



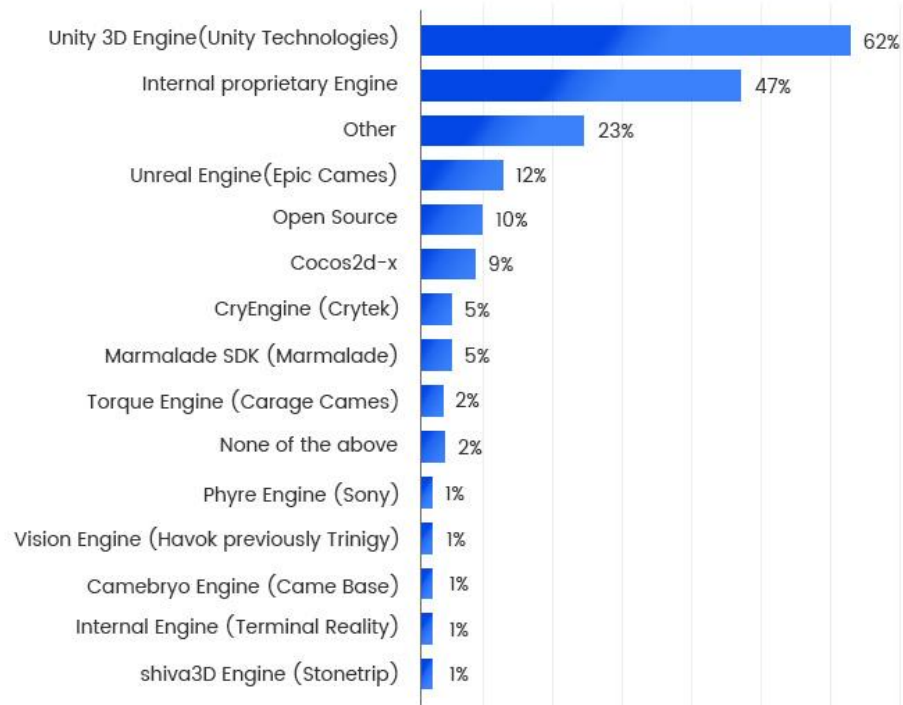
A Case Study: Unity

Computer Graphics

Yu-Ting Wu

Unity Overview

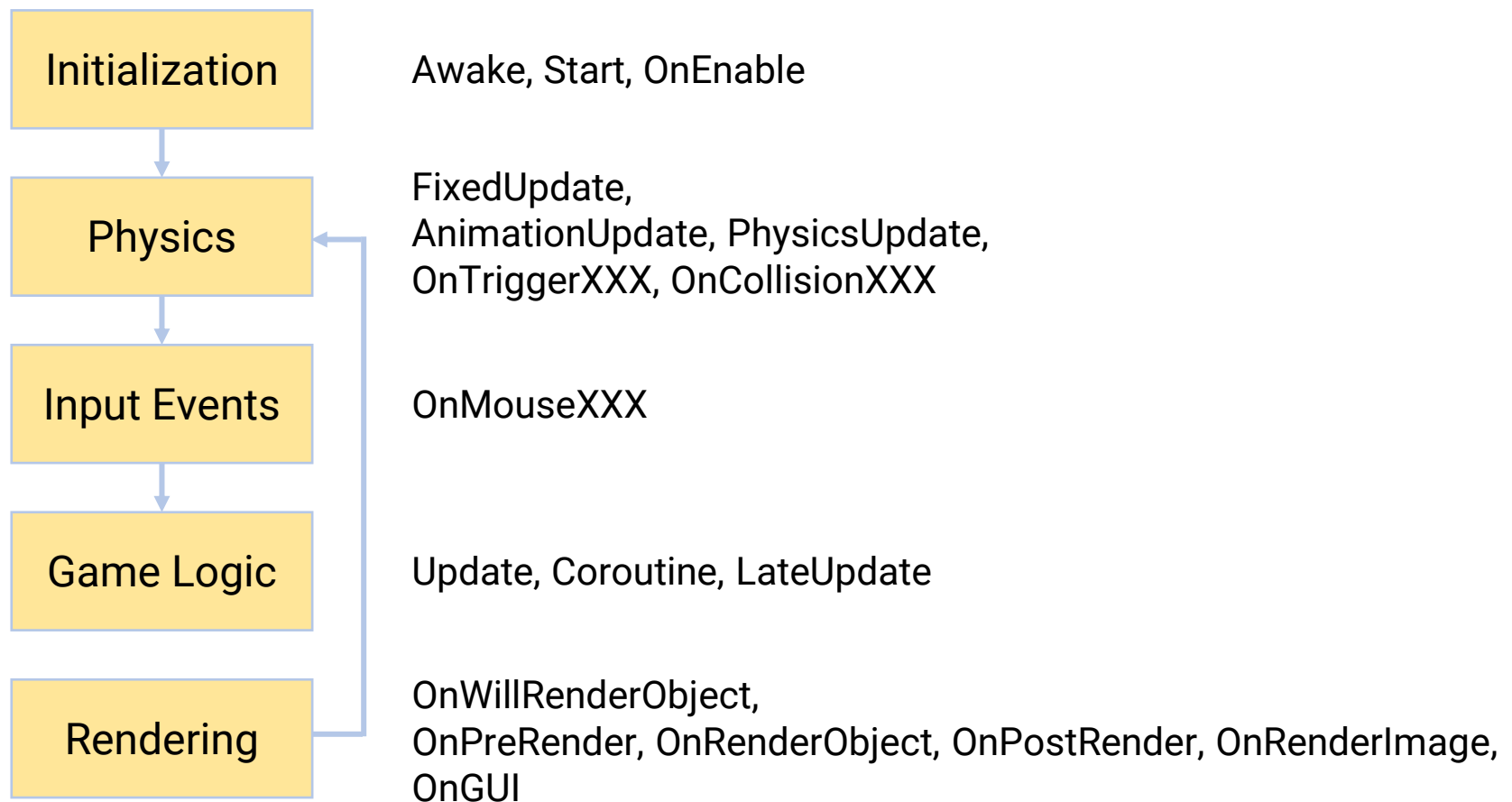
- The most widely used game engine (especially for mobile games) today
- Easier to jump in



Unity Overview (cont.)

- Unity event list order:

<https://docs.unity3d.com/Manual/ExecutionOrder.html>

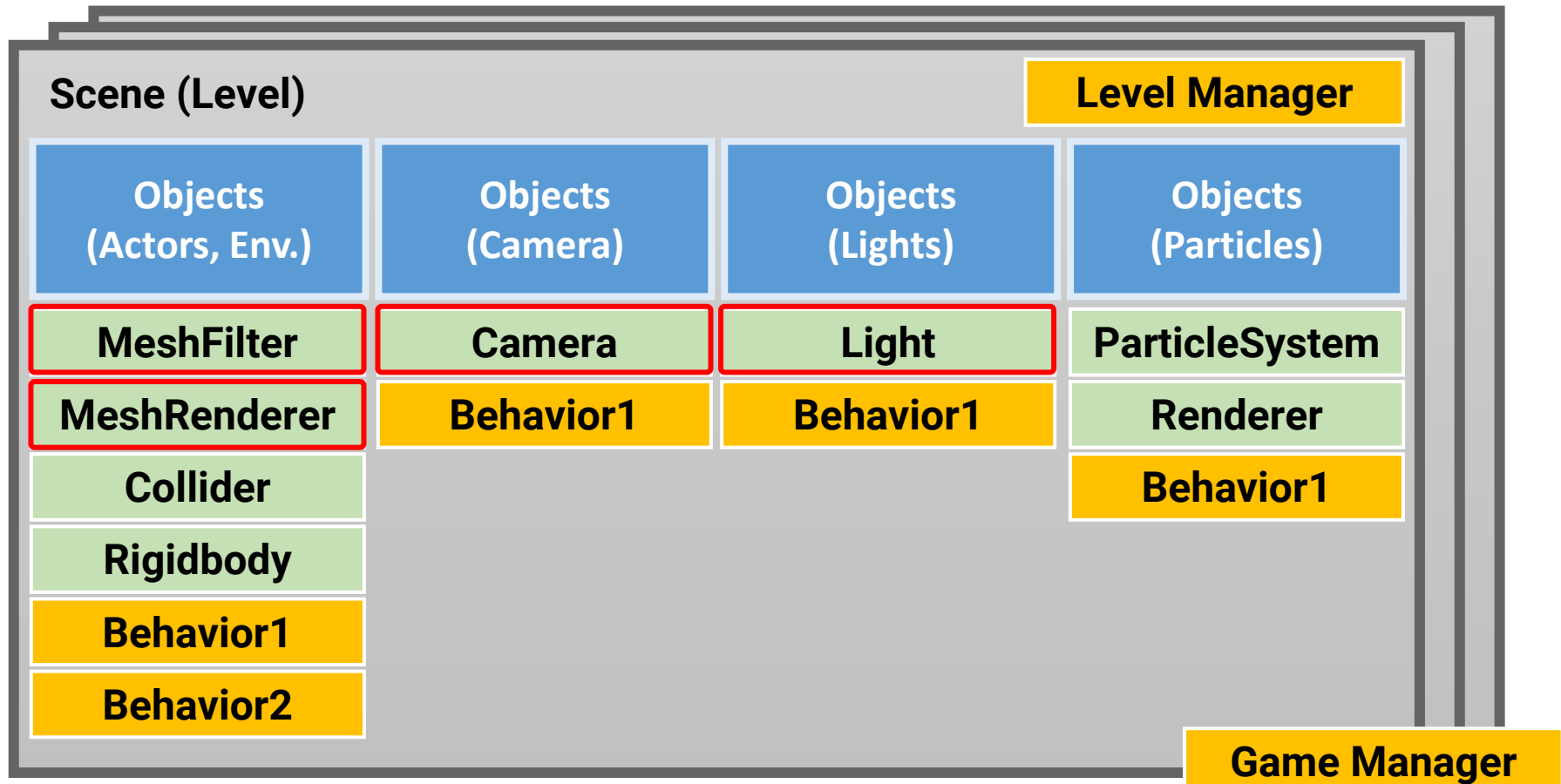


Unity Overview (cont.)

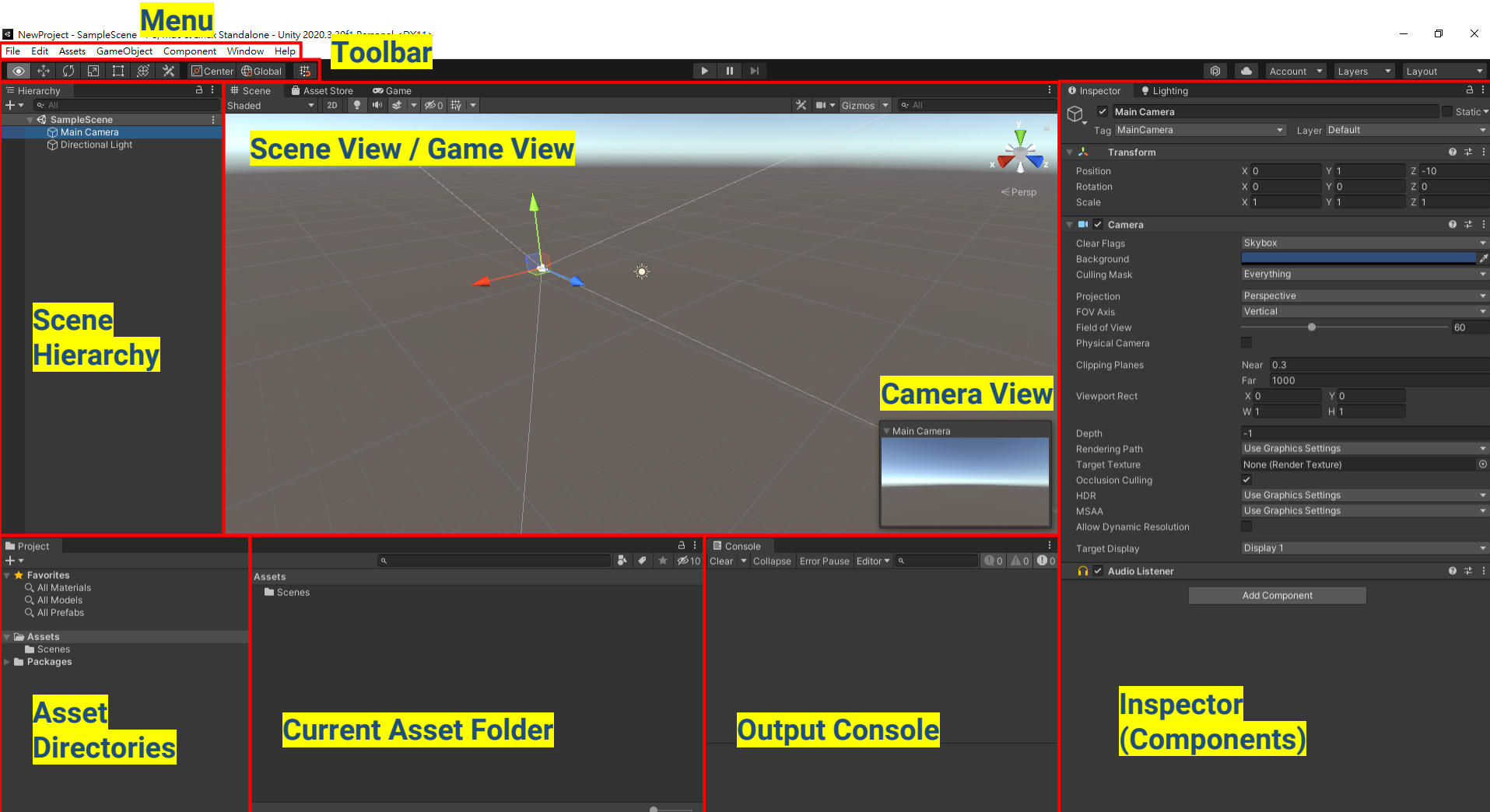
- Component-based (**C# scripts**)

Custom

Built-in



Unity Editor



Unity Editor

AllFrequenyLightingReconstruction - TestScene - PC, Mac & Linux Standalone - Unity 2020.3.30f1 <DX11>

The screenshot displays the Unity Editor interface. The central 3D view shows a fluffy, orange and white animal model (Arcanine) with a black mask. The Inspector panel on the right is highlighted with a red border and contains the following components:

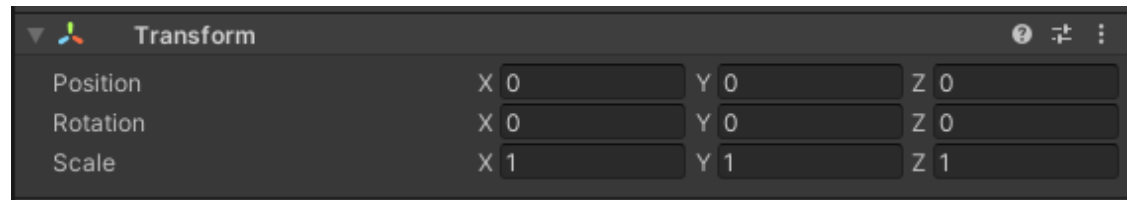
- Transform**: Position (X: 0, Y: 0, Z: 0), Rotation (X: 0, Y: 0, Z: 0), Scale (X: 1, Y: 1, Z: 1).
- Default (Mesh Filter)**: Mesh (default).
- Mesh Renderer**: Materials (Element 0: Material_#35, Element 1: Material_#36, Element 2: Material_#37, Element 3: Material_#38).
- Lighting**: Cast Shadows (On), Receive Shadows (checked), Contribute Global Illumination (checked), Receive Global Illumination (Light Probes).
- Additional Settings**: Material_#37 (Material), Material_#36 (Material), Material_#38 (Material), Material_#35 (Material).

The Project panel at the bottom left shows the following hierarchy:

- Assets > Models
 - Arcanine
 - Buddha
 - Bunny
 - Dragon

Geometry Data in Unity

- Geometry data in Object Space is described in a **MeshFilter** component
 - **Mesh**
 - vertexBufferTarget / indexBufferTarget
 - vertices (position) / normals / uv(12345678) / tangents
 - triangles (indices)
 - subMeshCount
 - ...
- An object is placed in the virtual world by a **World Transform**, described by
 - Position (translation)
 - Rotation
 - Scale



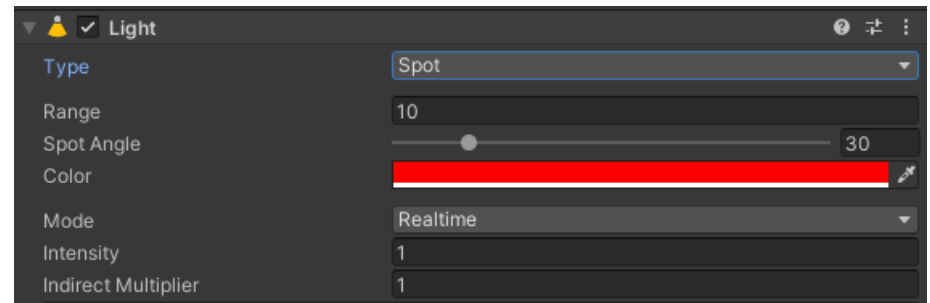
Camera in Unity

- An object that attaches a Camera component will become a camera



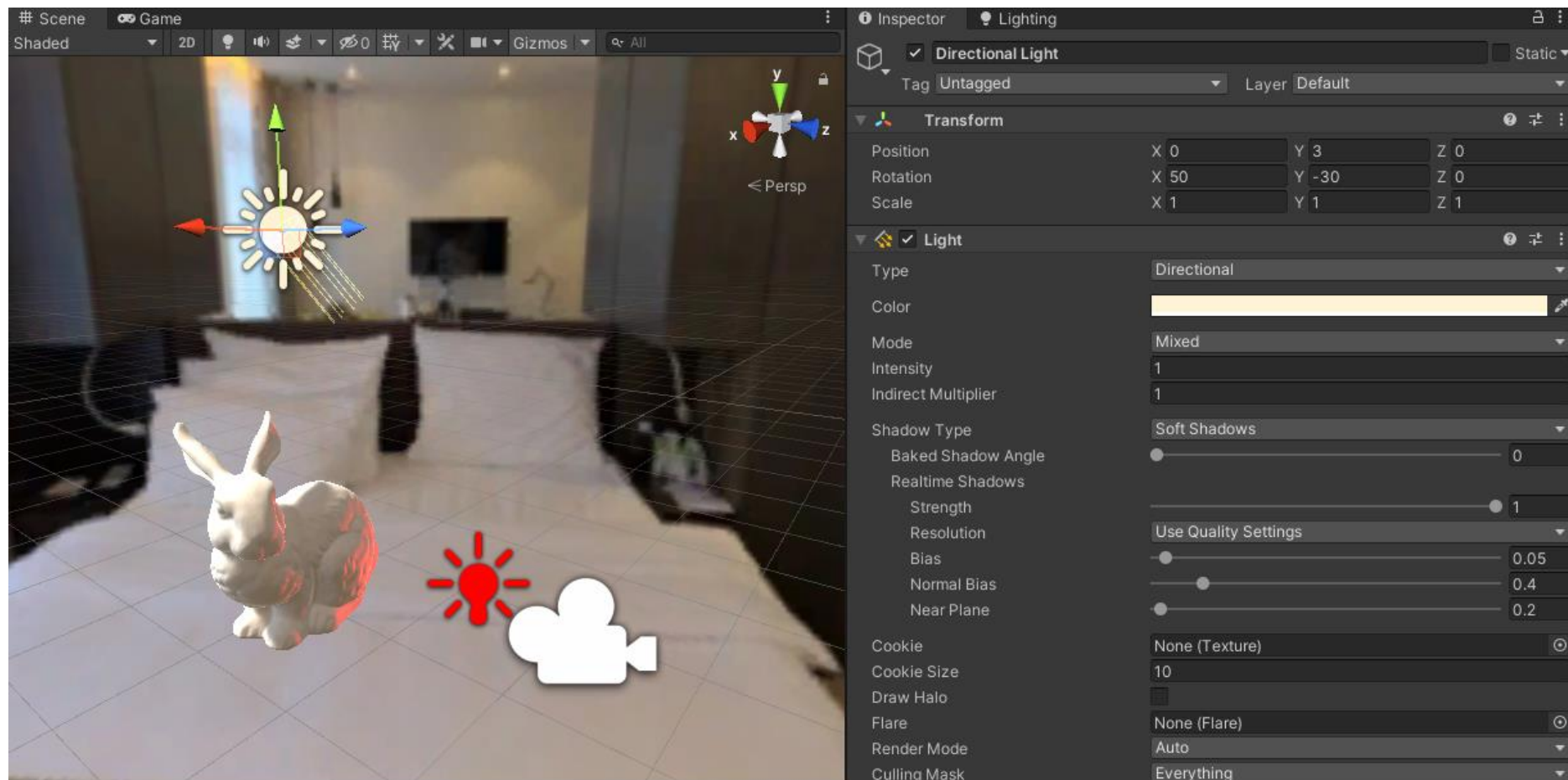
Lights in Unity

- An object that attaches a light component will become a light
- Unity supports several types of lights
 - Directional light
 - Point light
 - Spot light
 - Area light (bake only)
 - Environment light
(using spherical harmonics)



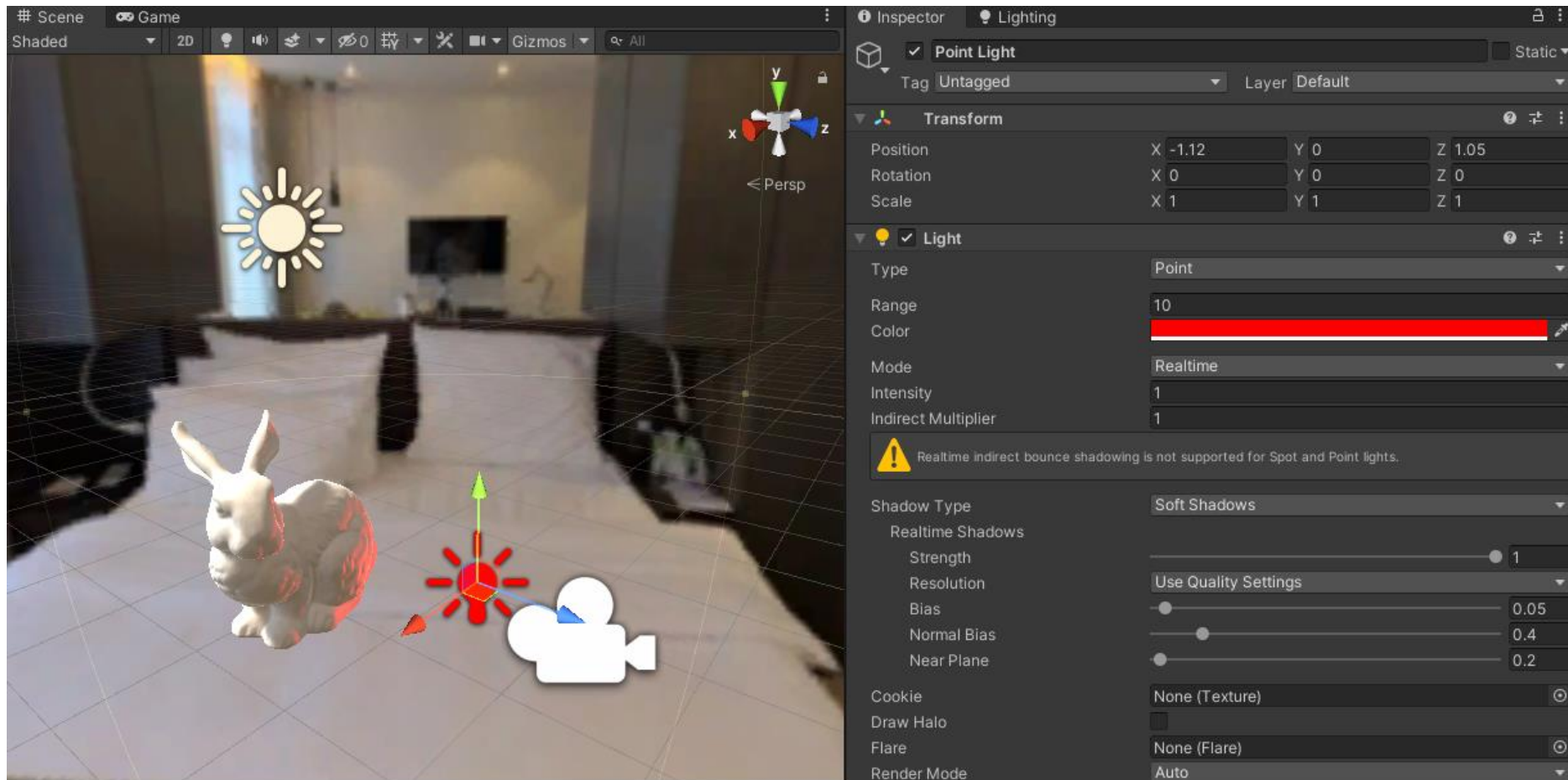
Lights in Unity (cont.)

- Directional light



Lights in Unity (cont.)

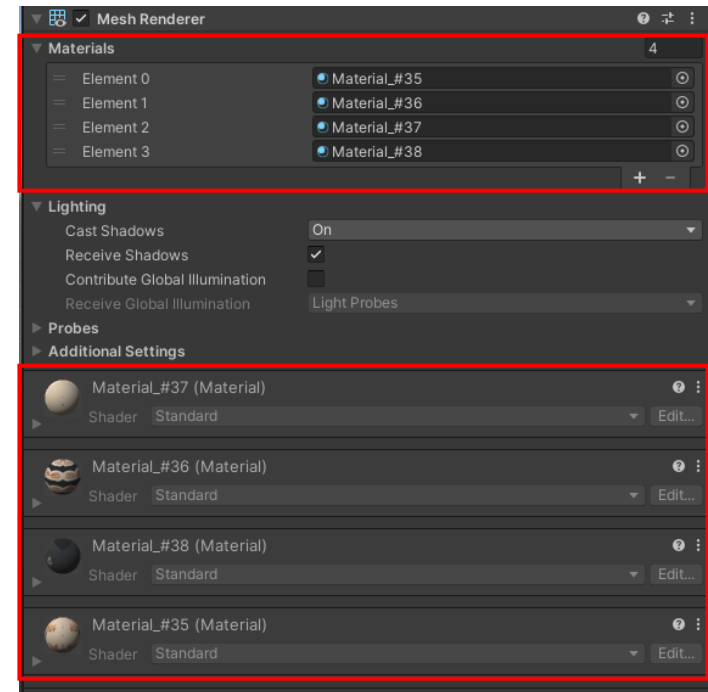
- Point light



Unity MeshRenderer

- Rendering features are described in a **(Mesh)Renderer** component
 - **Materials**
 - The material of each subMesh
 - **Lighting**
 - Does the object cast/receive shadows?
 - **Probe**
 - Does the object shade with light probes
(e.g., reflection cubemaps)

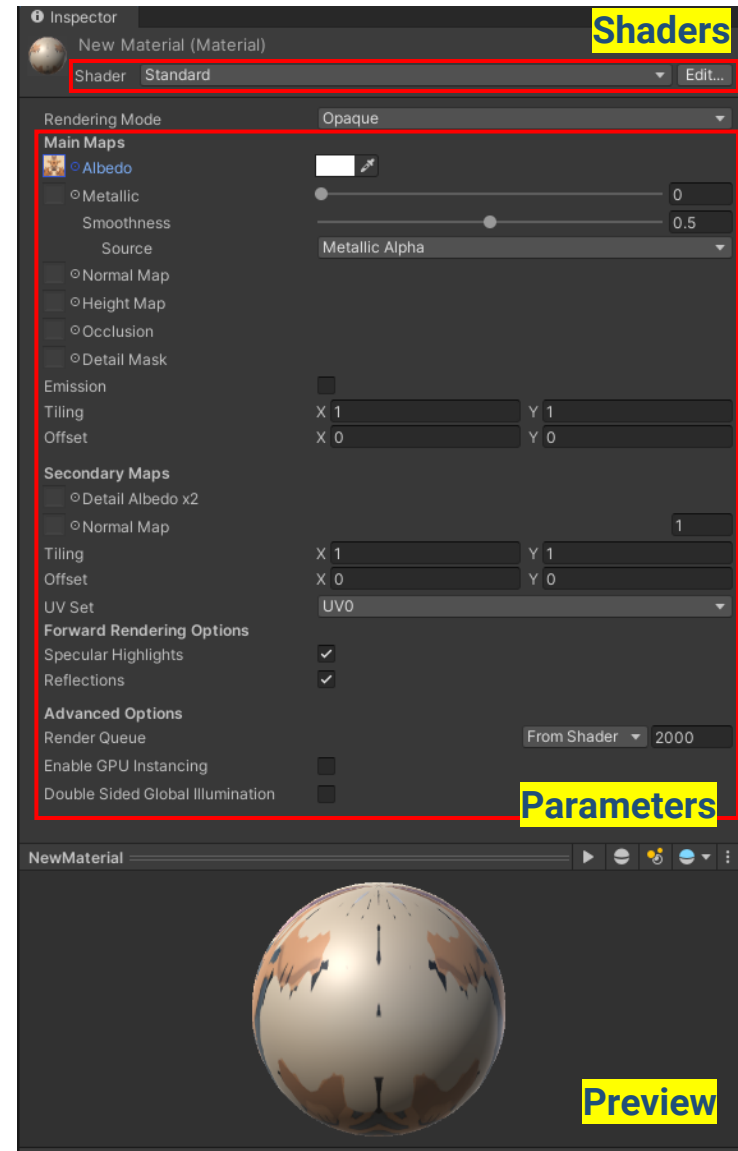
four subMeshes



four materials

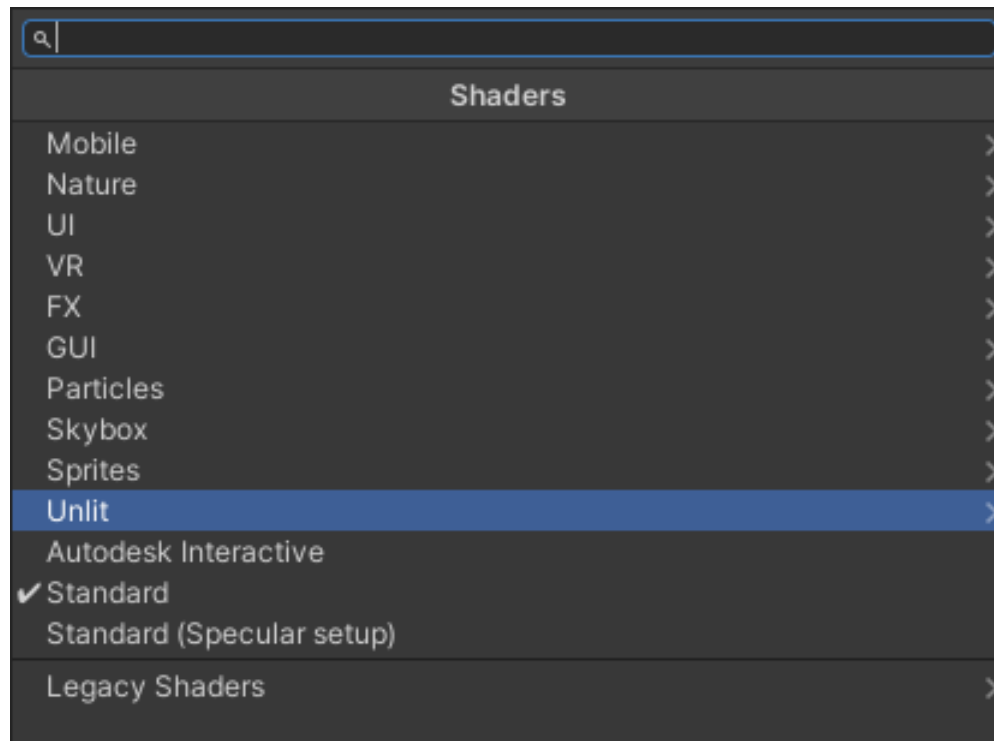
Unity Material

- **Material = Shader + Parameters**
 - A Unity shader file comprises at least a vertex shader and a fragment shader, and may include a geometry shader or tessellation shader
 - **Shader** defines the way (e.g., math) to transform objects and compute surface color
 - **Shader** also defines a set of parameters



Unity Built-in Shaders

- Unity provides a bunch of built-in shaders
- Developers can also create their own shaders by writing shader code (NVIDIA Cg)



Unity Built-in Shaders (cont.)

- You can download the built-in shaders for reference <https://unity.com/releases/editor/archive>

Unity download archive

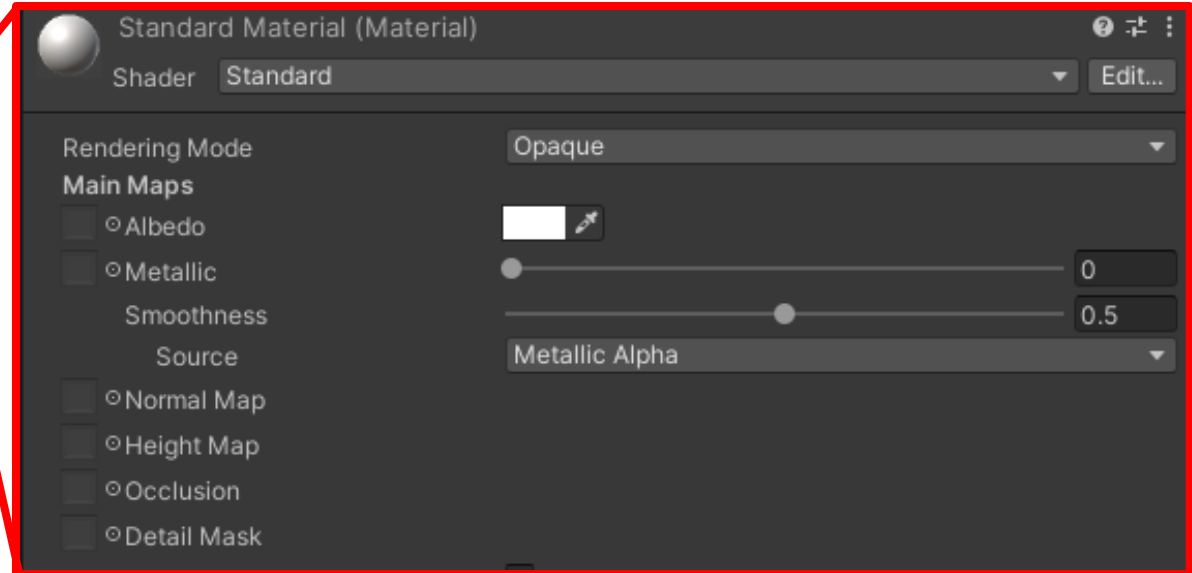
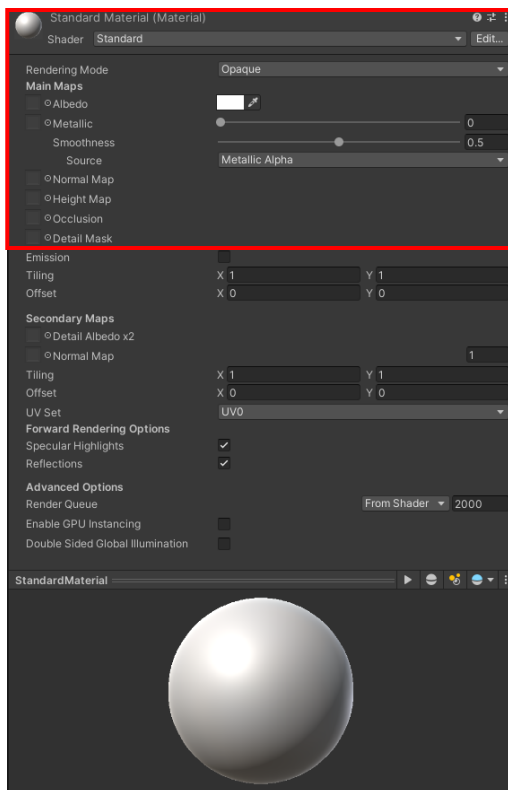
From this page you can download the previous versions of Unity for both Unity Personal and Pro (if you have a Pro license, enter in your key when prompted after installation). Please note that we don't support downgrading a project to an older editor version. However, you can import projects into a new editor version. We advise you to back up your project before converting and check the console log for any errors or warnings after importing.

Unity 2022.X	Unity 2021.X	Unity 2020.X	Unity 2019.X	Unity 2018.X	Unity 2017.X	Unity 5.X
Unity 2022.2.1 December 12, 2022	Unity Hub	Downloads (Win) ^	Downloads (Mac) v	Downloads (Linux) v	Release Notes	
Unity 2022.2.0 December 7, 2022	Unity Hub	Unity Installer Unity Editor 64-bit Built in shaders Unity Accelerator	Downloads (Mac) v	Downloads (Linux) v	Release Notes	
Unity 2022.1.24 ad_unity/4fead5835099/built_in_shaders-2022.2.1f1.zip	Unity Hub	Torrent download	Downloads (Mac) v	Downloads (Linux) v	Release Notes	

Unity Built-in Shaders (cont.)

- **Standard shader**

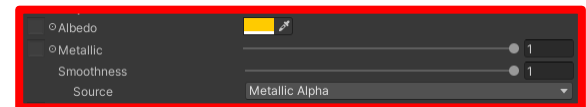
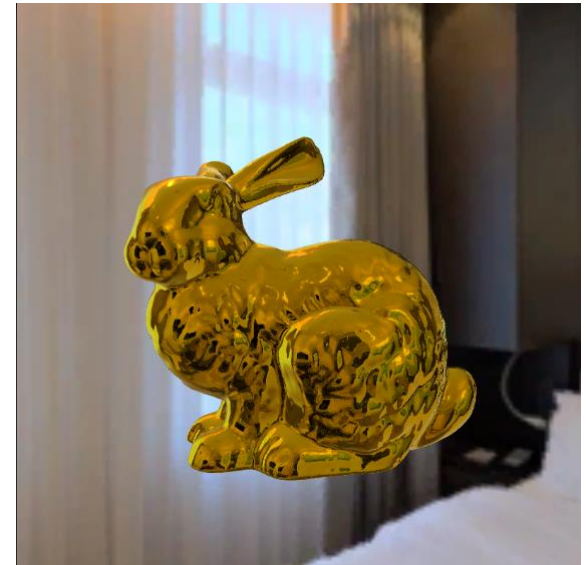
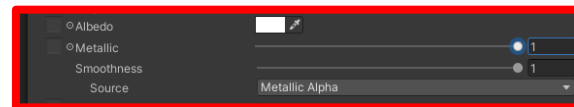
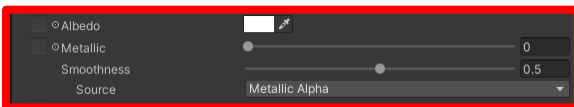
- You can use the Unity standard shader for most 3D objects
 - A variant of Disney's BRDF model



Unity Built-in Shaders (cont.)

- **Standard shader**

- You can use the Unity standard shader for most 3D objects
 - A variant of Disney's BRDF model



Design of Unity's Rendering System

- How does Unity handle the **arbitrary number** and **various types** of **lights**?
 - By multiple rendering passes

directional light
ZWrite
ZTest LEqual
Blend
SrcAlpha DstAlpha

```
// -----
// Base forward pass (directional light, emission, lightmaps, ...)
Pass
{
    Name "FORWARD"
    Tags { "LightMode" = "ForwardBase" }

    Blend [_SrcBlend] [_DstBlend]
    ZWrite [_ZWrite]

    CGPROGRAM
    #pragma target 3.0

    // -----

    #pragma shader_feature_local _NORMALMAP
    #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALP
    #pragma shader_feature_fragment _EMISSION
    #pragma shader_feature_local _METALLICGLOSSMAP
    #pragma shader_feature_local_fragment _DETAIL_MULX2
    #pragma shader_feature_local_fragment _SMOOTHNESS_TEXTURE_ALBEDO
    #pragma shader_feature_local_fragment _SPECULARHIGHLIGHTS_OFF
    #pragma shader_feature_local_fragment _GLOSSYREFLECTIONS_OFF
    #pragma shader_feature_local _PARALLAXMAP

    #pragma multi_compile_fwdbase
    #pragma multi_compile_fog
    #pragma multi_compile_instancing
    // Uncomment the following line to enable dithering LOD crossfad
    // #pragma multi_compile _ LOD_FADE_CROSSFADE

    #pragma vertex vertBase
    #pragma fragment fragBase
    #include "UnityStandardCoreForward.cginc"

    ENDCG
}
```

```
// -----
// Additive forward pass (one light per pass)
Pass
{
    Name "FORWARD_DELTA"
    Tags { "LightMode" = "ForwardAdd" }

    Blend [_SrcBlend] One
    Fog { Color (0,0,0,0) } // in additive pass fog should be black
    ZWrite Off
    ZTest LEqual

    CGPROGRAM
    #pragma target 3.0

    // -----

    #pragma shader_feature_local _NORMALMAP
    #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHA
    #pragma shader_feature_local _METALLICGLOSSMAP
    #pragma shader_feature_local_fragment _SMOOTHNESS_TEXTURE_ALBEDO_C
    #pragma shader_feature_local_fragment _SPECULARHIGHLIGHTS_OFF
    #pragma shader_feature_local_fragment _DETAIL_MULX2
    #pragma shader_feature_local _PARALLAXMAP

    #pragma multi_compile_fwdadd_fullshadows
    #pragma multi_compile_fog
    // Uncomment the following line to enable dithering LOD crossfade.
    // #pragma multi_compile _ LOD_FADE_CROSSFADE

    #pragma vertex vertAdd
    #pragma fragment fragAdd
    #include "UnityStandardCoreForward.cginc"

    ENDCG
}
```

per point (spot) light
ZWrite Off
ZTest LEqual
Blend
SrcAlpha One

If there are too many lights, the less important ones will be rendered by vertex lighting

Design of Unity's Rendering System (cont.)

- How does Unity handle **various materials**? For example, with or without an **Albedo** texture

```
Shader "Standard"  
{  
    Properties  
    {  
        _Color("Color", Color) = (1,1,1,1)  
        _MainTex("Albedo", 2D) = "white" {}    }  
}
```

**For materials that do not use an albedo texture
Unity will create a pure white one,
so the shader code can be unified**

Design of Unity's Rendering System (cont.)

- How does Unity handle **transparency**?
 - By defining RenderQueue
 - Background (1000)
 - Geometry (2000)
 - AlphaTest (2450)
 - Transparent (3000)
 - Overlay

