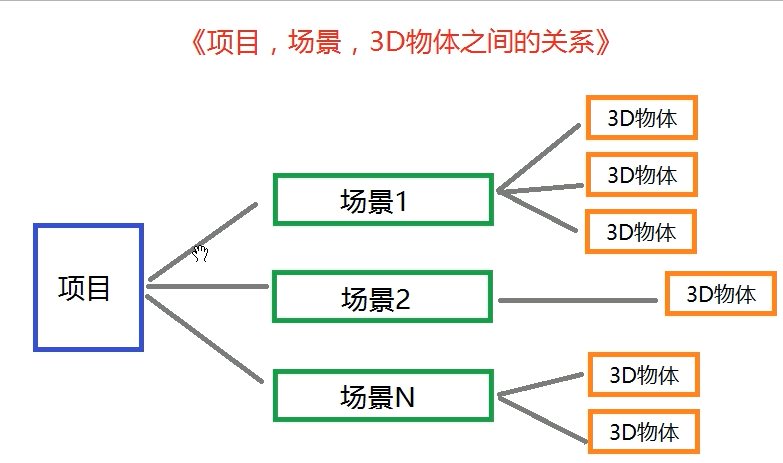
快捷键操作：

Alt+左键：旋转场景

Alt+右键：将场景拉远和拉近

项目、场景、3D物体之间的关系

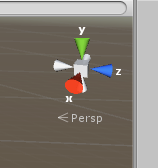
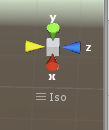


通过工具栏上的“Local”和“Global”可以切换物体自身和世界的坐标系。

世界坐标观察模式

Persp：透视模式，近大远小。

ISO：正交模式，远近一样大。

# 3D模型

Ctrl+D：复制物体

F：快速聚焦某个物体

工具栏对应：Q、W、E、R、T

## 材质球与贴图

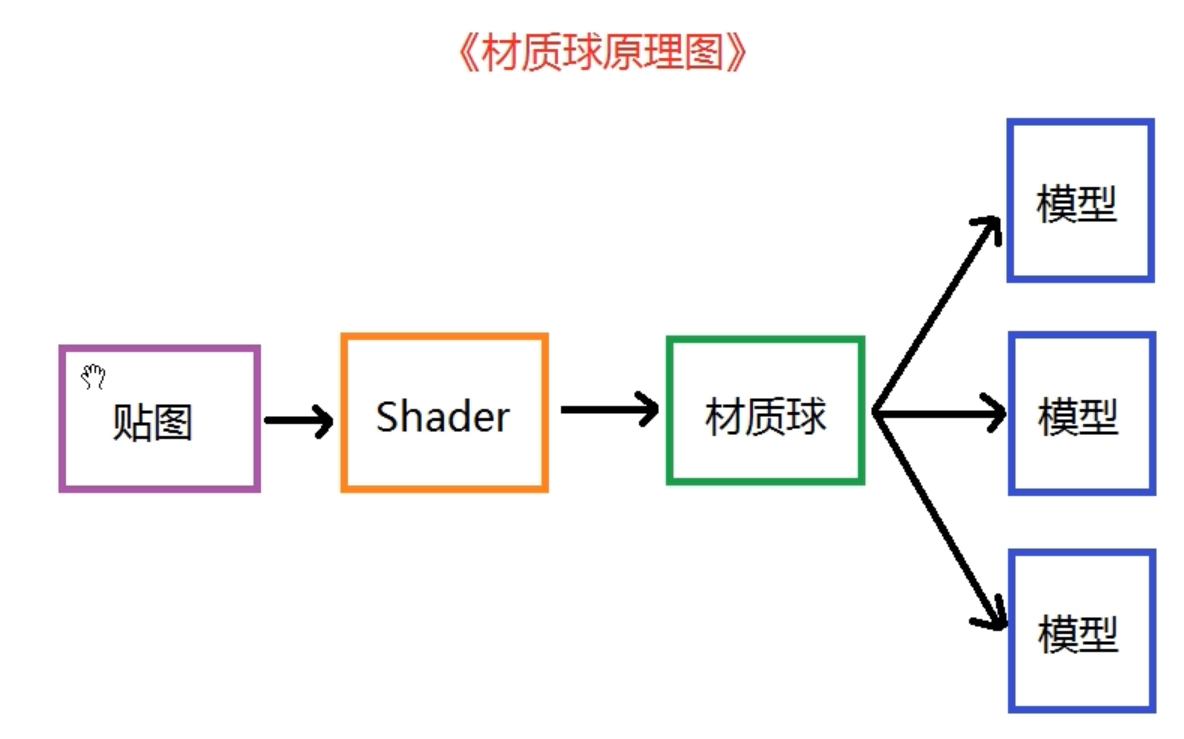
### 材质球与贴图介绍

材质球：Material，材质：物体的质地，物体看起来是什么做的。

贴图：Texture，就是“普通的图片”，用于材质球的显示。

Shader着色器

### 材质球原理图



### 父子关系

有时候一个物体由很多的小物体组成，管理小物体很不方便，这时候就可以在Hierarchy面板中新建一个空物体，然后将其他物体放入到该物体中，就可以将该空物体作为一个所选择的小物体的父类，方便管理。

在该面板上任意两个物体都可以建立父子关系。

### 预制体

Prefab，预先准备好的物体，可以重复利用和使用。比如，重复地面的砖块，栏杆等，重复的物体。

#### 管理预制体

在Assets中，建立Perfabs文件夹。

将层级面板中的物体拖拽到Perject面板的Assets中，即可创建一个预制体。预制体的文件后缀是.perfab

在层级面板中，预制体与非预制体的区别是，预制体是蓝色的，非预制体是白色的。



#### 预制体的好处

编辑其中一个预制体，然后点击Inspector面板中Perfab中的Apply按钮，所有预制体都可以获得相同的效果。

#### 寻找一个预制体在哪个文件中

选择相应的预制体，然后在Inspector面板中点击Perfab中的select按钮，预制体的位置就在Perject面板中显示出来了。

## 6、基本组件之摄像机

**场景美化**

1. 新建plane作为“地板”
2. 给地板附上材质球
3. 附上之后发现，瓷砖太大，需要进行密度的加大，点击材质球，在Inspector面板中，调整Tiling(贴瓷砖的意思)的值。

#### 游戏窗口

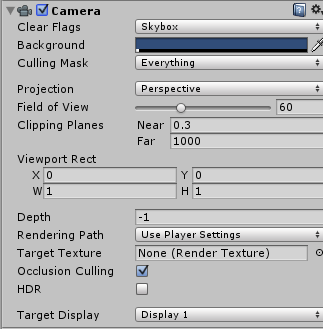
中间上部的游戏运行按钮屏幕剪辑，左边那个，如果选中，则表示处于游戏运行状态，这时切换下面的Scene面板进行编辑，都不会进行保存，重新点击运行按钮会恢复原状。

下面的Game窗口表示游戏预览窗口。

### 摄像机的常用操作

将摄像机快速到达当前的视角，选择GameObject菜单的Align With View，快捷键为Ctrl+Shift+F

#### 摄像机的相关属性



1. Clear Flags

Skybox：天空盒

Solid Color：固定颜色(纯色)，选择下面的background可以选择具体的颜色。

2、Projection(投影)

Perspective：透视模式，3D游戏使用

Orthographic：正交模式，2D游戏使用

1. Clapping planes(切割面)

摄像机有两个平行的平面，近的一个远的一个，Near近的那个的距离，Far远的那个的距离。

# 第7课 基本组件之灯光

## 灯光简介

当我们创建一个新的Scene的时候，默认会带有两个物体，一个摄像机，一个灯光。

隐藏和显示物体：，点击复选框就是显示，不选就是隐藏。

## 方向光

Directional Light，用于模拟太阳，模拟自然光。方向光任何地方都能照射到，就和太阳一样，但是要注意照射方向。

### 属性

Type：用于切换灯光的类型

Intensity：光照强度

Shadow Type：阴影类型

## 点光源

Point Light，用于模拟电灯泡的照射效果。

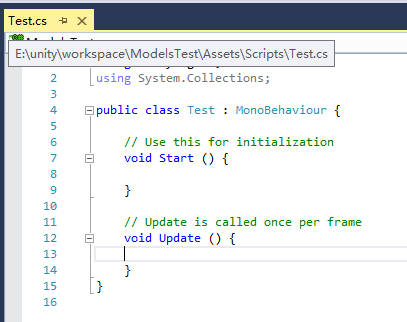
## 聚光灯

Spot Light

# 第8课 组件与脚本

每个物体右侧界面对应的大标题就是一个组件。

## 脚本代码介绍



Start()方法：当游戏运行起来，就会马上执行，且只执行一次。

Update()方法：循环调用，每帧调用一次。一秒钟大概执行60次。

## 使用脚本

1. 直接将脚本拖拽到Hierarchy面板上物体身上。
2. 直接将脚本拖拽到游戏物体的Inspector面板上。

运行游戏，脚本就会执行。

## Console面板

Clear：清除按钮，清除控制台中的信息。

Collapse：折叠功能，将相同的内容合并到一行显示，更新后面的数字。

Clear And Play：运行时清除之前的内容。

## 获取键盘输入

Input.GetKey(); //持续按下某键后，持续返回true

Input.GetKeyDown(); //按下某键的瞬间，返回true

Input.GetKeyUp(); //抬起某键的瞬间，返回true

## 2、获取鼠标的输入

Input.GetMouseButton(); //持续按下某键后，持续返回true

Input.GetMouseButtonDown(); //按下某键的一瞬间，返回true

Input.GetMouseButtonUp(); //抬起某键的一瞬间，返回true

返回值，bool类型

参数，鼠标按键索引值，0->左键，1->右键，2->中键

## 第10课 使用变换组件移动游戏物体

## 1、变换组件移动物体

1. 相关方法

gameObject.GetComponent<T>():获取相应组件的引用。查找当前游戏物体上的某个组件，然后保存它的引用。

Transform.Translate(Vector3, Space)：移动物体的位置。

游戏物体往某个方向移动；以自身坐标系或世界坐标系。

相关参数：Vector是向量，Vector3是三维向量。Vector3[struct]本质上是一个结构体，可以表示一个方向，也可以表示一个位置。

Space[enum]：空间

Space.Self：表示物体自身的坐标系

Space.World：表示物体所在的世界的坐标系

## 2、使用键盘控制移动的方向

使用键盘上的“W，A，S，D”来控制游戏物体的移动。