          float rotateIncrement = Time.deltaTime * rotateSpeed;
45
46          foreach (Transform child in transform) {
47              if (rotateX) child.Rotate (Vector3.right * rotateIncrement);
48              if (rotateY) child.Rotate (Vector3.up * rotateIncrement);
49              if (rotateZ) child.Rotate (Vector3.forward * rotateIncrement);
50          }
51      }
52  }
```

Diving in


Inspector
Bumped Specular
Cast shadows yes
Render queue 2000
LOD 400
Geometry position,normal,uv,uv2,t
Vertex shader SM2.0
Fragment shader SM2.0
Ignore projector no
Properties:
_Color Color: Main Color
_SpecColor Color: Specular Color
_Shininess Range: Shininess
_MainTex Texture: Base (RGB) Gloss
_BumpMap Texture: Normalmap
Open compiled shader

Huh?

Compiled-Normal-BumpSpec.s

```
Shader "Bumped Specular" {
Properties {
    _Color ("Main Color", Color) = (1,1,1,1)
    _SpecColor ("Specular Color", Color) = (0.5, 0.5, 0.5, 1)
    _Shininess ("Shininess", Range (0.03, 1)) = 0.078125
    _MainTex ("Base (RGB) Gloss (A)", 2D) = "white" {}
    _BumpMap ("Normalmap", 2D) = "bump" {}
}
SubShader {
    Tags { "RenderType"="Opaque" }
    LOD 400


    Pass {
        Name "FORWARD"
        Tags { "LightMode" = "ForwardBase" }
Program "vp" {
// Vertex combos: 12
//   opengl - ALU: 7 to 80
//   d3d9 - ALU: 7 to 83
//   d3d11 - ALU: 7 to 66, TEX: 0 to 0, FLOW: 1 to 1
//   d3d11_9x - ALU: 7 to 66, TEX: 0 to 0, FLOW: 1 to 1
SubProgram "opengl " {
Keywords { "DIRECTIONAL" "LIGHTMAP_OFF" "DIRLIGHTMAP_OFF" "SHADOWS_OFF" }
```

Huh??

```
25 Bind "vertex" Vertex
26 Bind "tangent" ATTR14
27 Bind "normal" Normal
28 Bind "texcoord" TexCoord0
29 Vector 13 [_WorldSpaceCameraPos]
30 Vector 14 [_WorldSpaceLightPos0]
31 Vector 15 [unity_SHAr]
32 Vector 16 [unity_SHAg]
33 Vector 17 [unity_SHAb]
34 Vector 18 [unity_SHBr]
35 Vector 19 [unity_SHBg]
36 Vector 20 [unity_SHBb]
37 Vector 21 [unity_SHC]
38 Matrix 5 [_Object2World]
39 Matrix 9 [_World2Object]
40 Vector 22 [unity_Scale]
41 Vector 23 [_MainTex_ST]
42 Vector 24 [_BumpMap_ST]
43 "!!ARBvp1.0
44 # 44 ALU
45 PARAM c[25] = { { 1 },
46         state.matrix.mvp,
47         program.local[5..24] };
```

Georgia Institute of Technology

Huh???

```
52 MUL R1.xyz, vertex.normal, c[22].w;
53 DP3 R2.w, R1, c[6];
54 DP3 R0.x, R1, c[5];
55 DP3 R0.z, R1, c[7];
56 MOV R0.y, R2.w;
57 MOV R0.w, c[0].x;
58 MUL R1, R0.xyzz, R0.yzzx;
59 DP4 R2.z, R0, c[17];
60 DP4 R2.y, R0, c[16];
61 DP4 R2.x, R0, c[15];
62 MUL R0.w, R2, R2;
63 MAD R0.w, R0.x, R0.x, -R0;
64 DP4 R0.z, R1, c[20];
65 DP4 R0.y, R1, c[19];
66 DP4 R0.x, R1, c[18];
67 ADD R0.xyz, R2, R0;
68 MUL R1.xyz, R0.w, c[21];
69 ADD result.texcoord[2].xyz, R0, R1;
```