

Unity引擎基础

FUNDAMENTAL

Unity3D介绍

工具介绍

基础概念

Material

Camera

InstantOC

光照系统

声音

祁天暄：1986086334

学习注意事项

- 每天知识点必须理解（定义、作用、适用性、操作/语法）。
- 每天练习必须会做（独立完成）。
- 形成笔记保留下来。
- 课上注意听！课下抓紧练、不懂马上问。

Unity3D介绍

Unity3D介绍

简介

Unity3D

游戏

虚拟现实

游戏引擎

Unity特点

著名游戏公司的加入

历史

游戏公司大致分工

课程体系

Unity引擎基础

Unity高级应用

游戏架构设计

完整项目实战

简介



Unity 3D

- 由 Unity Technologies 公司开发的用于轻松创建游戏和三维互动内容的开发工具，是一个国际领先的专业游戏引擎。



- 在移动平台，Unity几乎成为3D游戏开发的标准工具，使用它开发的游戏数不胜数。
- 2D游戏：视角完全锁定，二维坐标。
- 3D游戏：视角可以随意变化，三维坐标。



虚拟现实



虚拟现实(续1)

- Virtual Reality

当今世界前沿科技之一。

利用电脑模拟产生一个三维空间的虚拟世界，并提供视觉、听觉、触觉等感官的模拟。使用者通过各种输入设备与虚拟环境中的事物进行交互，从而产生身临其境的体验。

虚拟现实(续2)

- VR设备：移动端头显设备

只需要将手机放入即可，使用方便、价格低廉。但受硬件性能制约，用户体验感不足。



虚拟现实(续2)

- VR设备：外接式头戴设备

需要连接至PC，具有更强劲的性能，更良好的体验，可以捕捉肢体动作。但价格过高，需要额外硬件的支持。



虚拟现实(续2)

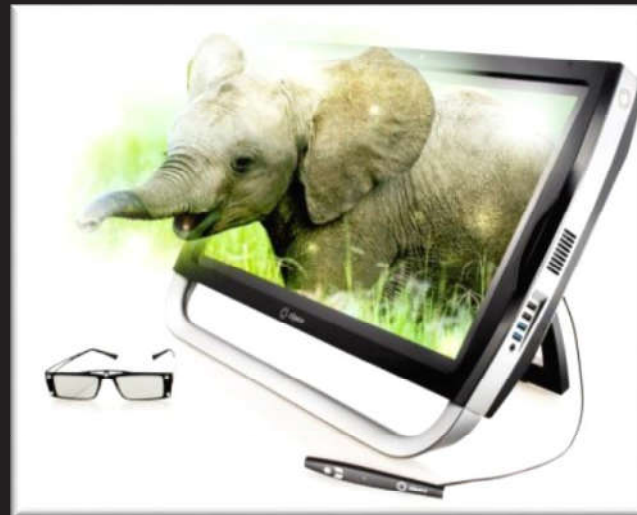
- VR设备：一体式头戴设备

具备独立的运算、输入输出功能，不受空间和其他设备（PC、手机）限制。但目前还不成熟，产品偏少。



虚拟现实(续2)

- VR其他设备：Xbox One + Kinect、zSpace



虚拟现实(续3)

- VR应用：

军工

科研

游戏

视频

教育

旅游

家装

虚拟试驾

房地产项目展示

.....

虚拟现实(续4)

- 增强现实 **A**ugmented **R**eality

通过电脑技术，将虚拟的信息应用到真实世界，真实的环境和虚拟的物体实时地叠加到了同一个画面或空间同时存在。



虚拟现实(续4)

- AR 与 VR 区别

VR 看到的场景和人物全是假的，是把你的意识代入一个虚拟的世界。AR 看到的场景和人物，一部分是真一部分是假，是把虚拟的信息带入到现实世界中。



游戏引擎

- 程序的框架，一款游戏最最核心的代码。
- 包含以下系统：渲染引擎、物理引擎、碰撞检测系统、音效、脚本引擎、动画系统、人工智能、网络引擎、以及场景管理。
- 使用游戏引擎，开发者可以重用已有的核心技术，将精力集中在游戏逻辑和设计上，从而简单快速的创建游戏。

Unity3D特点

- 简单易用
- 开发效率高
- 价格便宜
- 新手居多
- 23个平台间自由迁移，出色的部署，完全的覆盖



发布平台包括：IOS、Android、Windows Phone、Windows、Web、微软Xbox360、索尼PS3、任天堂Wii等。



全球游戏引擎市场份额



著名游戏企业的加入

- 2014年10月，Unity公司创始人兼CEO 宣布转任执行副总裁，接替他的人选是曾任EA公司CEO的约翰·里奇蒂耶洛(John Riccitiello)。



著名游戏企业的加入(续1)

- 2015年5月，Unity CEO宣布，前Epic Games总裁，迈克·卡普斯(Mike Capps) 将会正式加入Unity。



使用Unity3D开发的游戏

- 腾讯游戏 — 全民破坏神、我叫MT2
- 搜狐畅游 — 天空八部3D、秦时明月2
- 完美世界 — 格斗宝贝、御天降魔传

.....



历史

- 2004年丹麦哥本哈根三位热爱游戏的年轻人决定开发一个易于使用、与众不同并且费用低廉的游戏引擎，帮助所有喜爱游戏的年轻人实现游戏创作梦想。
- 2005年发布了Unity1.0版本，并将总部设在美国旧金山。
- 2007年发布了Unity2.0版本--新增了地形引擎、实时动态阴影、网络多人联机功能等。
- 2009年发布了Unity2.5版本--推出Windows版本,支持IOS和Wii任天堂游戏机。
- 2010年发布了Unity3.0版本--支持Android平台。
- 2012年发布了Unity4.0版本--支持Web及其他游戏机平台。

历史(续1)

- 2012年Unity上海分公司成立，正式进军中国市场。
- 同年苹果公司一份报告，在App Store中55%的3D游戏都是使用Unity开发的，而Android的市场应该比苹果更大。
- 截止2013年6月，Unity Technologies公司在中国、加拿大、丹麦、英国、日本、韩国、立陶宛、瑞典等国家和地区都建立了相关机构，在全球拥有30个不同国家和地区的超过290名雇员。
- 2015年3月发布了Unity 5.0版本—包括了许多新功能及改进，如：实时全局光照、音频、动画系统等

游戏公司大致分工

- 策划
- 美工
- 软件工程师
- 测试工程师
- 客服等



课程体系



Unity引擎基础

- Unity 环境搭建
- C# 语言基础
- Unity API
- 3D数学基础
- 物理引擎1
- UGUI



Unity高级应用

- Unity高级组件
- IO，协程，多线程
- Unity高级组件应用
- 常用数据结构
- AR项目，VR项目
- NGUI-ARPG项目贯穿
- 美术资源性能优化
- Shader

游戏架构设计

- 面向对象
- 类库开发
- 设计模式
- 人工智能



完整项目实战

- 美术规范及优化
- XML 及 Json
- 数据库
- 项目制作



工具介绍

工具介绍

面板

Project 项目资源面板

Hierarchy 层次面板

Scene 场景面板

Game 游戏面板

Inspector 检视面板

工具条

变换工具

变换切换

播放控件

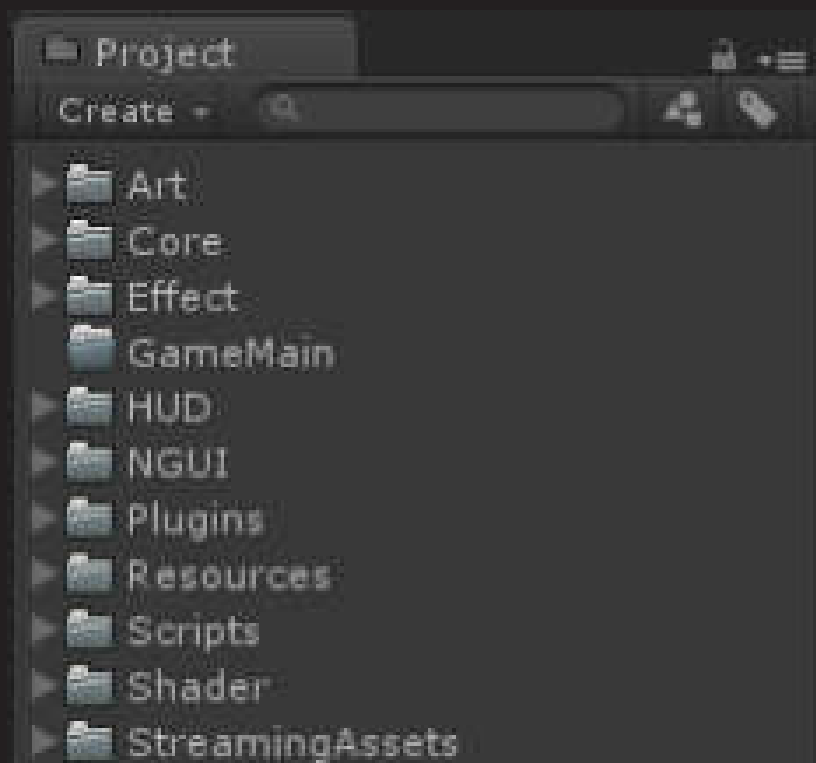
视图

面板



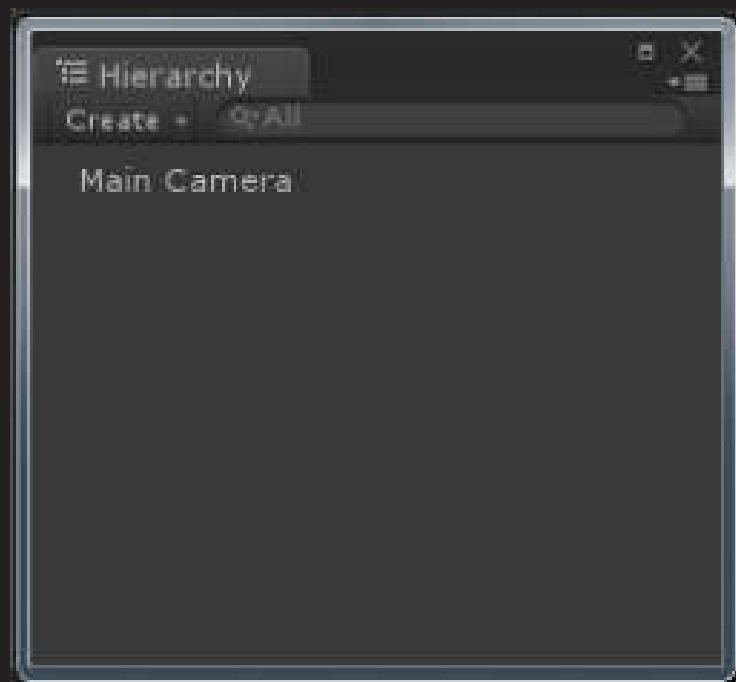
Project 项目资源面板

- 存放游戏的所有资源。
- 与项目中资源文件夹Assets对应，例如场景、脚本、模型、音频、图片等文件。




Hierarchy 层次面板

- 显示当前场景中所有游戏对象的层级关系。
- 包含了当前场景的游戏对象（GameObject），其中一些是资源文件的实例，如3D模型和其他预制组件的实例。

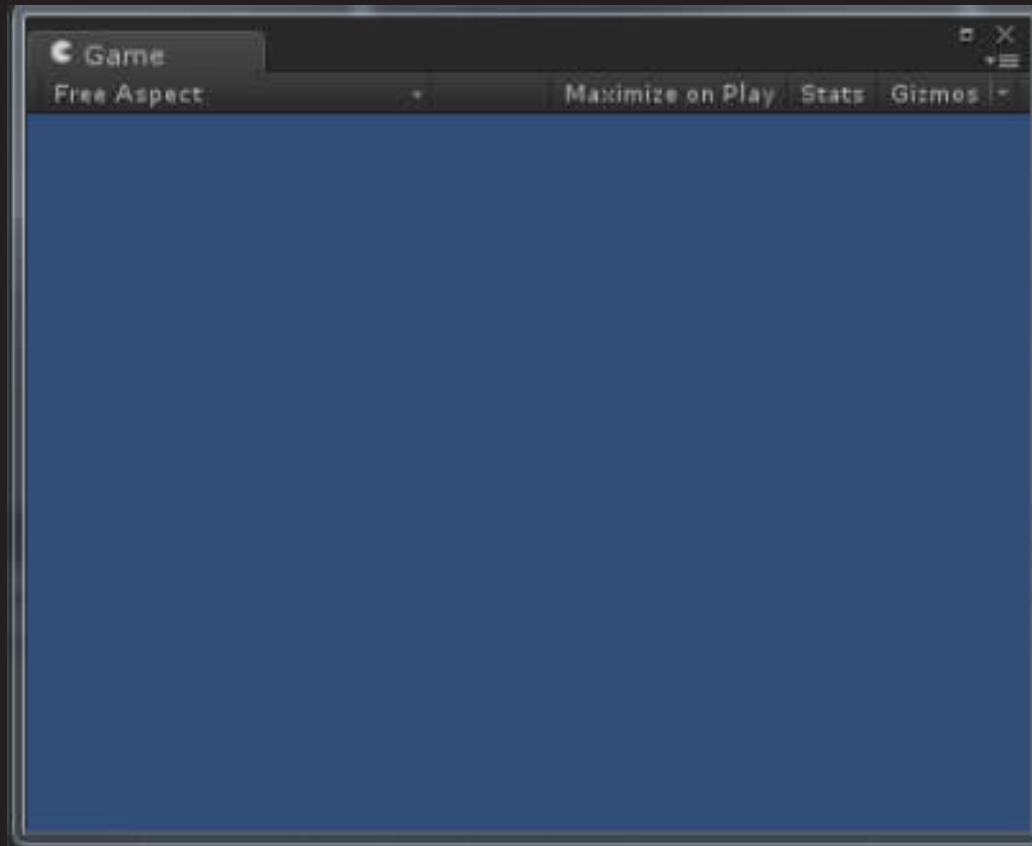


Scene 场景面板

- 提供设计游戏界面的可视化面板
- 常用快捷键
 1. 按下鼠标滚轮拖动场景，滑动滚轮缩放场景。
 2. 鼠标右键旋转场景，点击“”后，通过左键移动场景。
 3. 点击右键同时按下W/S/A/D/Q/E键可实现场景漫游。
 4. 在Scene面板选中物体后按F键，或在Hierarchy面板双击物体，可将物体设置为场景视图的中心。
 5. 按住alt键同时通过鼠标左键围绕某物体旋转场景，鼠标右键缩放场景。

Game 游戏面板

- 预览游戏运行后的界面



Inspector 检视面板

- 显示当前选定游戏对象附加的组件及其属性信息。
- 为重要游戏物体选择图标：



工具

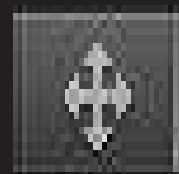


变换工具

- 移动场景 Q



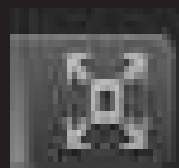
- 移动物体 W



- 旋转物体 E



- 缩放物体 R



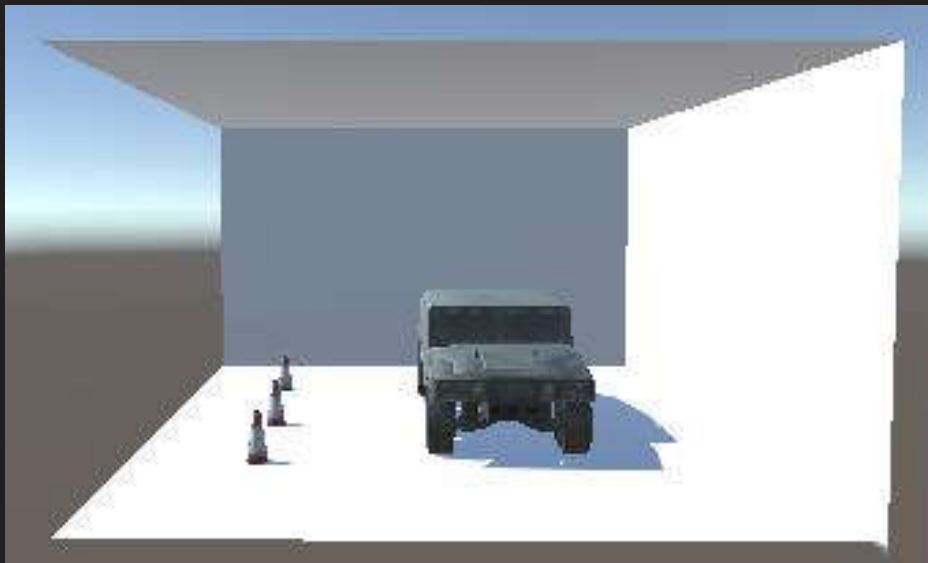
- 顶点吸附：选择物体按住 V 键，确定顶点后再拖拽到目标物体的某个顶点上。

备注：先松V键

练习：搭建车库

由4个Plane（平面），3个TrafficCone（交通圆锥体），1个humvee（悍马车）组成。

课堂练习



变换切换

- 左边是改变游戏对象的轴心点
Center：设置轴心点在物体中心
Pivot：使用物体本身的轴心
- 右边是改变物体的坐标，Global：世界坐标；Local：自身坐标



播放控件

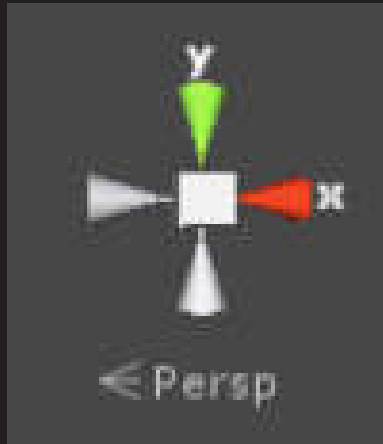
- 从左到右依次是预览游戏、暂停游戏、逐帧播放
查看游戏最终运行后的画面。在运行模式下，任何更改都只是暂时的，在退出运行时重置复位。



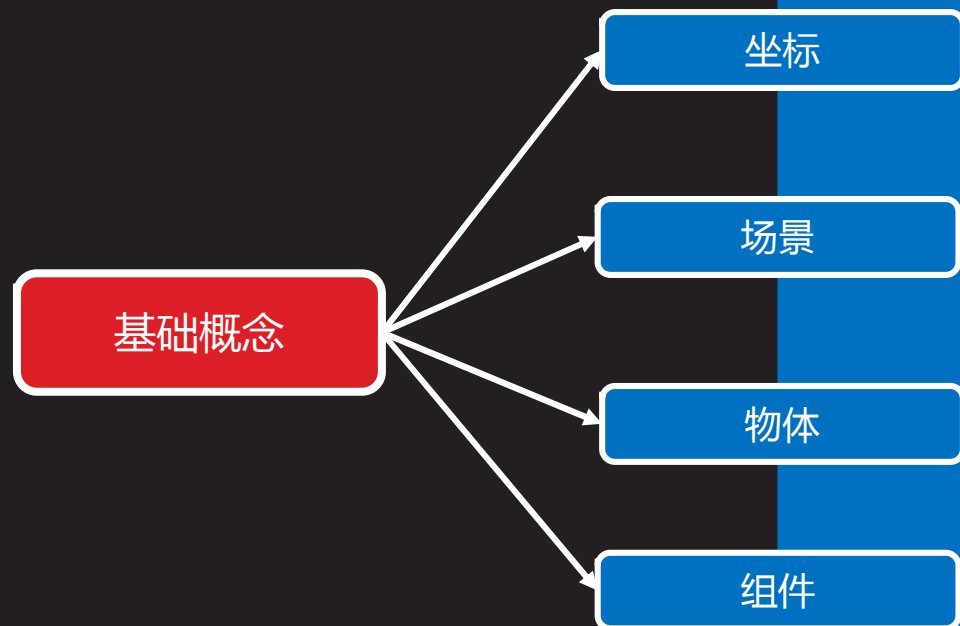
视图

- ISO：正交观察模式
- Persp：透视观察模式（近大远小）

视图角度：上下左右前后



基础概念



坐标

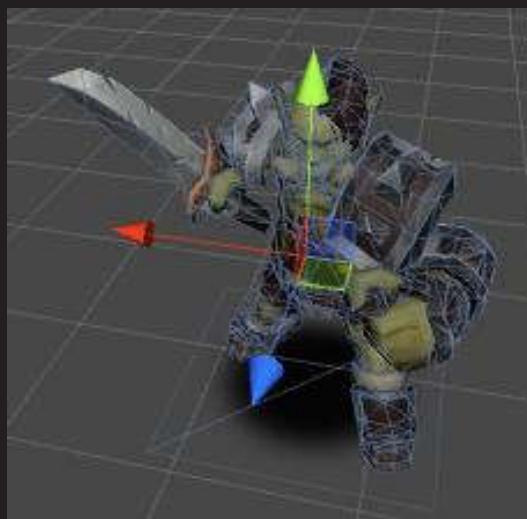


坐标

- 坐标：X红色、Y绿色、Z蓝色。

世界坐标：整个场景的固定坐标，不随物体旋转而改变。

本地坐标：物体自身坐标，随旋转而改变。



场景



场景 Scene

- 一组相关联的游戏对象的集合，通常游戏中每个关卡就是一个场景，用于展现当前关卡中的所有物体。



游戏对象



游戏对象 GameObject

- 运行时出现在场景中的游戏物体。

例如：人物、地形、树木

- 是一种容器，可以挂载组件。

- 父、子物体

在Hierarchy面板中，将一个物体拖拽到另外一个物体中。

子物体将继承父物体的移动，旋转和缩放属性，但子物体不影响父物体。

组件



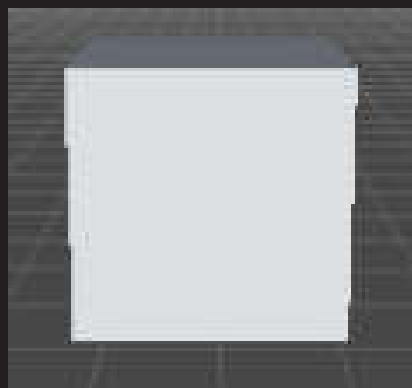
组件 Component

- 是游戏对象的功能模块。
- 每个组件都是一个类的实例。
- Transform 变换组件：决定物体位置、旋转、缩放比。
- Mesh Filter 网格过滤器：用于从资源中获取网格信息。
- Mesh Renderer 网格渲染器：从网格过滤器中获得几何形状，再根据变化组件定义的位置进行渲染。
- 网格过滤器 与 网格渲染器 联合使用，使模型显示到屏幕上。



创建立方体

- 1.创建空物体：GameObject—Create Empty
- 2.创建网格过滤器：Component—Mesh—Mesh Filter
设置Mesh属性
- 3.创建网格渲染器：Component—Mesh—Mesh
Renderer
设置材质Materials属性



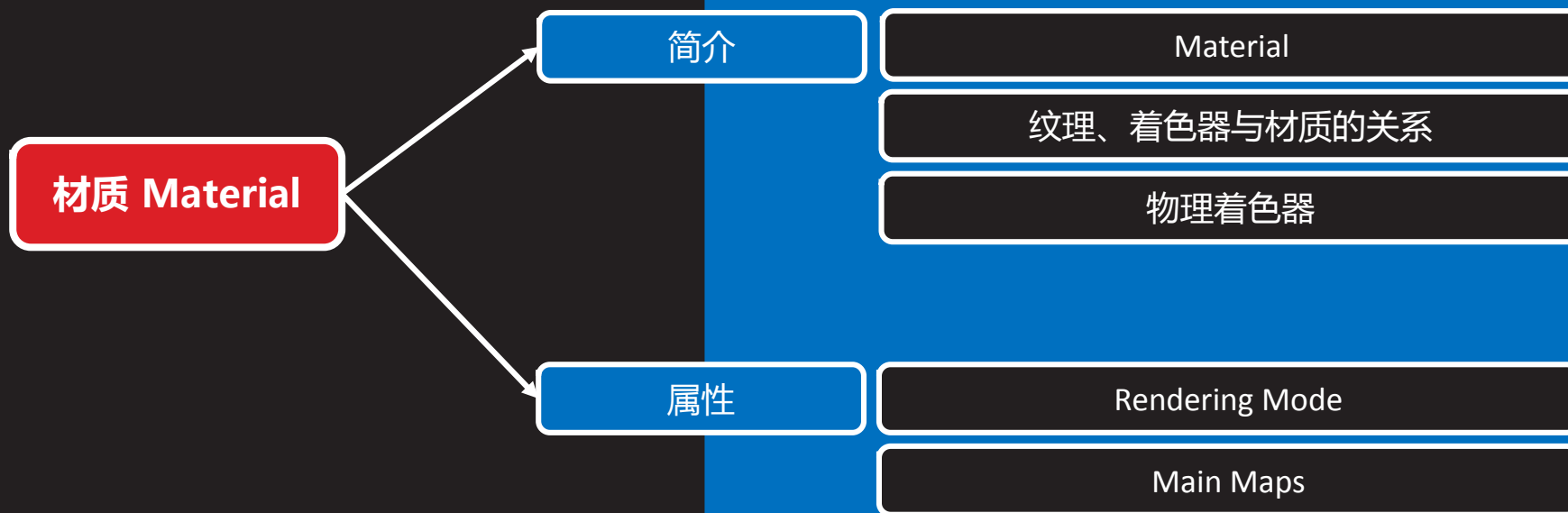
练习：创建子弹

由2个Capsule（胶囊）,1个Cylinder（圆柱体）组成。

课堂
练习



材质 Material



简介



Material

- 材质：物体的质地，指色彩、纹理、光滑度、透明度、反射率、折射率、发光度等。实际就是Shader的实例。
- Shader 着色器：专门用来渲染3D图形的技术，可以使纹理以某种方式展现。实际就是一段嵌入到渲染管线中的程序，可以控制GPU运算图像效果的算法。
- Texture 纹理：附加到物体表面的贴图。

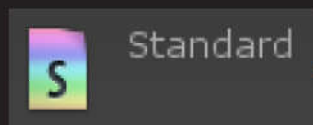


纹理、着色器与材质的关系

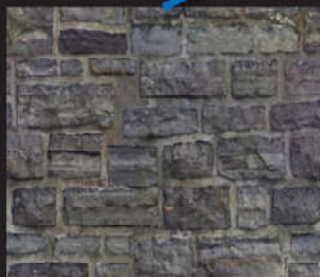
Material

MeshRenderer

Shader



Texture



Color



物理着色器

- 基于物理特性的Shader是Unity 5.x的重大革新之一，所谓物理着色器(Physically Based Shading,PBS)就是遵从物理学的能量守恒定律，可以创建出在不同光照环境下都接近真实的效果。



Unity 4.x

Unity 5.x

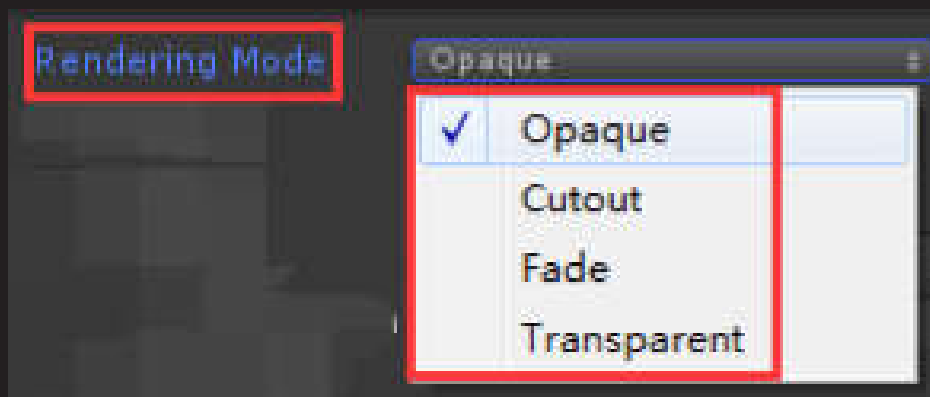


属性



Rendering Mode

- 渲染模式
 - Opaque 不透明，默认选项。
 - Transparent 透明，用于半透明和全透明物体，如玻璃。
 - Cutout 镂空，用于完全透明或完全不透明物体，如栅栏。
 - Fade 渐变，用于需要淡入淡出的物体。



Main Maps

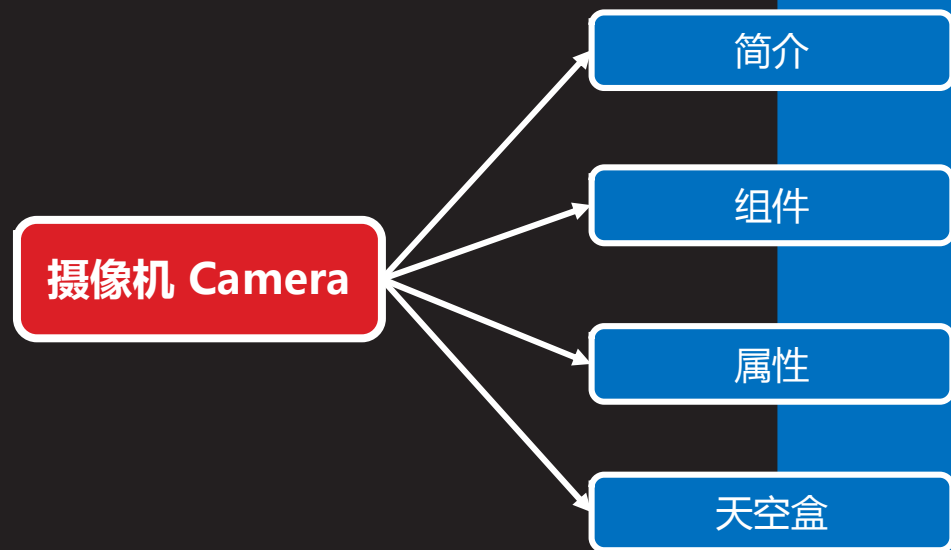
- Albedo 基础贴图：决定物体表面纹理与颜色。
- Metallic 金属：使用金属特性模拟外观。
- Specular 镜面反射：使用镜面特性模拟外观。
- Smoothness 光滑度：设置物体表面光滑程度
- Normal Map 法线贴图：描述物体表面凹凸程度。
- Emission 自发光：控制物体表面自发光颜色和贴图。
 - None 不影响环境
 - Realtime 实时动态改变
 - Backed 烘焙生效
- Tiling 平铺：沿着不同的轴，纹理平铺个数。
- Offset 偏移：滑动纹理。

物体轮廓

- 1. 创建材质 Outline
- 2. 指定着色器 BasicOutline
- 3. 指定纹理 Texture
- 4. 设置到物体中。



摄像机 Camera

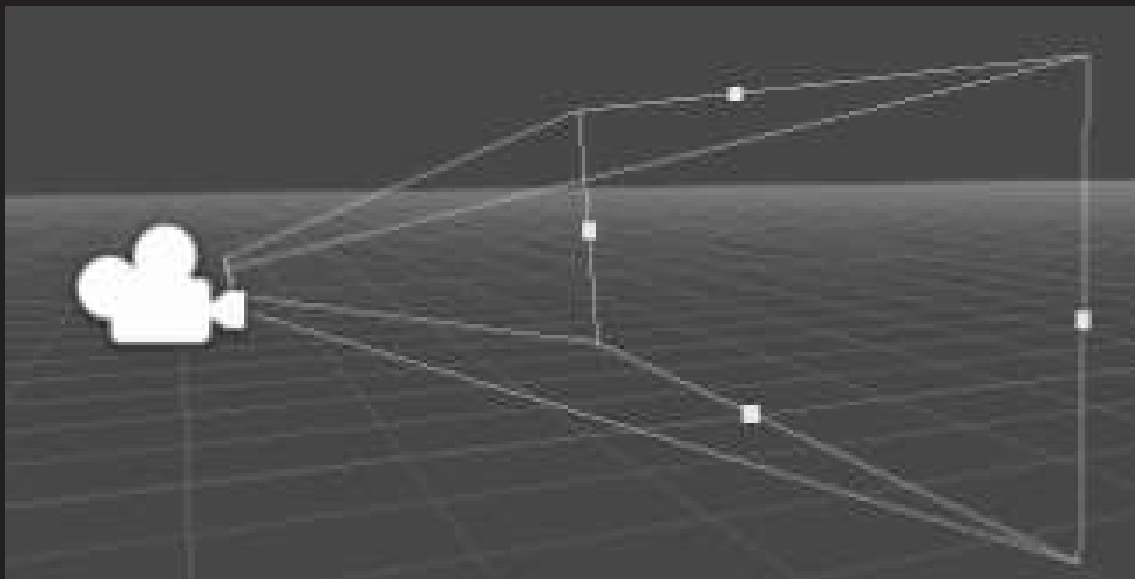


简介



简介

- 附加了摄像机Camera组件的游戏对象
- 向玩家捕获和显示世界的设备
- 场景中摄像机的数量不受限制

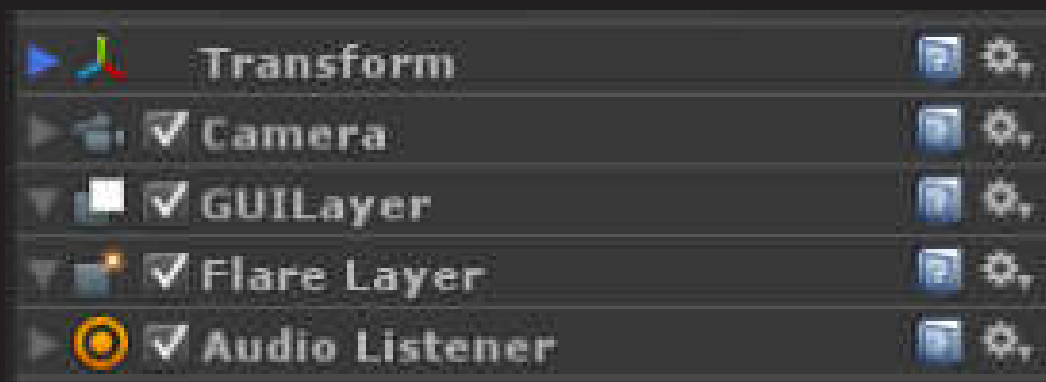


组件



组件

- Transform 变换组件
- Camera 摄像机：向玩家捕获和显示世界
- Flare Layer 耀斑层：激活可显示光源耀斑
- GUI Layer：激活可渲染二维GUI元素
- Audio Listener 音频监听器：接收场景输入的音频源Audio Source并通过计算机的扬声器播放声音。



属性



属性

- Clear Flags 清除标识：决定屏幕的空白部分如何处理

Skybox 天空盒：空白部分显示天空盒图案

Solid Color 纯色：空白部分显示背景颜色

Depth Only 仅深度：画中画效果时，小画面摄像机选择该项可清除屏幕空部分信息只保留物体颜色信息。

Don' t Clear 不清除：不清除任何颜色或深度缓存。

- Background 背景：所有元素绘制后，没有天空盒的情况下，剩余屏幕的颜色。
- Culling Mask 选择遮蔽层：选择要照射的层Layer。

属性(续1)

- Projection 投射方式：

Perspective 透视：透视图，物体具有近大远小效果。

Orthographic 正交：摄像机会均匀地渲染物体，没有透视感，通常小地图使用。

- Size 大小（正交模式）：摄影机视口的大小
- Field of view 视野（透视模式）：设置相机视野的远近距离
- Field of view 裁剪面：相机到开始和结束渲染的距离

Near 近：绘制的最近点。

Far 远：绘制的最远点。

属性(续2)

- Viewport Rect 视口矩形：标明这台相机视图将会在屏幕上绘制的屏幕坐标。

X:摄像机视图的开始水平位置。

Y:摄像机视图的开始垂直位置。

W宽度:摄像机输出在屏幕上的宽度。

H高度:摄像机输出在屏幕上的高度。

- Depth 深度：相机在渲染顺序上的位置。具有较低深度的摄像机将在较高深度的摄像机之前渲染。

天空盒



天空盒 SkyBox

- 围绕整个场景的包装器，用于模拟天空的材质。
- 天空盒材质种类：6 Sided，Procedural，Cubemap。



使用天空盒

- 设置摄像机 Clear Flags 属性为 Skybox。
- 方式一：摄像机添加组件 Skybox



- 方法二：光照窗口
Window – Lighting – Environment Lighting -- Skybox
可作为反射源将天空色彩反射到场景中物体。

6 Sided 属性

- Tint Color 色彩
- Exposure 亮度
- Rotation 旋转



Procedural 属性

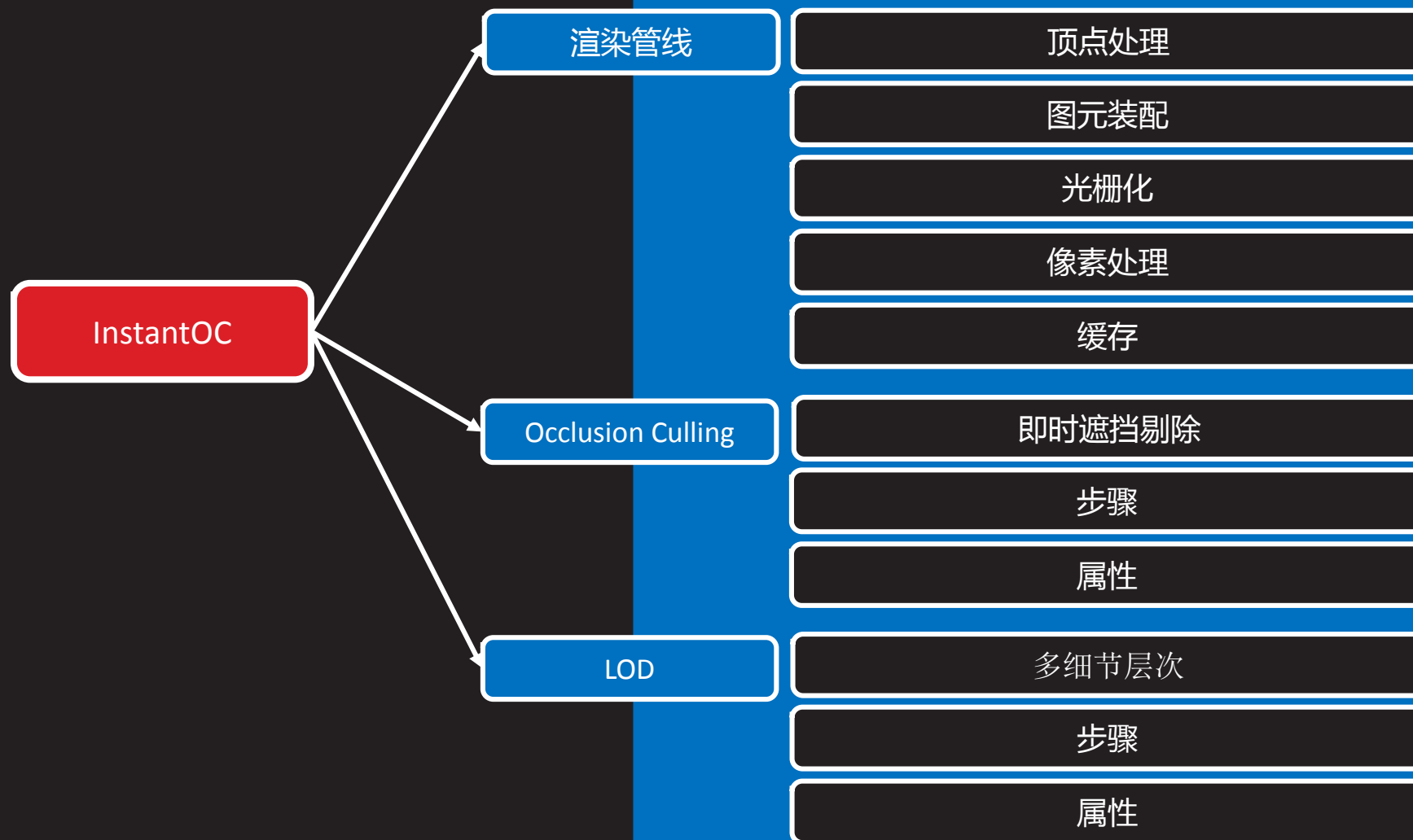
- Sun 太阳模式
 - None 没有
 - Simple 简单
 - Hight Quality 高质量
- Atmoshpere Thickness 大气层厚度
- Ground 地面颜色
- 如果为Environment Lighting的Sun属性设置一个平行光，场景中会根据平行光角度自动创建太阳，并且位置随平行光旋转而改变。如果不设置，系统将默认选择场景中最亮的平行光。



练习：制作游戏场景小地图

1. 创建游戏主角 Player ,
子物体包括：主摄像机，地图标记。
2. 地图标记 PleneMark
创建物体 Plane，材质 mt_PlayerMark，并指定颜色为蓝色，纹理为 mark
3. 创建敌人 Enemy，并添加地图标记(红色)
4. 为所有地图标记指定层 Mark
5. 地图摄像机 MapCamera
设置属性：Clear Flags Culling Mask Projection
Size Viewport Rect Depth

InstantOC

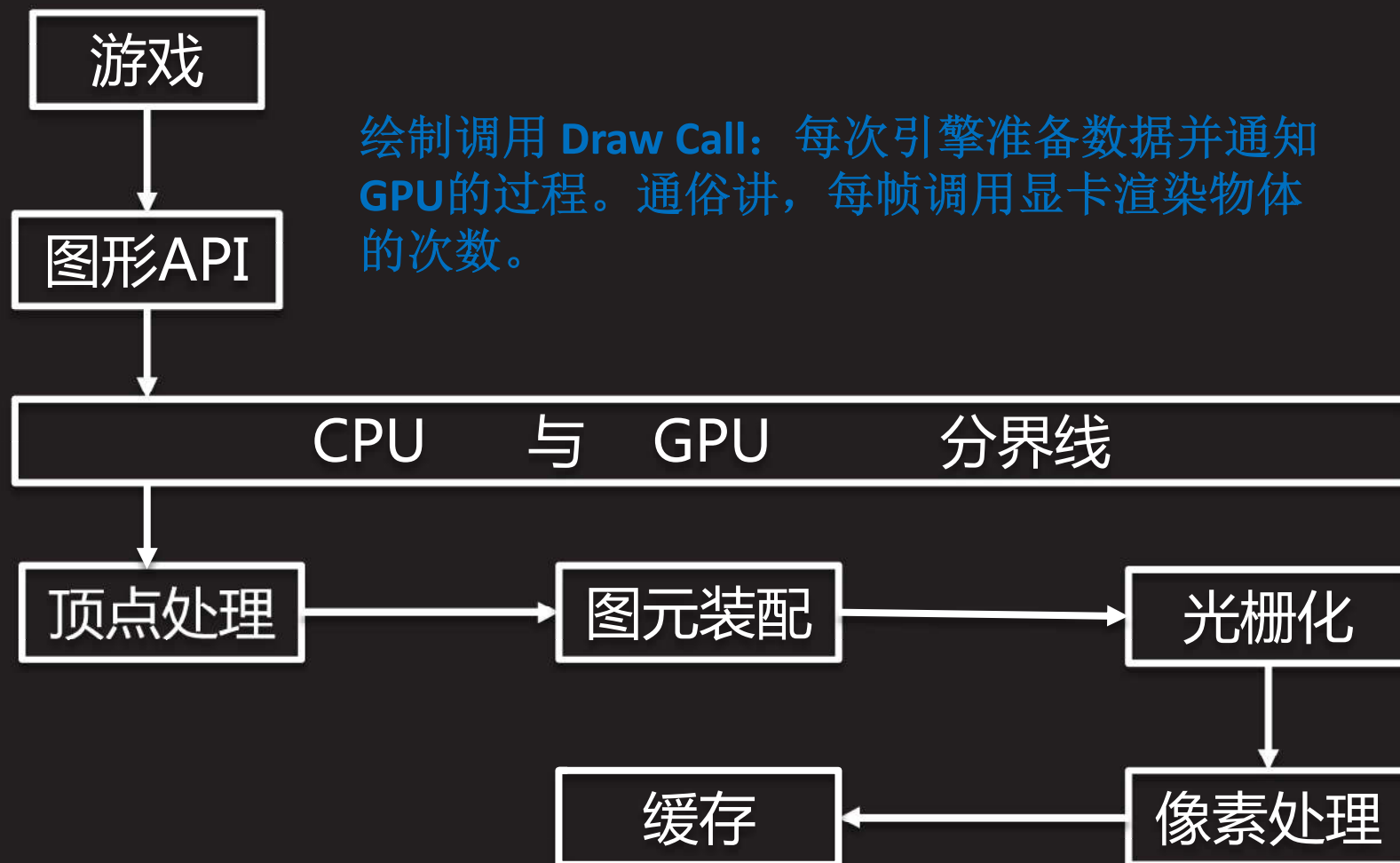


渲染管线



渲染管线

- 图形数据在GPU上经过运算处理，最后输出到屏幕的过程。

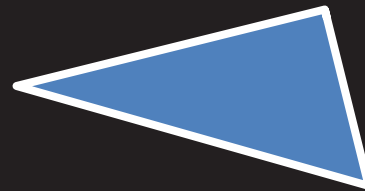
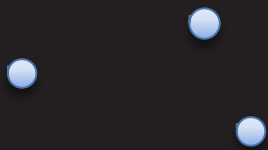


顶点处理

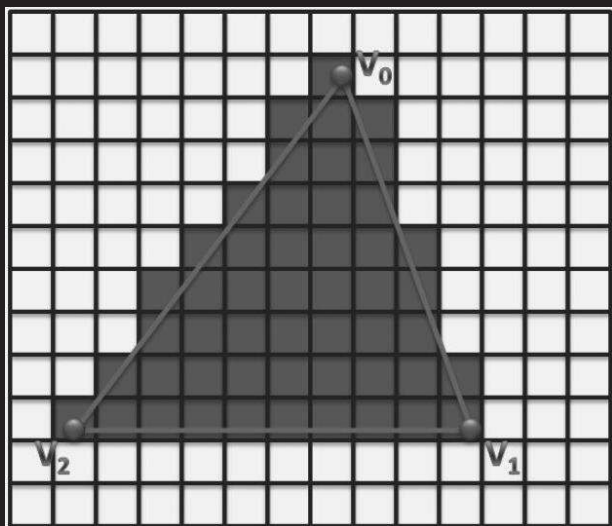
- 接收模型顶点数据。
- 坐标系转换。



- 组装面：连接相邻的顶点，绘制为三角面。



- 计算三角面上的像素，并为后面着色阶段提供合理的插值参数。



像素处理

- 对每个像素区域进行着色。
- 写入到缓存中。



缓存

- 一个存储像素数据的内存块，最重要的缓存是帧缓存与深度缓存。
- 帧缓存：存储每个像素的色彩，即渲染后的图像。帧缓存常常在显存中，显卡不断读取并输出到屏幕中。
- 深度缓存 z-buffer：存储像素的深度信息，即物体到摄像机的距离。光栅化时便计算各像素的深度值，如果新的深度值比现有值更近，则像素颜色被写到帧缓存，并替换深度缓存。

Occlusion Culling



即时遮挡剔除

- 即时遮挡剔除 Instant Occlusion Culling
- 遮挡剔除：当物体被送进渲染流水线之前，将摄像机视角内看不到的物体进行剔除，从而减少了每帧渲染数据量，提高渲染性能。



步骤

- 1.创建层
- 2.为游戏物体指定层(将参与遮挡剔除)与标签(将自动附加IOClod脚本)
- 3.物体添加碰撞器Collider组件
- 4.摄像机附加脚本IOCcam



属性

- Layer mask : 参与遮挡剔除的游戏对象层。
- IOC Tag : 将为指定标签的游戏对象自动添加IOClod脚本对象。
- Samples : 每帧摄像机发射的射线数目。数量多剔除效果好，但性能开销大。通常在150—500之间。
- Rays FOV : 射线视野，应大于摄像机视野Field of View。
- View Distance : 视图距离，射线长度。
将影响摄像机Clipping Planes -Far 数值。
- Hide Delay : 延迟隐藏，当物体被剔除时延迟的帧数，建议50—100之间。

属性(续1)

- PreCull Check：检查采集信息，建议勾选，可以提高剔除效率。
- Realtime Shadows：实时阴影，如果场景需要实时阴影，建议启用，确保剔除物体显示正常的阴影。

LOD



多细节层次

- 多细节层次 Levels of Detail

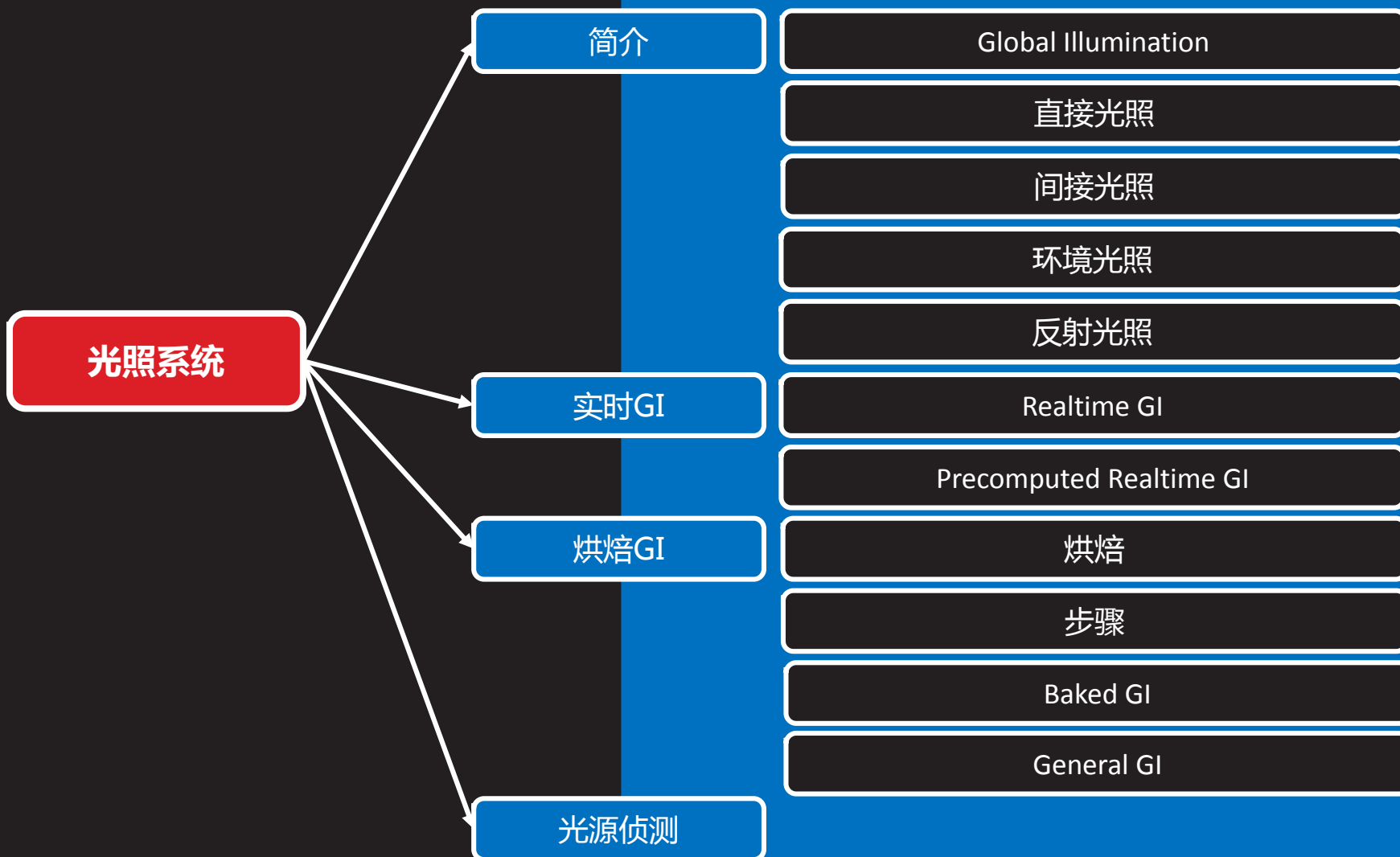
LOD技术指根据物体模型的节点在显示环境中所处的位置和重要度，决定物体渲染的资源分配，降低非重要物体的面数和细节度，从而获得高效率的渲染运算。

- 1.创建层
- 2.创建空物体并将模型添加到其中
模型命名：Lod_0 Lod_1 Lod_2
- 3.为父物体指定层与标签
- 4.父物体或子物体添加碰撞器组件
- 5.摄像机附加脚本IOCcam

属性

- Lod 1 distance : 摄像机到物体距离小于当前距离时, 使用Lod_0模型。
- Lod 2 distance : 摄像机到物体距离大于Lod1且小于当前距离时, 使用Lod_1模型, 大于当前值使用Lod_2模型。
- Lod Margin : LOD边缘, 如果物体在LOD各阶段过渡时发生闪烁, 尝试提高当前值。

光照系统



简介



Global Illumination

- 简称GI，即全局光照。
- 能够计算直接光、间接光、环境光以及反射光的光照系统。
- 通过GI算法可以使渲染出来的光照效果更为真实丰富。



直接光照

- 从光源直接发出的光，通过Light组件实现。
- Type 类型: 灯光对象的当前类型
 - Directional Light 平行光：平行发射光线，可以照射场景里所有物体，用于模拟太阳。
 - Point Light 点光源：在灯光位置上向四周发射光线，可以照射其范围内的所有对象，用于模拟灯泡。
 - Spot Light 聚光灯：在灯光位置上向圆锥区域内发射光线，只有在这个区域内的物体才会受到光线照射，用于模拟探照灯。
 - Area Light 区域光：由一个面向一个方向发射光线，只照射该区域内物体，仅烘焙时有效，用在光线较为集中的区域。

直接光照(续1)

- Range 范围：光从物体的中心发射的范围。仅适用于点光源和聚光灯。
- Spot Angle 聚光角度：灯光的聚光角度。只适用于聚光灯。
- Color 颜色：光线的颜色。
- Intensity 强度：光线的明亮程度。
- Culling Mask 选择遮蔽层：选择要照射的层Layer。

练习：为场景添加光源

课堂练习



直接光照(续2)

- Shadow Type 阴影类型：Hard 硬阴影、Soft 软阴影
 - Strength 硬度：阴影的黑暗程度。
 - Resolution 分辨率：设置阴影的细节程度。
 - Bias 偏移：物体与阴影的偏移。
- 通过 Mesh Renderer 组件启用禁用阴影
 - Cast / Receive Shadows 当前物体是否投射/接收阴影。
 - Off 不投射阴影，On 投射阴影，Two Sided 双面阴影，Shadows Only 隐藏物体只投射阴影
- 阴影剔除：设置显示阴影的距离
Edit->Project Settings->Quality->Shadows Disdance

间接光照

- 物体表面在接受光照后反射出来的光。
- 通过 Light 组件中 Bounce Intensity 反弹强度控制。
- 可以通过 Scene 面板 Irradiance 模式查看间接光照。
- 注意：
只有标记 Lightmapping Static 的物体才能产生间接反弹光照。

环境光照

- 作用于场景内所有物体的光照，通过 Environment Lighting 中 Ambient 控制。
- Ambient Source 环境光源
 - Skybox 通过天空盒颜色设置环境光照
 - Gradient 梯度颜色
 - Sky 天空颜色、Equator 地平线颜色、Ground 地面颜色
 - Ambient Color 纯色
- Ambient Intensity 环境光强度
- Ambient GI 环境光GI模式
 - Realtime 实时更新，环境光源会改变选择此项。
 - Baked 烘焙，环境光源不会改变选择此项。

反射光照

- 根据天空盒或立方体贴图计算的作用于所有物体的反射效果，通过 Environment Lighting 中 Reflection 控制。
- Reflection Source 反射源
 - Skybox 天空盒
Resolution 分辨率 Compression 是否压缩
 - Custom 自定义
Cubemap 立方体贴图
- Reflection Intensity 反射强度
- Reflection Bounces 使用 Reflection Probe 后允许不同游戏对象间来回反弹的次数。

实时 GI



Realtime GI

- 所谓"实时"是指在运行期间任意修改光源，而所有的变化可以立即更新。
- 正是由于 Unity 5 引入了行业领先的实时全局光照技术 Enlighten 系统，才可以在运行时产生间接光照，使场景更为真实丰富。
- 操作步骤：
 1. 游戏对象设置为 Lightmapping Static
 2. 启用 Lighting 面板的 Precomputed Realtime GI
 3. 点击Build按钮（如果勾选 Auto 编辑器会自动检测场景的改动修复光照效果）

Precomputed Realtime GI

- Realtime Resolution : 实时计算分辨率。
- CPU Usage : CPU利用率，值越大实时渲染效率越高。



烘焙 GI



烘焙 Lightmap

- 当场景包含大量物体时，实时光照和阴影对游戏性能有很大影响。使用烘焙技术，可以将光线效果预渲染成贴图再作用到物体上模拟光影，从而提高性能。适用于在性能较低的设备上运行的程序。



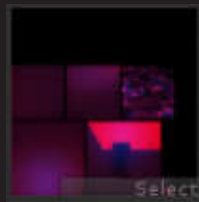
- 1. 游戏对象设置为 Lightmapping Static。
- 2. 设置Light组件Baking属性。
- 3. 启用 Lighting 面板的 Baked GI。
- 4. 点击Build按钮。（如果勾选 Auto 编辑器会自动检测场景的改动修复光照效果）
- Light 组件 Baking 属性：烘焙模式
 - Realtime 仅实时光照时起作用。
 - Baked 仅烘焙时起作用。
 - Mixed 混合，烘焙与实时光照都起作用。
- 可以通过 Scene 面板 Baked 模式查看光照贴图。

Baked GI

- Baked Resolution : 烘焙光照贴图分辨率, 建议比实时GI分辨率高10倍。
- Baked Padding : 光照贴图中各形状间距, 取值2到200。



BP = 2



BP = 20

- Compressed : 是否压缩光照贴图, 压缩则提高性能, 缩小容量, 但画质会降低。



Baked GI(续1)

- Ambient Occlusion : 环境遮挡表面的相对亮度, 值越高遮挡处和完全曝光处区别越大, 建议为1。



AO = 0



AO = 5

- Final Gather : 选中后提高烘焙质量, 但会消耗更多时间。
- Atlas Size : 光照贴图尺寸。

General GI

- 常规GI设置，同时适用于实时GI与烘焙GI。
- Directional Mode 定向模式：
 - Non – Directional 无定向模式，使用1种光照贴图存储光照信息。
 - Directional 定向模式，使用2种光照贴图存储光照信息，相比之下效果更好，但空间占用更大。
 - Direction Specular 定向反射模式，使用4种光照贴图存储光照信息，效果最好，但占用空间最大。
- 除无定向模式外必须运行在GL ES2.0与SM2.0以上的设备。
 GLES 指 OpenGL ES 针对移动端，从 iPhone5s 开始支持；
 SM 指 Shader Model 针对 PC 端，目前大部分显卡支持。

General GI(续1)

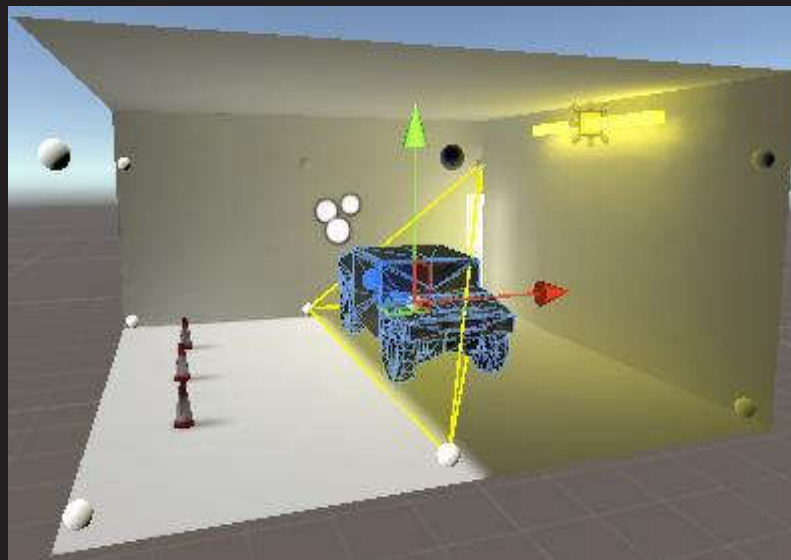
- Indirect Intensity 最终间接光照、反射光照的强度。1为默认强度，小于1则减低强度，大于1则增大强度。
- Bounce Boost 增强间接光照。
- Default Parameters 高级GI参数。
 - Default 默认
 - Default – HighResolution 高分辨率
 - Default – LowResolution 低分辨率
 - Default – VeryLowResolution 非常低的分辨率

光源侦测



光源侦测 Light Probes

- 由于 LightMapping 只能作用于 static 物体，所以导致运动的物体与场景中的光线无法融合在一起，显得非常不真实。而 Light Probes 组件可以通过 Probe 收集光影信息，然后对运动物体邻近的几个 Probe 进行插值运算，最后将光照作用到物体上。



步骤

- 1. 创建游戏对象 Light Probe Group。
- 2. 添加侦测小球 Add Probe。
- 3. 点击 Build 按钮。（如果勾选 Auto 编辑器会自动检测场景的改动修复光照效果）
- 4. 勾选需要侦测物体的 MeshRenderer 组件的 Use Light Probes 属性。

声音

声音

声音

简介

Audio Source

3D Sound Settings

声音



简介

- Unity 支持的音频文件格式：
mp3 , ogg , wav , aif , mod , it , s3m , xm。
- 声音分为2D、3D两类
3D声音：有空间感，近大远小。
2D声音：适合背景音乐。
- 在场景中产生声音，主要依靠两个重要组件：
Audio Listener 音频监听器：接收场景中音频源Audio Source发出的声音，通过计算机的扬声器播放声音。
Audio Source 音频源

Audio Source

- 音频源：
 - Audio Clip 音频剪辑：需要播放的音频资源。
 - Mute 静音：如果启用，播放音频没有声音。
 - Play On Awake 唤醒播放：勾选后场景启动时自动播放。
 - Loop 循环：循环播放音频。
 - Volume 音量：音量大小
 - Pitch 音调：通过改变音调值调节音频播放速度。1是正常播放。
 - Stereo Pan：2D声音设置左右声道。
 - Spatial Blend：2D 与 3D声音切换。

3D Sound Settings

- 3D 声音设置
 - Volume Rolloff 音量衰减方式
 - Min Distance 开始衰减距离
 - Max Distance 结束衰减距离

