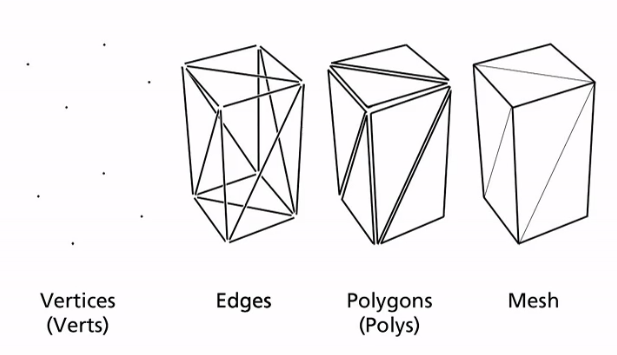
**Mesh** 多边形网格

Unity游戏场景中的各种物体都有由各种网格构成。3D网格或多边形网格，Meshes[ˈmeʃɪz]又是由点（points）和 线（lines）构成，*polygonal* meshes [pɒˈlɪgənl]表示立体模型的一种方法，polygon [ˈpɒlɪɡən]（多边形）由Vertex [ˈvɜːteks]（顶点）、Edge [edʒ]（线）构成。三角形是最简单的多边形，可以简化渲染过程。复杂的Mashes只能由其他的网格建模工具创建。



**Mesh Renderer** [ˈrɛndərə]网格渲染器

渲染一个网格时，需要两个组件，Mesh Filter [ˈfɪltə(r)]（网格过滤器）和Mesh Renderer（网格渲染器），当从外部导入3d模型时，Unity会自动创建prefab [ˈpriːfæb]，其中包括这两个组件，Mesh Filter存储了模型的网格数据，Mesh Renderer负责在场景中渲染网格数据，但不渲染Skin Meshes(皮肤网格)，皮肤网格用于包裹角色，角色由骨骼动画控制，骨骼的任意变化都会影响网格数据，如果导入的模型有皮肤，Unity会为止创建皮肤网格渲染器。如果一个模型是由多个独立的网格组成，那么每个网格都会有其prefab，Mesh Renderer的属性具体决定了如何在场景中绘制模型，这些属性包括阴影、材质、光探针（Light Probes）3类，阴影方面，你可以设置模型是否投射和接收阴影，Materials [məˈtɪərɪəlz]材质方面，可以使用多个材质来渲染此模型，网格渲染器至少包含一个材质，最后是光探针，网格可以被光探针照亮，光探针在一个点上模拟灯光并投射到网格上，这常用于顶替灯光映射（light mapping），像游戏角色这样的动态物体，无法被烘培光照亮，就使用光探针，勾选Use Light Probes选项，

The Mesh Renderer, to render a mesh, we need two components. The mesh Filter and mesh Renderer. When import Mesh Model.

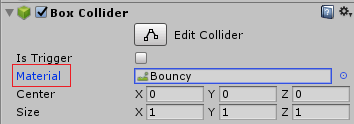
**Material** 材质，通过材质来改变物体的外观，可以把Texture[ˈtekstʃə(r)] 贴图作为一个物体的材质。

Unity中的材质是控制游戏中物体的外观的资产asset ，材质被赋予渲染组件，材质把shader [ˈʃeɪdə] （绘制物体表面的脚本）和颜色、纹理等信息联系起来了。当你从一些建模软件（3DMax等）导入3D资产时，Unity会在Assets文件夹Materials子文件夹中自动创建材质，也会自动把纹理贴图赋给相应的物体，Unitry自带了很多shader以满足日常游戏开发所需，你也可以编写自己的shader。

**Texture** 贴图，通常是图片的各种格式

**prefab**

**Box Collider** 将GameObject的材料设置为Bouncy那么这个物体就会像皮球一样弹来弹去



**bouncy**[ˈbaʊnsi] adj.快活的; 有弹性的; 精神的; 自大的 She was *bouncy* and full of *energy.*她朝气蓬勃，充满活力。

**Lights**

灯光分为动态光Dynimic和烘培光Bake，动态光在游戏运行中是实时计算的，烘培光是在线下（off-line）计算的并保存成纹理（Texture）的，这些灯光纹理被贴到烘培的物体上以提升性能。

灯光有4种类型，Range属性决定了光线照射的距离远近，只对点光和聚光灯有意义，Color控制灯光的颜色，Intensity控制灯光的强弱，高级属性包括阴影（Shadows）、光晕（Halos）、耀斑（Flare）和Cookie（是一个虚拟遮罩或者标记，挡在光源前面来产生某种阴影）

point点光

spot聚光灯 ，像手电筒或者汽车灯头，Spot Angle决定了聚光灯的圆锥体照射范围的角度大小

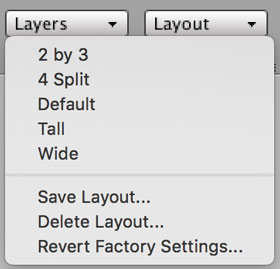
directional直线光，平行光，和太阳一样，所以位置属性对平行光没有意义

area（baked only）区域光

2种光源，环境光（Ambient Light）和发光材质（Emissive Material）

环境光控制场景中的无方向全家光

**layers** 英 ['leɪəs] n. 层，层次( layer的名词复数 ); 这里翻译为布局。



**Velocity** [vəˈlɒsəti] n.速度

**Description** 描述

Base class for all entities in Unity scenes.

Note: Many variables in the GameObject class have been removed. To access, for example GameObject.renderer in csharp use GetComponent<Renderer>() instead. In JS script use GetComponent.<Renderer>().

See Also: Component.

Unity场景中所有实体的基类。

注意：GameObject类中的许多变量已被删除。例如，要访问csharp中的GameObject.renderer，请改用GetComponent <Renderer>（）。在JS脚本中使用GetComponent。<Renderer>（）。

另请参见：组件。

**Properties** 属性

**Constructors** 构造函数

**Public Methods** 公共方法

**Static Methods** 静态方法

**Inherited Members** 继承的成员