# 第1章 初识Unity

本章将介绍Unity的安装和初步使用。学习完本章后，你应当搭建好了一个开发环境，以及知道如何进行基本操作。

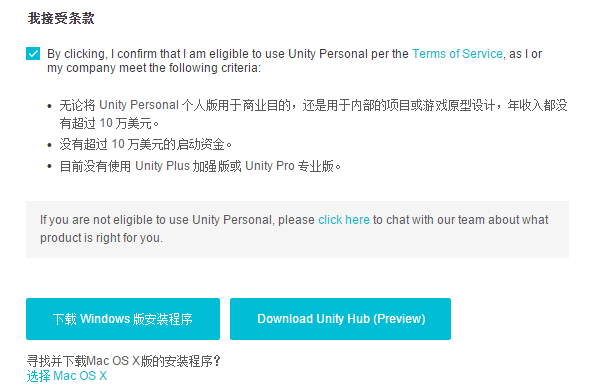
## 下载与安装

### 下载Unity安装程序

在Unity官网上下载Unity个人版（网址：https://unity3d.com）。



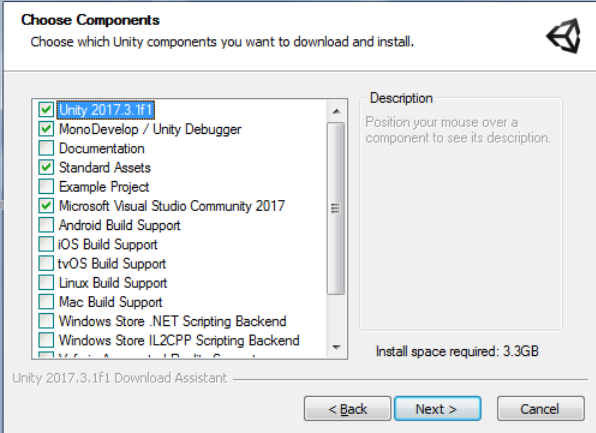
阅读个人版本的使用条件，确认后下载安装程序。



一般选择“下载Windows版安装程序”即可，Mac OS系统下默认下载Mac版本的安装程序。右边的“Unity Hub”是Unity新推出的一个方便同时管理多个Unity版本共存的工具，对初学者来说可以不用关心Unity Hub，直接用最新版本的Unity即可。而Unity Hub对于职业开发者来说很有必要。

### 安装Unity

安装程序UnityDownloadAssistant是一个很小的程序，执行后它会引导你下载并安装Unity。其中选择组件的一步需要注意：



建议勾选Standard Assets（标准资源）以及Example Project（示例工程），以便学习使用。如果这时候没有勾选标准资源，未来也可以在Asset Store中重新获取。

建议勾选Visual Studio Community 2017，目前VS2017已经是大量Unity开发者的首选IDE，且VS2017已经有微软官方的Mac OS版本。

### 多版本并存

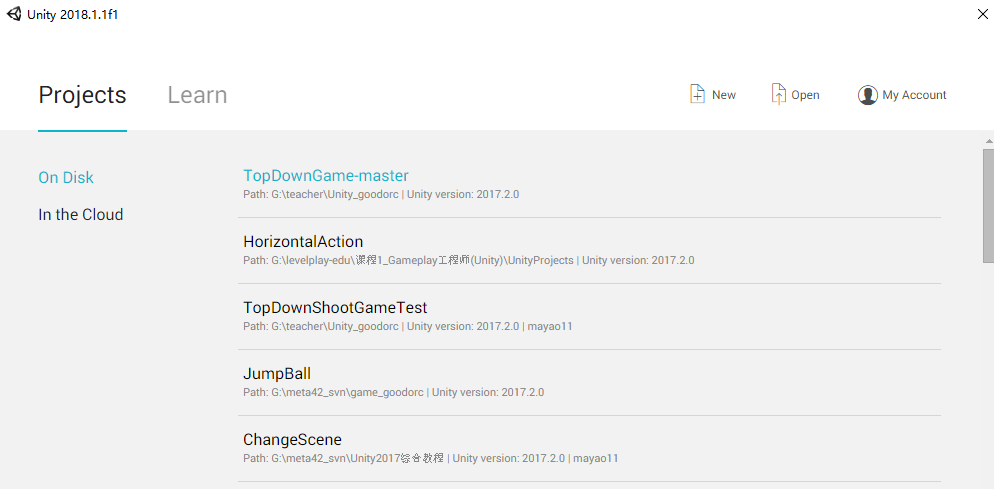
Unity支持同时安装多个版本，这在实际开发中非常方便，因为某些旧的项目可能需要使用低版本开发。

如有多版本共存的需求，请下载使用Unity Hub工具。

## 初次运行

### 工程页面

运行Unity程序，会打开下面的工程页面。



上图左边可以选择在本地或者云端的项目，一般我们建立本地项目即可。

中间较大的区域是所有项目的列表，每项都标明了项目路径和版本号以便查阅。点击某个项目即可打开该项目。

右上角分别是新建项目、打开现有项目以及账户功能。

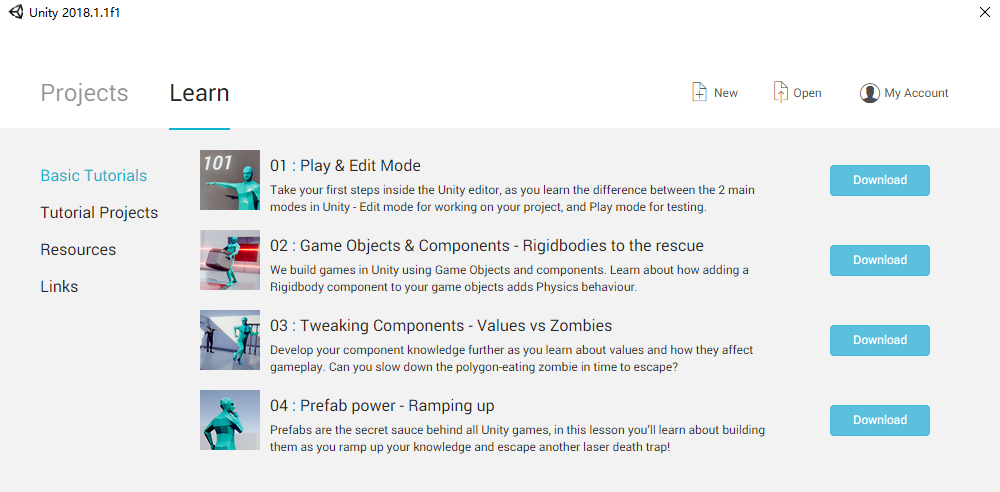
### 学习资料页面

Unity很人性化地提供了学习资料页面，点击这个窗口的Learn标签即可看到。

如上图，Unity提供了基本教程、案例工程、资源和链接共四大类学习资源。工程和案例可以下载学习；链接是一些文档，可供在网页上阅读。

### 新建工程

我们回到工程页面，点击右上角的New按钮即可来到新建工程的页面：



如上图，我们可以设置工程名称、在磁盘上的位置，指定3D或2D项目，导入资源包，以及打开或关闭Unity Analytics开关。

工程名称（Project Name）建议输入一个有意义的名称。

目录位置（Location）可以自选，保证充足的磁盘空间。

3D/2D项目选项，是为了能更方便地新建3D或2D工程。但是可以放心，Unity工程并不严格区分3D或2D，未来可以通过简单地设置即可在两种模式中间切换。如果不确定项目类型，设置为默认的3D即可。

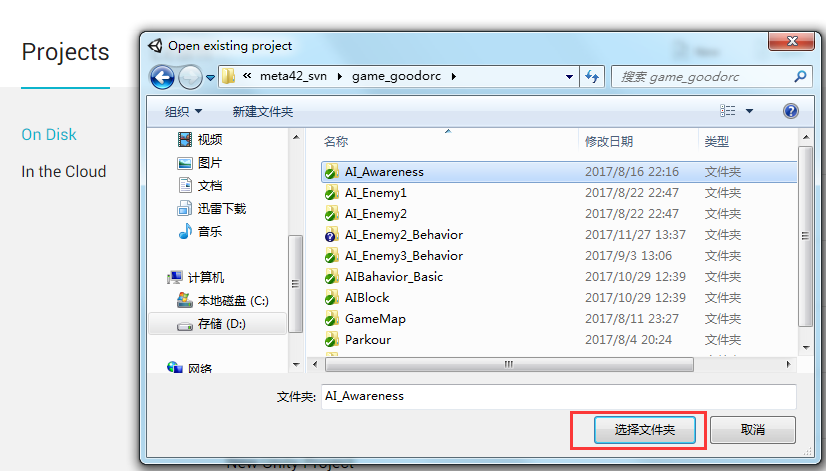
添加资源包（Add Asset Package）按钮用于新建工程时便导入资源包。资源包可以以后再导入，或者稍后在Asset Store上面下载，所以这步不是必须的。

Unity分析开关（Enable Unity Analytics），是Unity官方为开发者提供的用于优化工程的一个服务。此开关开启，则Unity会通过网络上传你的项目的部分数据，以便与许多其他工程进行对比分析，为开发者评测自己的项目提供参考。默认此开关关闭。

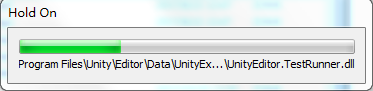
### 打开工程

使用工程页面的Open按钮，可以打开现有项目。

在打开现有工程窗口中选择选择或者进入一个Unity工程目录，则“选择文件夹”按钮就会变为可用，点击它就可以打开工程了。

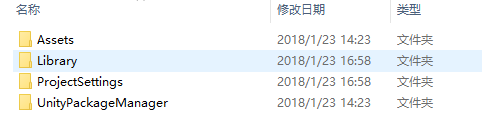


如上图，点击“选择文件夹”后，工程窗口就会关闭，Unity会开始加载项目。根据项目需要处理的资源量，会耗费一定时间。



稍等一会儿就正式进入Unity的工作界面了。

值得注意的是，Unity工程并不是特定的某个文件，而是一个特定结构的目录。



如上图，典型的Unity工程包含Assets、Library、ProjectSettings和UnityPackageManager四个目录。有时某些目录可能暂时缺失，但是至少Assets目录是一定存在的。

## 工程窗口（Project Window）

从本节开始，我们将逐一介绍Unity中最常用的一些窗口，以及它们的详细用法。主要包括：

1. 工程窗口（Project Window）

2. 场景视图窗口（Scene View）

3. 层级窗口（Hierarchy Window）

4. 检视窗口（Inspector Window）

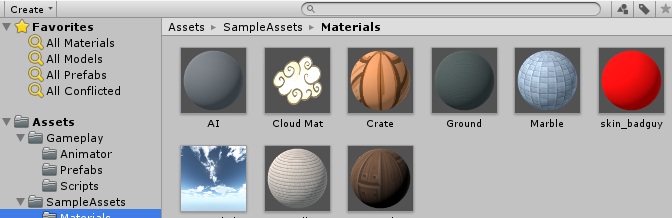
5. 工具栏（Toolbar）

6. 游戏视图窗口（Game View）

本书中称它们为六大界面，先从工程窗口开始介绍。

### 工程窗口基本功能

工程视图中，可以访问和管理属于这个工程的所有资源。



如上图，左侧的区域以树状结构展示了文件夹的结构。当选中左边的任何一个文件夹以后，右侧的窗口就会显示它的内容。可以点击文件夹左边的小三角形来展开或隐藏下一级文件夹，还可以按住Alt键来递归展开或隐藏所有子文件夹。

右侧的每一个资源都以图标的形式展示出来，图标通常代表了资源的类型（脚本、材质、子文件夹等）。最右下角的滑动条可以用来调整图标的大小，滑到最左边的时候，就会换成另一种层级的形式展示了。当选中某个资源的时候，滑动条左侧还会显示出资源的完整路径和名称，这在搜索资源时很有用。

文件夹结构的上方还有收藏夹（Favorites），你可以将经常用到的资源放在这里，以便快速找到。只需要直接将资源文件或文件夹拖到这里即可，另外还可以保存查询条件，下面会详细说明。

IMG_257

在右侧资源的上方还可以看到一个路径指示，它展示了当前所看到的文件夹的具体路径。可以在上面进行点击，在文件夹之间跳转。当使用搜索功能时，它又会变成搜索区域的显示（比如工程的Assets目录、选中的文件夹或是资源商店）。

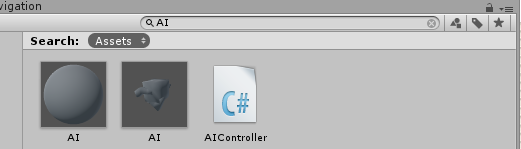
整个工程视图的上方还有一个工具栏。



工具栏最左边有一个“新建”（Create）按钮，可以用它在当前文件夹内新建资源或者子文件夹。它的右侧则是一系列搜索相关的工具。

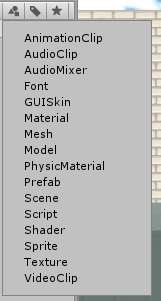
### 搜索功能

工程视图窗口有着强大的搜索功能，特别适合在大型工程中使用，或者当用户不太熟悉工程结构的时候使用。它的基本用法很简单，就是可以过滤符合搜索条件的资源显示到窗口中。



如果你输入多于一个关键字，则搜索结果会进一步缩小。例如如果输入“coastal scene”则会搜索名称中同时包含这两个单词的资源。

搜索栏的右侧有三个按钮，第一个用来限定搜索的资源类型。



第二个按钮用来限定资源的标签（Label），标签的概念会在后面讲解检视窗口（Inspector）时说明。由于标签可能会很多，标签按钮还带有一个微型搜索框。



使用更多关键字会进一步缩小搜索的范围。还有一些高级的搜索技巧，比如说，使用“t:”再加上资源类型名称，就可以指定资源类型；使用“l:”加上标签名称就可以指定搜索的标签。资源和标签也可以有多个，特别的是，指定多个类型可以同时搜索多种类型的资源，多个类型之间是“或”的关系。而相比之下多个标签和关键字都是“且”的关系。

例如搜索：

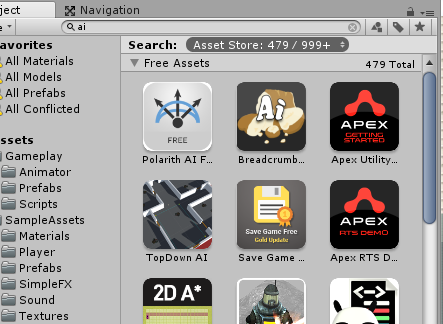
flash t:Material t:Texture l:Weapon

表示搜索名称为flash，类型为材质或贴图，且标签（label）为Weapon的资源。

第三个按钮用来将搜索的条件添加到书签中。

### 搜索资源商店

工程视图窗口的搜索功能，甚至还可以用在资源商店中。在搜索资源时，可以在路径显示那里切换到资源商店（Asset Store），这样就可以直接以同样的方式搜索并显示所有资源商店中的资源了。同样，前面所说的高级搜索技巧也可以用在资源商店中。且搜索资源商店时不仅会搜索资源的名称，还会搜索资源的描述信息，以便在大量资源中找到合适的。



选中了某个资源商店中的资源之后，可以在检视窗口中看到购买或是下载的选项。某些资源具有预览功能，比如3D模型甚至可以旋转查看。还可以跳转到专门的资源商店窗口中查看资源的详细信息。

### 快捷键

当窗口被激活时，可以使用该窗口的快捷键。以下是工程视图窗口的快捷键列表。

|  |  |
| --- | --- |
| Ctrl/Cmd + F | 选中搜索框 |
| Ctrl/Cmd + A | 全选当前文件夹的所有资源 |
| Ctrl/Cmd + D | 直接拷贝旋转的资源（复制并粘贴）。 |
| Delete | 删除（需要二次确认） |
| Delete + Shift | 删除（不再次确认） |
| Backspace + Cmd | 删除（不再次确认） (OSX系统) |
| Enter | 重命名资源或文件夹 (OSX系统) |
| Cmd + 下箭头 | 打开选中的资源 (OSX系统) |
| Cmd + 上箭头 | 回到上级文件夹 (OSX系统) |
| F2 | 重命名资源或文件夹 (Windows系统) |
| 回车 | 打开选中的资源 (Windows系统) |
| Backspace | 回到上级文件夹 (Windows系统) |
| 右箭头 | 展开文件夹或者右移 |
| 左箭头 | 折叠选中的文件夹或者左移 |
| Alt + 右箭头 | 展开某些带有子资源的对象 |
| Alt + 左箭头 | 折叠带有子资源的对象 |

## 场景视图窗口（Scene View）



场景视图窗口是用来创造游戏世界的窗口。我们会用场景视图来选择和定位背景、角色、相机、灯光以及所有类型的游戏物体（GameObject）。在场景视图中选择和移动物体是大多数人学习Unity的第一步。

### 浏览场景

场景视图有一系列工具帮助你在场景中高效地移动和浏览。

### 场景Gizmo

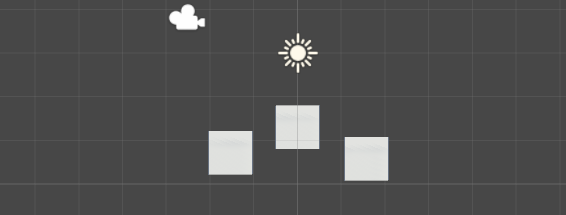
场景Gizmo位于场景视图的右上角。它显示了当前查看场景的视角方向，且可以通过点击它快速改变查看场景的视角。



场景Gizmoz图标的每个面都有一个柄。三种颜色的柄分别代表X、Y、Z轴。点击任意一个轴可以让视角立即旋转到该角度，所以分别可以得到顶视图、前侧视图和左侧视图。另外还可以用右键点击它选择一些其它的预置角度。

还可以在这里切换透视摄像机和正交摄像机。具体方法是点击Gizmoz中间的立方体，或者点击立方体下方的文字即可。正交视角适于某些刻意不需要透视视角的游戏，在前面的章节已经有了详细的介绍。另外正交视角可以方便地从固定的视角查看场景的比例。





（同一个场景的两个正交视角。）

在2D模式下场景Gizmoz是不显示的，因为2D模式下沿X、Y轴旋转视角是没有意义的。

#### 移动、旋转和缩放视角

移动、旋转和缩放视角是场景视图中最主要也最常用的功能。Unity提供了多种方式尽可能提高操作的效率：

1、方向键移动：可以使用4个方向键在场景中移动，就好像在场景中行走一样。按住shift键可以加快速度。

2、小手工具：当小手工具被选中时（快捷键：Q），以下鼠标操作会被启用。

IMG_259 鼠标左键拖拽场景，可以在场景中移动。

IMG_260 按住Alt键并用鼠标左键拖拽场景，可以在场景中旋转视角。

IMG_261 按住Alt键并用鼠标右键拖拽场景，可以缩放视角（也可以看作前进和后退）。

以上操作都可以按住Shift键以加快移动速度。

#### 飞行浏览模式

飞行浏览模式可以让你自由在场景中以主视角穿梭浏览，就像在很多第一人称视角游戏一样。

使用飞行浏览模式首先需要按住鼠标右键，然后用W、A、S、D键来分别朝前、左、后、右来移动，Q和E键用来向下和向上移动。按住Shift键可以移动得更快。

只能在透视摄像机下使用飞行模式，在正交摄像机下此功能的移动方式会不一样。

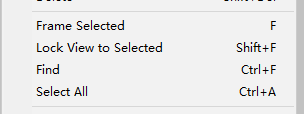
#### 一些快捷的浏览操作

为了提高效率，Unity还提供了另外一些移动方法，这些方法的优点是无论当前选中哪一种小工具，都可以快速浏览而不用切换工具。

| 动作 | 三键鼠标 | 双键鼠标或触控板 | Mac系统下单键鼠标或触控板 |
| --- | --- | --- | --- |
| 移动 | 按住Alt和鼠标中键并拖动 | 按住Alt键、Ctrl键和鼠标左键并拖动 | 按住Alt、Command和鼠标左键并拖动 |
| 环绕当前显示的中心旋转 | 按住Alt和鼠标左键并拖动 | 按住Alt和鼠标左键并移动鼠标 | 按住Alt和鼠标左键并拖动 |
| 前进后退 | 使用鼠标滚轮，或按住Alt和鼠标右键并拖动 | 按住Alt和鼠标右键并拖动 | 使用双指滑动的手势来操作，或者按住Alt、Control和鼠标左键并拖动 |

#### 将物体置于视野中心

具体观察一个物体时，我们需要将它置于视野范围的中间，可以这样操作：在层级视图（Hierarchy）中选中该物体，然后将鼠标移动到场景视图中，然后再按F键，这样视野就会以物体为中心了。有时物体正在运动，使用Shift+F键就可以一直跟踪物体。这两种功能分别对应主菜单中的Edit > Frame Selected和Edit > Lock View选项。



### 修改物体位置

要选中一个游戏物体（GameObject）最常用的有两种方法，在场景视图中点击该物体或者在层级窗口中点击它的名称。要选择或者取消选择多个物体，只需按住Shift键不放同时点击鼠标、或者拖拽方框进行多选即可。

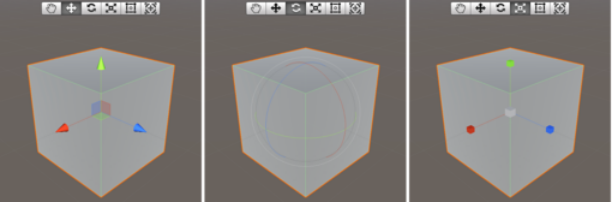
被选中的物体在场景视图中会被高亮显示，默认高亮的方式是对物体进行一个显眼的描边处理。可以通过选择菜单的Preference > Color选项，修改Selected Wireframe和Selected Outline的颜色来改变默认的表示。另外，选中的物体上会出现可以操作的小图标，具体的图标由当前选中的工具决定。

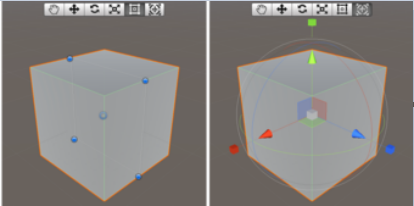
#### 移动、旋转、缩放以及RectTransform



场景视图工具栏中的第一个工具——小手工具已经在前面介绍过了。接下来分别是移动、旋转、缩放、Rect Transform以及Transform工具，它们各有各的作用，我们可以借助它们来编辑游戏物体。一般编辑物体的transform组件（也就是修改物体的旋转、位移、缩放），可以对选中的物体选择旋转、位移或缩放工具，并在场景视图中拖动来修改物体的transform属性，另外也可以直接在检视窗口中修改物体的Transform组件的参数。

以上几种小工具的快捷键分别是W、E、R、T、Y，顺序与键盘按键的位置相对应。





(分别选中几种工具时，场景视图中有不同的表示方法）

**移动**

移动的图标在场景中以三个箭头表示，既可以分别拖动三个独立的箭头来修改物体在X、Y、Z轴的位置，也可以拖动三个箭头两两之间的小平面让物体在该平面上移动。

另外还有一种有用的操作方法，按住Shift键时，图标会变成一个扁平的方块。这个扁平方块表示这时拖动物体，会让物体在垂直于当前视线的平面上移动。

**旋转**

选中了旋转工具后，就可以拖动窗口中物体中间表示旋转的图标。和移动一样，旋转轴也表示为红、绿、蓝，和X、Y、Z轴分别对应，表示以三个轴为重心进行旋转。最后，最外面还有一层大的圆球，可以用来让物体沿着从屏幕外到屏幕内的这根轴进行旋转，可以理解为当前的屏幕空间的Z轴。

**缩放**

缩放工具用来改变物体的放大、缩小比例，可以同时沿X、Y、Z轴放大，也可以只放缩一个方向。具体的操作方法可以尝试拖动红、绿、蓝三个轴的方块，还有中间白色的方块。要特别注意的是，由于Unity的物体具有层级关系，父物体的缩放比例会影响子物体的缩放，所以不等比的缩放可能会让子物体处于一个奇怪的状态。

**矩形变换（RectTransform）**

矩形变换（RectTransform）通常用来给2D元素，比如精灵和UI元素定位，但是其实在给3D物体定位时也是有用的。它把旋转、位移、缩放的操作统一为一种图标：

在矩形范围内点击并拖动，可以让物体在该矩形的平面上移动。

点击并拖动矩形的一条边，可以沿一个轴缩放物体的大小。

点击并拖动矩形的一个角，可以沿两个轴缩放物体的大小。

当把鼠标放在靠近矩形的点的位置，但又不过于靠近时，鼠标指针会变成可旋转的指示，这时拖拽鼠标就可以沿着矩形的法线旋转物体。

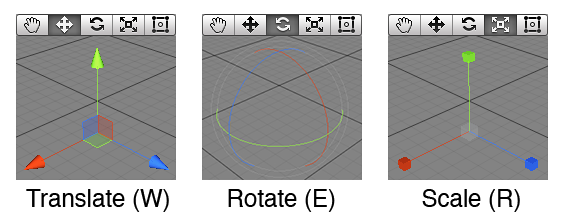
注意到在2D模式下，无法改变物体沿Z轴方向的旋转、位移和缩放，这种限制其实是很有用的。矩形变换工具一次只能在一个平面上进行操作，将场景视图的当前视角转到另一个侧面，就可以看到矩形图标出现在另一个方向，这时就可以操作另一个平面了。

#### 具体操作说明

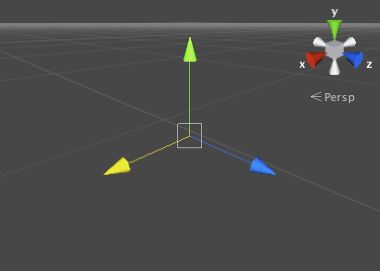
这里我们再深入介绍一下。

IMG_257

位移、旋转、缩放工具可以对场景中的任意物体使用。当你选中了一个物体时，你就会看到下图中所示的辅助线框，三种线框表示方法不同，以后会经常用到。

（上图分别是位置、旋转、缩放三者的辅助线框）

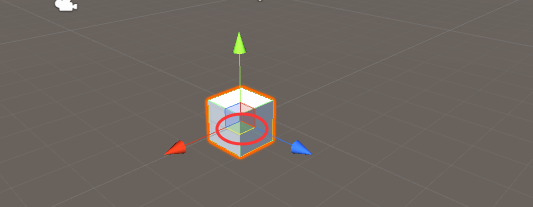
当你选中某一个轴并拖动时，你会发现该轴变成了黄色，而且对旋转、缩放变换也是类似的。你会发现物体只会修改和该轴有关的参数，而不会影响到另外两个轴的参数。



（当选中一个轴时，该轴变为黄色）

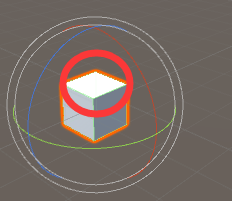
#### 一些高效的操作方法

除了只调整某一个轴上的参数，我们也可以同时改变多个轴上的参数，但是改变多个值时，三种工具（位移、旋转、缩放）的具体操作不太一样。对于移动操作来说，我们可以同时修改两个轴上的位置，也就是让物体沿着某个平面滑动，例如沿着XZ平面移动。具体做法是拖拽两个轴之间的平行四边形辅助框线即可。一共有XY、YZ、XZ三个平面。



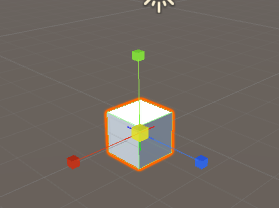
（拖拽圆圈所标注的区域，即可同时修改X轴和Z轴的位置）

对旋转操作来说，拖拽非轴线的位置，就可以自由旋转物体。但是实际操作中，建议还是尽可能只改变某一个轴线进行旋转，否则反而会给自己带来混乱。



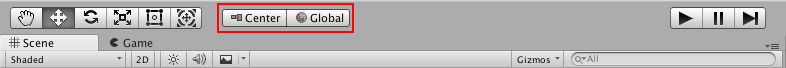
（对于旋转工具来说，拖拽大部分非轴线的位置都可以随意旋转物体）

缩放工具也不太一样，因为“等比例缩放”可能比沿某一个轴缩放更为常用，所以缩放工具的辅助框线提供了四个点的位置，分别是周围的红、绿、蓝色方块以及中央的白色方块。拖动三个轴上的方块可以让物体只沿一个轴放缩，这个放缩实际上会引起拉长和压扁的结果。如果你需要等比例缩放物体，那么就可以拖动中央白色的方块，让物体在X、Y、Z三个轴上以等比例的方式放大或缩小。



（中央的白点可以用于等比例放大和缩小物体）

#### 局部/世界坐标系切换，中心/基准点切换



3D世界中物体的定位还有两个复杂的问题，它会影响到前面的旋转、位移、缩放等工具的行为。点击上图中的Center和Local按钮，可以分别切换到另外两种模式：Pivot和Global：



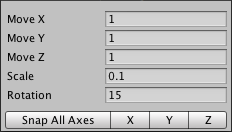


* Pivot模式 将用来操作的小图标显示在物体三维网格的基准点上。
* Center模式 将用来操作的小图标显示到物体外形边界的中心。
* Local局部坐标系 用来操作的小图标的旋转角度，和当前物体保持一致。
* Global世界坐标系 用来操作的小图标的旋转角度，保持和世界坐标系一致。

在给物体定位时，局部坐标系和世界坐标系的切换会经常用到。比如一个常见的俯视角游戏中，摄像机会沿X轴向下旋转45度到90度，形成俯视的效果。这时，在Local模式下，移动摄像机的Z轴，摄像机会有拉近、拉远的效果。而在Global模式下，摄像机沿Z轴移动会和游戏场景的地面平行移动。

#### 按照单位坐标移动

当移动物体时按住Ctrl键（Mac系统下是Command键），物体的移动会以指定的单位长度移动，这样可以方便在某些游戏中调整位置。这个单位坐标有默认的设定，也可以在菜单的Edit > Snap Settings 里修改。



**吸附到平面**

当使用移动工具时，同时按住Shift键和Ctrl键（Mac系统下是Command键），可以让物体快速吸附到碰撞体（Collider）的表面，这个功能在布置场景时也十分有用。

**吸附到顶点**

“吸附到顶点”的功能在搭建某些场景时特别有用，它可以将一个物体模型的某个顶点，放在另一个物体模型的某个顶点上。例如，我们可以将赛车游戏的道路精确地放置在背景上，或是将一个道具精确地放置在某个模型上面。

操作步骤如下：

1. 选择你要移动的物体并选择移动工具。
2. 一直按住V键进入吸附顶点模式。
3. 鼠标指向要移动的物体的某个顶点，会有提示的白框指定以哪个顶点为准。
4. 拖拽该顶点到另一个物体上，就可以让拖拽的物体在目标物体的多个顶点之间进行移动。无论如何移动，编辑器都会保证要移动的物体的顶点和目标物体的顶点是重合的。
5. 松开V键和鼠标左键，移动完毕。另外还可以同时按下Shift+V键持续开启或关闭这个功能，就不需要一直按住V键了。

注：不仅可以让顶点和顶点对齐，还可以让顶点和平面对齐、顶点和模型基准点对齐。

### 场景视图工具条

如下图，场景视图工具条有多个选项来调整查看场景的显示方式。包括是否显示光照、是否开启声音等等。这些选项只对当前场景视图起作用，和最终的游戏效果无关。

IMG_256

#### 渲染模式

工具条最左侧的下拉菜单，用于选择要使用哪种渲染模式（DrawMode）来绘制场景。选项包含：

**1、Shading mode**

Shaded：显示物体的表面材质。

Wireframe：以线框图的形式显示物体的模型网格。

Shaded Wireframe：显示物体的表面材质并叠加网格线框。

**2、Miscellaneous**

Shadow Cascades：显示方向光源的[shadow cascades](https://docs.unity3d.com/Manual/DirLightShadows.html)。

Render Paths：渲染路径，每种颜色来对应一种渲染路径。蓝色代表deferred shading，绿色代表deferred lighting，黄色表示forward rendering，红色表示vertex lit。

Alpha通道：显示透明通道。

Overdraw：将物体显示为半透明的剪影，不透明度会叠加，用来展示各种物体交叠的情况。

Mipmaps：展示纹理尺寸是否合适，以不同的颜色显示。红色表示纹理贴图比必要的大，蓝色表示纹理尺寸不太够。通常，需要的纹理尺寸大小和分辨率以及镜头的远近有关系。

**3、Deferred**

包含几种模式，用来查看G-buffer（Albedo, Specular, Smoothness，Normal），具体的含义需要查看Deferred Shading的相关文档。

**4、Global Illumination**

全局光照相关。

#### 2D模式、光照和声音的开关

渲染模式右边还有几个按钮，是一些开关，它们只在场景视图中发挥作用。

* 2D：在2D和3D视图之间切换。2D模式下，摄像机会保持指向Z轴，且X轴在右侧、Y轴指向上方。
* Lighting：开启和关闭光照，包括环境光和物体的着色器。
* Audio：开启和关闭音频。

#### 渲染效果开关

小喇叭（音频）开关的右侧还有一个山形的按钮，是渲染效果开关。它里面有一些选项，用来开启和关闭一些渲染效果。

* Skybox：开启和关闭天空盒的显示。
* Fog：开启和关闭雾。
* Flares：灯光的光晕开关。
* Animated Materials：动态材质是否以动画方式显示。

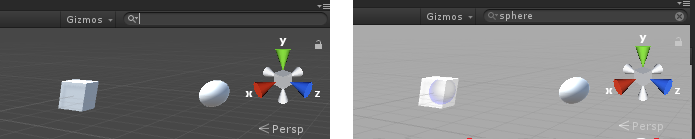
另外，这个渲染开关本身可以一起改变以上四个子项的开启和关闭。

#### 辅助线框（Gizmoz）

辅助线框（Gizmoz）菜单控制一系列的图标显示，辅助线框菜单的入口在场景视图和游戏视图中都可以看到，接下来的章节我们会细说关于辅助线框菜单的具体用法。

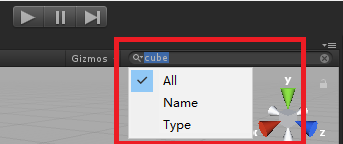
#### 搜索框

场景工具栏最右侧，是一个搜索框，它主要是起到一个过滤器的功能。它以搜索框内的信息作为关键字，过滤场景中的内容。被过滤掉的物体会以减弱对比度的方式显得不明显，且会变成半透明，相对来说要搜索的物体就很明显了。且这时层级窗口中也会只显示搜索到的物体。



搜索前（左）和搜索sphere的效果（右图）

另外，搜索物体还可以指定几种模式：搜索物体名称、搜索物体类型或是二者都包括。点击搜索框左侧的放大镜小图标进行选择，如图：

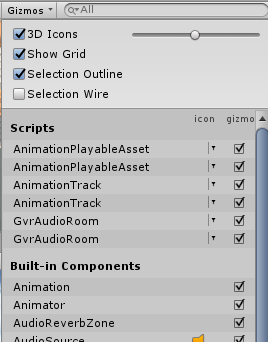


## 辅助线框（Gizmoz）菜单

场景视图和游戏视图都拥有各自的辅助线框（Gizmoz）菜单。在工具条中点击辅助线框的按钮就可以设置辅助线框。再次提示：辅助线框和图标，只在编辑器中，开发时可以看到，它们不会出现在最终的发布版本中。



（场景视图中的辅助线框）



（辅助线框菜单的具体设置，截图是一部分）

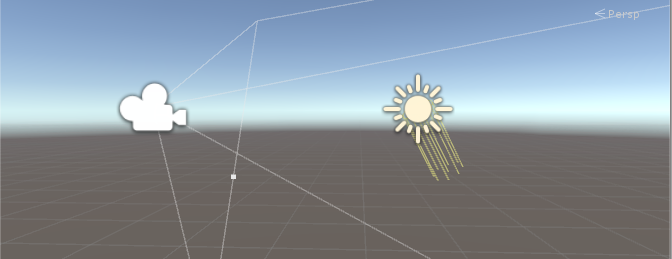
| 选项 | 功能 |
| --- | --- |
| 3D Icons | 3D图标的勾选框，控制图标是否以3D方式显示。3D方式和2D方式的区别在于：3D图标会有近大远小的透视性，以及会被遮挡。而2D图标会一直显示到界面最上层，且不会因距离远近而变化大小。  当3D图标勾选框被选中时，右侧的滑动条可以调节图标的整体大小。 |
| Show Grid | 是否显示网格线。网格线在制作场景时可以当作标尺使用，方便估计物体的位置。  这个选项是场景视图窗口特有的。 |
| Selection Outline | 被选中物体是否边缘高亮，默认是开启的。  这个选项是场景视图窗口特有的。 |
| Selection Wire | 是否显示选中物体的线框，默认是关闭的。  这个选项是场景视图窗口特有的。 |
| Scripts | 自定义组件的列表，也包含一些预制的组件。下面的几个自定义组件，都可以单独开启和关闭它们的图标、辅助线框显示。 |
| Built-in Components | 内置组件列表，下面的许许多多内置组件，都可以单独开启和关闭它们的图标、辅助线框显示。 |

### 辅助线框

辅助线框（Gizmoz）与场景中的游戏物体有关，某些线框只在物体被选中时显示，某些会一直显示出来。这些辅助线框通常都是程序生成的射线和线段，会根据当前视角实时变化。最常用和有用的线框是灯光和摄像机的，自定义的脚本也可以拥有定制的线框表示，用来直观展示某些参数，但是那属于比较高级的应用了。

某些辅助框线只能单纯查看，但是某些框线还能用来操作，比如说音源（AudioSource）范围的框线，就可以点击和拖拽，用来调节音源的范围。

常用工具中，移动、旋转、缩放工具都有各自的辅助线框，可以进行交互操作，在前文中已经提到过了。

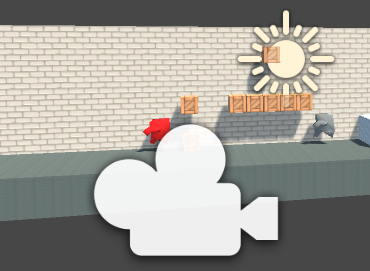


（摄像机和光源的辅助线框，它们都只在物体被选中时才显示出来）

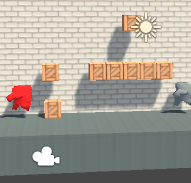
在脚本中可以通过实现OnDrawGizmoz方法来自定义辅助线框的展示，详细方法这里先略过。

### 辅助图标

除了辅助线框，游戏视图窗口和场景视图窗口中还会显示辅助图标。它们从外观上看是扁平的、广告牌风格的图标，覆盖在界面的最上面一层，通过它们你可以方便地看到一些没有外形的物体（比如摄像机和灯光本身是没有模型的）的大致位置。最常见的图标就是摄像机和灯光。和辅助线框一样，也可以在脚本中自定义辅助图标的外观。

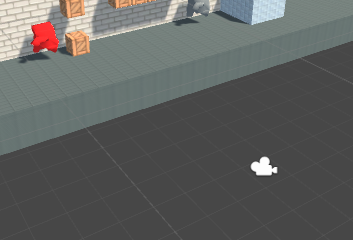


（默认的摄像机和灯光的辅助图标，3D图标大小可调整。）

（2D辅助图标，没有“近大远小”的透视效果。）

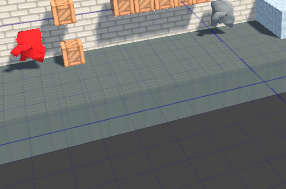
### 显示网格

显示网格（Show Grid）的选项用来控制是否在场景中显示辅助网格，如下图所示：



（开启辅助网格。）

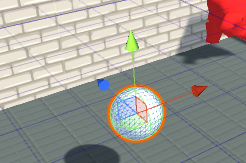
要改变网格的颜色，可以在主菜单中的Edit > Preferences > Colors中改变Grid的颜色，下图将Grid颜色改为蓝色，让线框显示更明显。



### 选中时高亮和选中框线

当“选中时高亮”（Selection Outline）被勾选时，被选中的物体会在边缘处出现橘色的描边：

当选择框线别启用时，在场景中或者层级窗口中选中物体以后，就会在物体上显示模型的框线了。



框线的颜色和高亮的颜色都可以在Edit > Preferences > Colors 中修改。

### 内置组件的显示

在辅助框线菜单中选中和取消内置组件的选择框或小图标，就可以控制辅助框线或辅助图标是否显示。

某些内置组件没有图标（比如刚体组件），所以在辅助框线菜单中也找不到它。

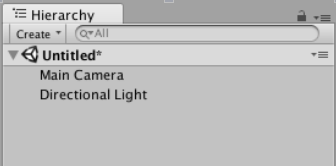
之前说过，除了内置组件，还有一些自定义脚本组件也会出现在菜单中。其中包含：

* 指定了图标的脚本。
* 实现了OnDrawGizmoz方法的脚本。
* 实现了OnDrawGizmozSelected方法的脚本。

某些类型的组件具有图标，某些具有辅助框线，某些组件二者都有。它们在菜单中会有相应的显示效果。

简单来说，点击图标就可以显示/隐藏该组件的图标，点击勾选框就可以显示/隐藏该组件的框线。只要简单尝试就可以理解该菜单的使用方法。

## 层级窗口（Hierarchy Window）



（默认的层级窗口外观）

层级窗口包含了当前场景中所有游戏物体（GameObject）的列表。其中某些游戏物体是独立的物体，而某些是预制体（Prefab）。任何新建或者删除物体的操作都会在层级窗口中反映出来。

默认情况下，层级窗口中的顺序是物体被创建的顺序，但是这个顺序是可以任意改变的，只需要通过拖拽的操作即可实现。层级窗口可以表示物体的父子（Parent和Child）关系，这也是层级窗口名称的由来。

### 父子关系（Parenting）

Unity具有一个非常重要的概念——父子关系。举例来说，先创建一个对象，然后再创建一系列对象，将后面这些对象置于第一个对象下级，这样第一个对象就称之为这一组对象的父物体（Parent object），而其它的对象可以称之为父物体的子物体（Child object或Children）。可以创建嵌套的父子关系，也就是说任何一个节点都可以拥有下一级子节点。

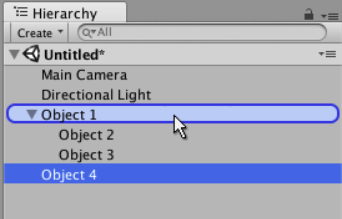


（上图中，Child和Child 2都是Parent物体的子物体。Child 3是Child 2的子物体，也是Parent的间接子物体（或者说子孙））

点击Parent物体左侧的三角形，就可以显示或隐藏它的子节点，和资源管理器中的文件夹类似。点击三角形只改变第一层的显示或隐藏状态，按住Alt键再点击可以递归显示或隐藏该物体的所有层级的子节点。

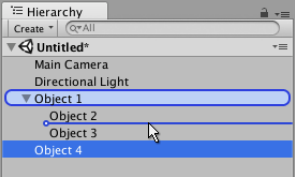
### 将物体设置为子物体

要让一个物体成为另一个物体的子物体，只需要先选中它，然后将它拖拽到另一个物体上即可，拖拽时会有明显的指示。



（上图中，Object 4是要拖拽的物体，将它拖拽到Object 1上即可，拖拽时会有胶囊形状的图标指示Object 1）

另外还有一种操作方法，可以让上图中的Object 4直接插入到Object 2和Object 3之间，同时也就成为了Object 1的子物体。只需要选中Object 4之后，拖拽它到Object 2和Object 3之间即可。



（上图中，Object 4被拖动到Object 2和Object 3之间，目的地用一个蓝色的横线表示插入。另外Object 1依然会用胶囊图标指示，说明这个操作会引起父子关系的变化）

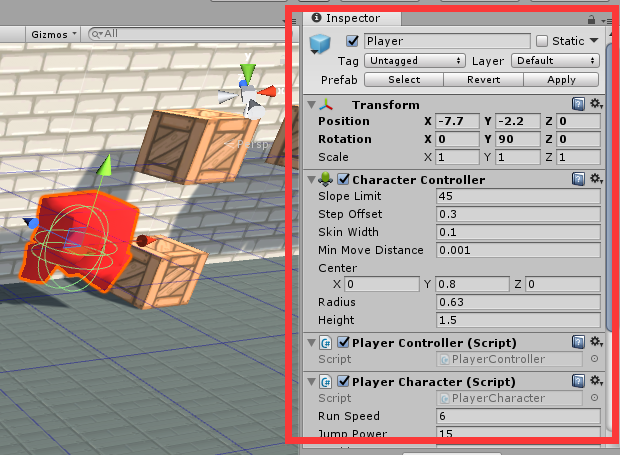
父子关系是一个很大的话题，比方说父物体的移动、旋转、缩放也会直接影响子物体，这个问题将在未来继续讨论。

### 同时编辑多个场景

直接拖拽另一个场景到到当前的层级窗口中，就可以启动多场景同时编辑模式。这种做法在同时处理两个场景时会比较有用，比方说复制一个物体到另一个场景里面。这个功能在之后的多场景管理章节中会单独讨论。

## 检视窗口（Inspector Window）

Unity的场景通常是由很多GameObject组成的，每个GameObject可能包含脚本、声音、模型等等多个组件。检视窗口（Inspector）显示了当前选中物体的细节信息，包括了GameObject所挂载的所有组件，而且还能在检视窗口中修改这些信息。



（默认的检视窗口）

### 检视物体和选项

检视窗口可以查看和修改Unity编辑器中的几乎所有东西的属性和设置，不仅对实体的物体比如GameObject、资源、材质有效，修改编辑器设置和预设选项的时候，也会用到Inspector。



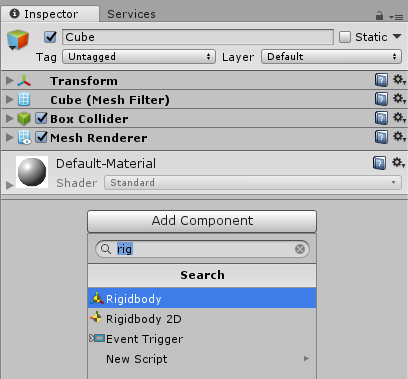
（典型的例子，用检视窗口查看带有摄像机组件的物体）

在层级窗口或场景视图中选中了物体以后，检视窗口就会显示当前物体中所有组件的信息。使用检视窗口可以编辑这些信息和设置。

上图的例子中，我们选中的是Main Camera物体。不仅包含物体的位置、旋转和缩放信息，很多其他信息也被显示并可以被编辑。

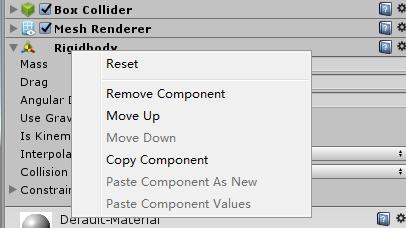
### 添加删除组件

点击检视窗口下方的Add Component按钮，就可以添加组件。点击后会显示一个各种组件的选择框。Unity目前包含的组件非常多，已经被分为很多组，可以分两步依次选择，也可以用附带的小搜索工具进行快速筛选，如下图：



（为物体添加Rigidbody组件）

删除组件更为简单，只需要在组件标题处点击鼠标右键，打开组件菜单，选择Remove Component即可删除。



### 复制组件或组件参数

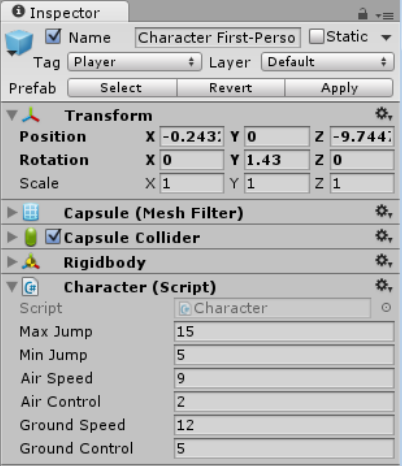
有时我们创建的组件需要复用另一个组件的参数，某些组件参数较多，一个个手工填写参数比较费时且容易填写错误。这时，我们可以在组件标题上点击鼠标右键打开菜单，选择Copy Component选项来复制组件的参数。

复制了组件参数之后，之后选中要操作的目标物体，有两种方法来复制组件属性：

1. 打开目标物体的任意一个组件菜单，选择Paste Component As New，这样就新建了一个组件且参数和复制的组件一致。
2. 打开目标物体的同类组件的菜单，选择Paste Component Values，这样不会新建组件，而会将原始组件的参数复制到同类型的目标组件上。

由于某些组件只允许存在一个，比如刚体组件（Rigibody），所以某些选项会是禁用状态。

### 查看脚本参数



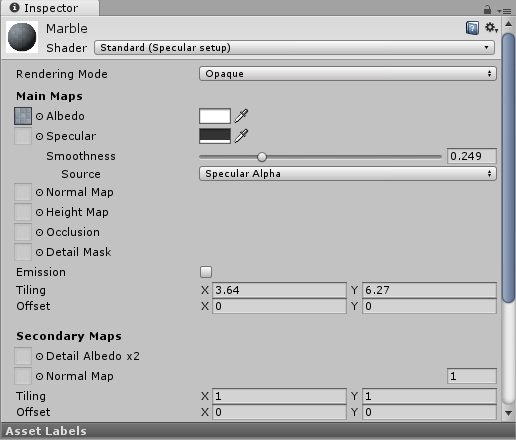
（脚本组件，可以修改其中一些字段的值）

当游戏物体挂载了自定义脚本，该脚本组件的部分字段（比如公共字段）是可以显示和编辑的。编辑他们的方法和编辑常规组件一样。这意味着可以方便地修改自定义组件的参数和属性，而不需要去修改脚本代码。

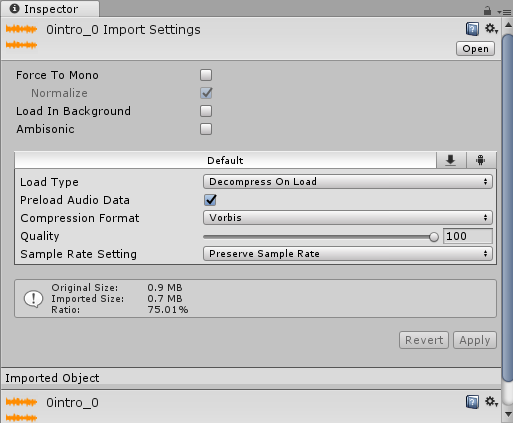
### 查看素材

当在工程窗口中选中了一个资源时，检视窗口也会显示该资源的设置和参数，这些设置影响了该资源如何被导入，以及在运行时会产生什么具体效果。

每一种类型的资源的参数和设置都不相同。比方说下面的查看材质和查看音频资源，参数和设置就完全不同。

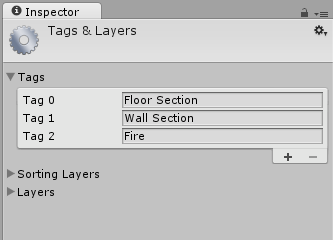


（检视窗口中正在查看一个材质）



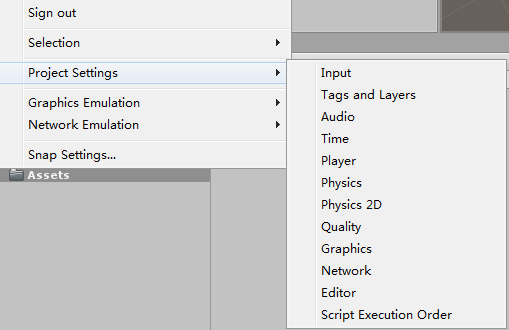
（检视窗口中正在查看一份音频文件的设置）

### 工程设置



（在检视窗口中，正在查看Tag和Layer的设置）

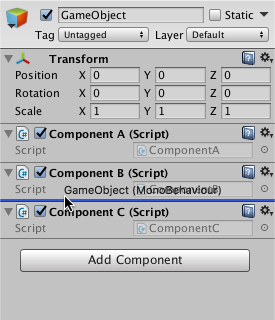
查看和修改工程设置也会用到检视窗口，比如在菜单中选择Edit > Project Settings下面的多个选项，就会在检视窗口中显示相应的工程设置了。



如上图，有许多改变工程基本参数的设置，例如输入设置（Input），Tag和Layer设置，音频设置，时间设置，物理设置，等等。时间设置可以改变游戏运行的帧率，物理设置可以改变重力加速度的数值，这些工程设置会对整个工程中的所有相关功能造成影响。

### 修改组件的顺序

要改变检视窗口中组件的顺序，只需要拖拽要改变的组件的名称位置到目的地即可。操作过程中会有明显的蓝色标记提示，可以很清楚的看到组件从哪里移动到哪里。

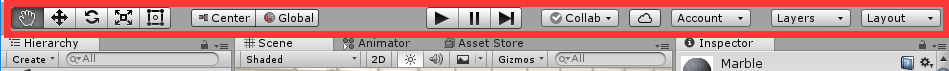


（通过拖拽操作修改脚本组件的顺序）

有几点值得说明：

1. 只能修改一个游戏物体中组件的顺序，不能直接将组件从一个物体拖拽到另一个物体，
2. 可以将脚本文件直接拖拽到检视窗口中，自动新建一个脚本组件。
3. 当同时选中多个游戏物体时，检视窗口中会显示所有物体所共有的组件。这时候同时改变这些物体中组件的顺序，也是可行的。
4. 物体上挂载组件的顺序是真实存在的，比如在脚本中获取组件的时候，这些组件就会以这个顺序获取到。典型的情况出现在物体上同时挂载了多个同类型组件的时候。

## 工具栏（Toolbar）



上图即是工具栏所在的位置，每个按钮控制的是不同的模块，但都是比较常用的操作。

IMG_257

变换工具组——在场景视图中用到。

IMG_258

变换工具相关功能——在使用变换工具时发挥作用

IMG_259

开始运行/暂停/单步运行按钮，直接控制工程的运行。

IMG_260

云服务按钮——打开Unity的服务（Service）窗口。

IMG_261 .

账户下拉菜单——用于登录和查看当前的Unity帐号。

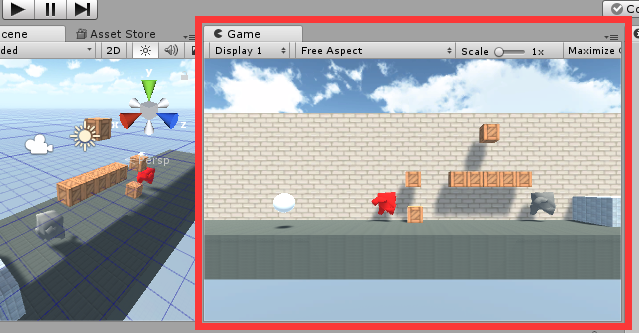
IMG_262

Layers下拉菜单——选择在场景视图中显示哪些层（Layer）的物体。

IMG_263

布局下拉菜单——控制所有窗口的布局。

## 游戏视图窗口（Game View）



游戏视图显示的是实际游戏中看到的画面。对初学者来说，关键是要知道它所渲染的内容是从场景中的摄像机中获得的。初始场景中只有一个Main Camera相机，所以游戏视图中显示的就是Main Camera所看到的画面。许多时候我们会用到多个相机，它们的切换、叠加关系是摄像机组件的章节中所要讨论的内容。

### 播放和暂停

IMG_257

以上三个按钮分别是工具栏中的开始播放、暂停和单步执行按钮，它直接控制了游戏的运行状态，所以和游戏视图是密切相关的。我们在开发中每天都需要用到很多次播放、暂停的功能，所以下面展开探讨一下。

#### 运行状态

当按下播放按钮以后，Unity进入运行状态。游戏视图中所显示的画面，大致就是最终发布后用户看到的画面（细节不完全一致）。有几个要点：



1. 工具栏中的播放按钮外观改变，且编辑器窗口整体会变暗，以提示我们正处于运行状态。
2. 在播放状态下，我们依然可以进行绝大部分的编辑操作，比如在场景视图中移动物体位置，在检视窗口中添加删除物体的组件等等。但是，务必注意所有播放过程中的改动，在退出运行模式之后，都会回到原始状态。
3. 再次按下播放键就停止播放，回到正常编辑模式。

#### 暂停状态



在播放模式下按下暂停键，就可以进入暂停状态。暂停状态的本质依然是运行状态，只是游戏的时间被暂停，也不会触发事件。所以暂停状态下的修改也不会被保存。在这个状态下方便我们查看游戏瞬间的状态，比如物体的位置、当时的参数信息等等。以下是一些要注意的：

1. 除了在播放状态下按下暂停键，还可以先按暂停键，再按播放键，帮助我们在游戏开始的一瞬间就暂停，专门调试游戏一开始运行立即就会产生的问题。
2. 可以随时再次按下暂停键，在暂停和播放之间切换。
3. 暂停状态下，按下暂停右边的单步调试按钮，就可以让游戏前进一帧然后再暂停，专门调试一些时间点要求非常精确的问题。
4. 暂停只在游戏逻辑帧的间歇起作用，如果由于脚本原因，游戏逻辑死循环、因同步而停住等情况，按暂停键会无效。这种方法也可以用来判断游戏是否出现卡住等严重问题。

### 游戏视图的工具条



下面是对工具条中每个选项功能的具体介绍：

|  |  |
| --- | --- |
| **按钮** | **功能** |
| Display | 如果当前有多于一个摄像机，这个选项可以选择具体哪一个摄像机用于渲染。在摄像机组件中可以选择摄像机对应的显示序号，也就是目标显示对象（Target Display）。 |
| Aspect | 显示器大小和比例，不同的显示设备差异很大（小屏幕手机、大屏手机、平板、桌面显示器等等），所以这个选项有助于测试各种情况下的显示效果。默认的Free Aspect是自由模式，自动根据窗口大小调整显示器参数。  除了Free Aspect，Unity还内置了许多标准分辨率的设置（比如WVGA等等），另外还可以自定义分辨率的大小，甚至还能只定义比例，不定义具体分辨率（例如有许多安卓设备都是16:9分辨率但是分辨率不同）。 |
| Scale slider | 缩放滑动条，该滑动条只是在游戏视图中缩放，方便查看细节或整体，并不影响实际的屏幕分辨率。 |
| Maximize on Play | 该按钮为两态按钮，当处于按下状态时，一开始播放游戏游戏视图就会最大化，方便预览游戏效果。当按钮处于弹起状态时不起作用。 |
| Mute audio | 该按钮为两态按钮，控制是否屏蔽音频。 |
| Stats | 该按钮为两态按钮，开启时，会在游戏视图上叠加一层统计信息。这个功能在监控游戏性能时非常有用，可以帮助开发者及早发现潜在的性能问题。 |
| Gizmos | 显示或隐藏Gizmoz，内含详细的Gizmoz设置。Gizmoz只能在开发时看到，并不会出现在最终游戏的发布版中。详见相关章节。 |

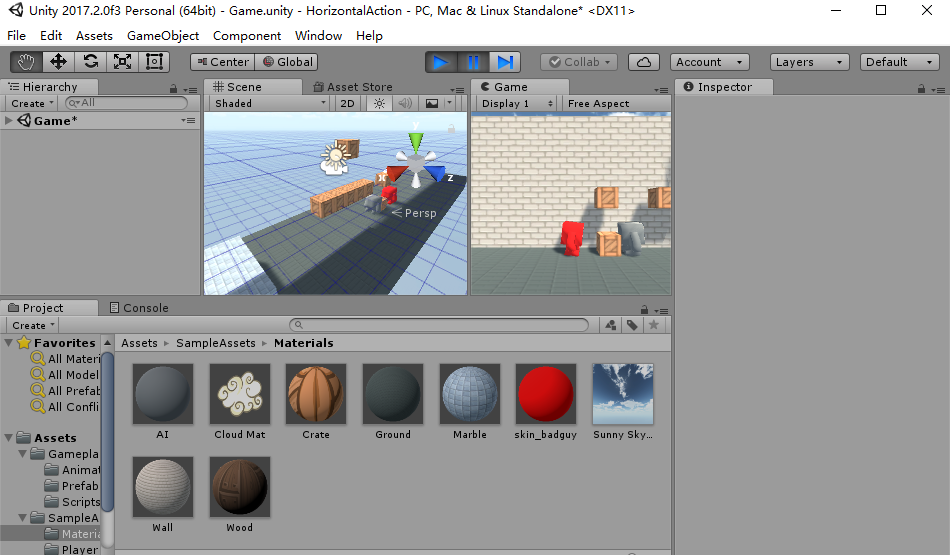
**关于Gizmoz菜单：**

和场景视图一样，游戏视图也包含完整的辅助线框（Gizmoz）菜单。它的用法和功能与场景视图中的类似，不再赘述。

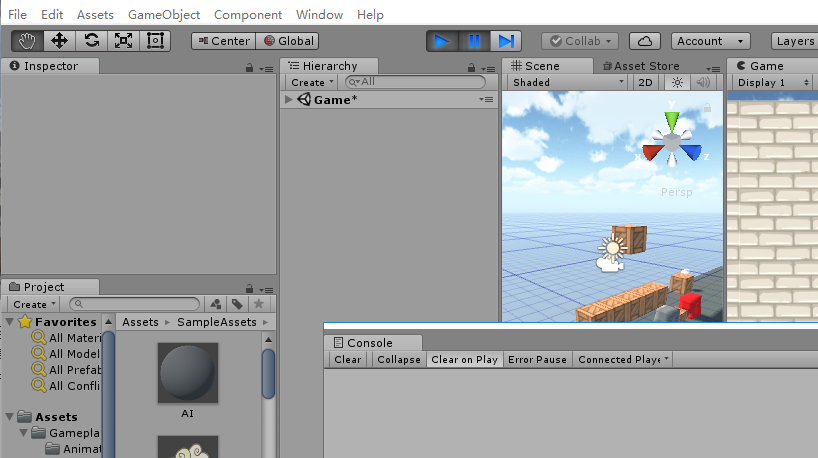
### 自定义Unity开发环境

我们可以自定义Unity编辑器的窗口布局，让它更符合自己的操作习惯。首先，可以拖动任意一个窗口左上角的标签到任意位置。其次，当拖到另一个窗口的标签处时，可以和目标窗口共用一个区域，形成多标签页的形式。最后，当拖动窗口到可停靠（Dock）的位置时，编辑器就会划出一片区域专门放置该窗口，原来占据位置的窗口就会相应缩小。

窗口可以浮动移动，可以停靠在现有窗口的侧面、顶部或底部，也可以以标签页的形式和另一个窗口共用一块位置。Unity的窗口布局非常灵活，不同的人也有不同的设置方法。总的来说，只要自己看着舒服、方便个人使用，就可以任意布局。特别在使用超宽屏显示器或多显示器时，可以采用的布局的方式就会更灵活了。

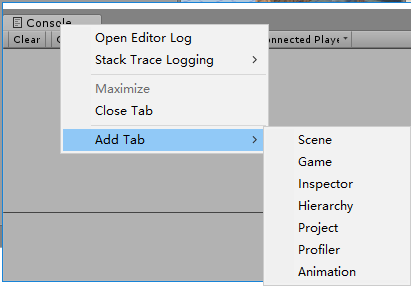


当我们将布局调整到合适的位置以后，就可以选择保存当前布局，未来可以随时将它还原。保存的方法是在主工具栏（Unity上方）的右侧，点击Layout下拉菜单，选择其中的Save Layout即可。只需要为当前布局取一个名字，之后就可以随时从Layout下拉菜单中还原它了。



（上图是一个完全自定义的布局）

大部分窗口中，都可以右键点击窗口标签，然后选择Add Tab来添加一个标签页。用这种方式也可以打开关闭的窗口，比如之前将场景视图窗口关闭了，可以在这里重新打开。



最后，Unity还提供了几种默认布局，适用于某些典型的应用场景，可供参考。

要使用预置布局，请在菜单栏Window > Layouts 里查找。典型的布局有：

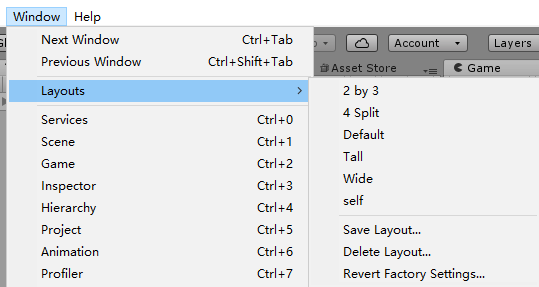
2 by 3：很好用的常规布局，可以同时看到场景视图和游戏视图。

4 Split：四个场景窗口，和3D软件一样方便看到场景的标准四视图。

Default：默认布局，场景和游戏视图在同一个窗口内，适合较小的显示器，也很常用。

Tall、Wide：这两种布局下，场景视图分别是较高和较宽的。

Self：少量窗口配合多个标签页的布局。



## Unity常用快捷键

下面将总体介绍Unity中的快捷键，完整的文档请参考Unity网站上的官方文档。下面所说的Ctrl/Cmd键在Windows下代表Ctrl键，在MacOS系统下代表Command键。

|  |  |
| --- | --- |
| **工具栏** |  |
| Q | 小手工具 |
| W | 移动工具 |
| E | 旋转工具 |
| R | 缩放工具 |
| T | RectTransform工具 |
| Z | 基准点模式切换 |
| X | 世界坐标系/本地坐标系切换 |
| V | 顶点吸附模式 |
| CTRL/CMD+LMB | 单位吸附模式 |

|  |  |
| --- | --- |
| **游戏物体GameObject** |  |
| Ctrl/Cmd+Shift+N | 新建物体 |
| Alt+Shift+N | 新建物体，作为当前选中物体的子物体 |
| Ctrl/Cmd+Alt+F | 让场景视图移动到该物体上 |
| Shift+F 或 双击F | 让场景视图移动到该物体上并一直保持查看该物体 |

|  |  |
| --- | --- |
| **选择窗口** |  |
| Ctrl/Cmd+1 | 场景视图窗口 |
| Ctrl/Cmd+2 | 游戏视图窗口 |
| Ctrl/Cmd+3 | 检视窗口 |
| Ctrl/Cmd+4 | 层级窗口 |
| Ctrl/Cmd+5 | 工程窗口 |
| Ctrl/Cmd+6 | 动画窗口 |
| Ctrl/Cmd+7 | 性能分析窗口 |
| Ctrl/Cmd+9 | 资源商店窗口 |
| Ctrl/Cmd+0 | 版本控制窗口 |
| Ctrl/Cmd+Shift+C | 控制台窗口 |

|  |  |
| --- | --- |
| **编辑** |  |
| Ctrl/Cmd+Z | 回退到上一步 |
| Ctrl+Y (Windows only) | Windows下的重复操作（与回退操作相反） |
| Cmd+Shift+Z (Mac only) | MacOS下的重复操作（与回退操作相反） |
| Ctrl/Cmd+X | 剪切 |
| Ctrl/Cmd+C | 复制 |
| Ctrl/Cmd+V | 粘贴 |
| Ctrl/Cmd+D | 直接复制，相当于复制并粘贴 |
| Shift+Del | 删除 |
| Ctrl/Cmd+F | 查找 |
| Ctrl/Cmd+A | 全选 |
| Ctrl/Cmd+P | 开始/停止运行游戏 |
| Ctrl/Cmd+Shift+P | 暂停游戏 |
| Ctrl/Cmd+Alt+P | 单步调试游戏 |

|  |  |
| --- | --- |
| **选择** |  |
| Ctrl/Cmd+Shift+1 | 读取选择范围1。数字可以是从1到9分别是9个栏位 |
| Ctrl/Cmd+Alt+1 | 保存选择范围1。数字可以是从1到9分别是9个栏位 |

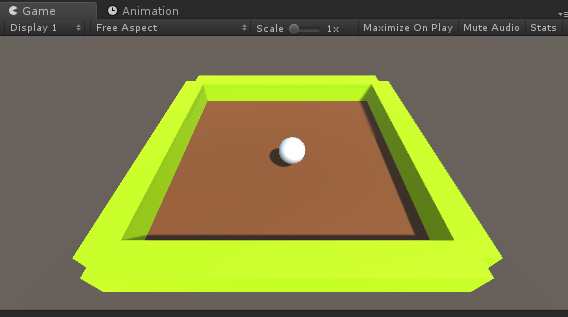
|  |  |
| --- | --- |
| **资源操作** |  |
| Ctrl/Cmd+R | 刷新 |

注意：以下快捷键用在动画（Animation）窗口中。

|  |  |
| --- | --- |
| **动画** |  |
| Shift+逗号 | 第一个关键帧 |
| Shift+K | 修改当前关键帧 |
| K | 设置或添加关键帧 |
| Shift+句号 | 最后一个关键帧 |
| 句号 | 下一帧 |
| Alt+句号 | 下一个关键帧 |
| 空格 | 播放动画 |
| 逗号 | 前一帧 |
| Alt+逗号 | 上一个关键帧 |

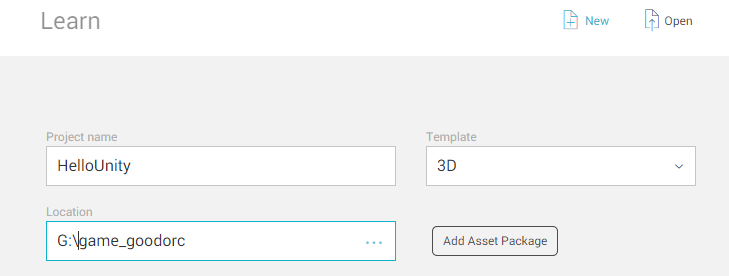
## 动手搭建游戏场景

接下来，我们搭建一个最简单的游戏场景，一方面可以熟悉Unity的基本操作，另一方面也为之后的学习做准备。我们要做的场景非常非常简单，如下图：

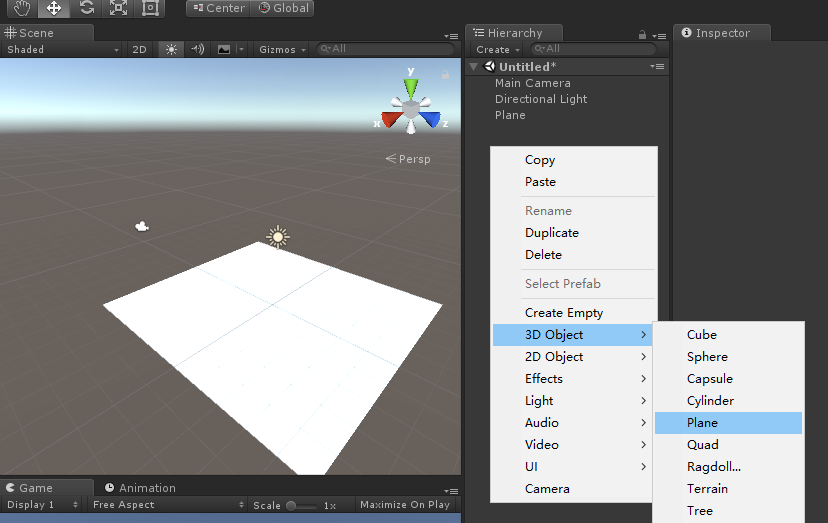


1. **创建工程**

打开Unity并创建一个工程HelloUnity。



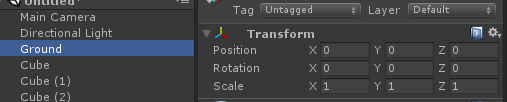
1. **添加地板**



在层级窗口（Hierarchy）中点击右键，选择3D Object > Plane，就可以新建一个平面。平面适合作为简单游戏的地板。注意：

* 如果在创建之前选中了某个物体，则创建的物体会成为子物体。如果出现这种情况，只需要在层级窗口中拖动物体重新调整父子关系即可，见本章1.6节。
* 新创建的物体名为Plane，将其改名为Ground是一种很好的习惯。否则物体多了不利于查看和查找。重命名物体类似于重命名文件，有多种操作，比如可以在右键菜单中选择Rename（重命名），或者选中物体按下F2键，还可以再次点击名称。

确保平面层次正确之后，在检视窗口中设置其位置为(0, 0, 0)。



将地板放在坐标原点有助于我们以后计算坐标。

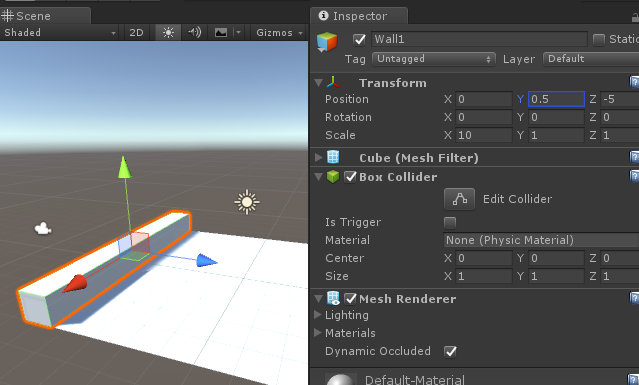
这步操作以及接下来的操作中，读者会发现实际上经常需要调整查看场景的角度。也就是说无论要制作什么场景，浏览场景的操作是最频繁使用的。关于浏览场景的基本操作可以查看本书1.6节，对照阅读效果更佳。

1. **添加第一道围墙**

与添加地板类似，添加4道围墙，用Cube（立方体）即可。

方便起见可以先只制作一个，之后可以再复制。将Cube命名为Wall1。

通过设置正方体缩放的某个维度，可以拉伸正方体成长条状，这里我们拉伸X方向10倍左右。然后再摆放正方体的位置到平面的一侧，作为围墙。

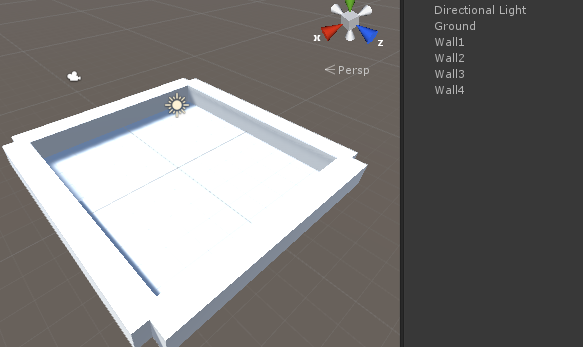


正方体的参考位置和缩放如上图，用鼠标直接拖拽位置很不精确，可以在检视窗口中直接调节数值。

1. **其它三道围墙**

有了第一个围墙，其它三个围墙也是同样的操作。实际工作中有很多方法可以快速搭建这个场景。首先，和第一道围墙相对的围墙，可以选中第一道墙后Ctrl+D复制（Duplicate），然后设定位置、改名为Wall2即可。

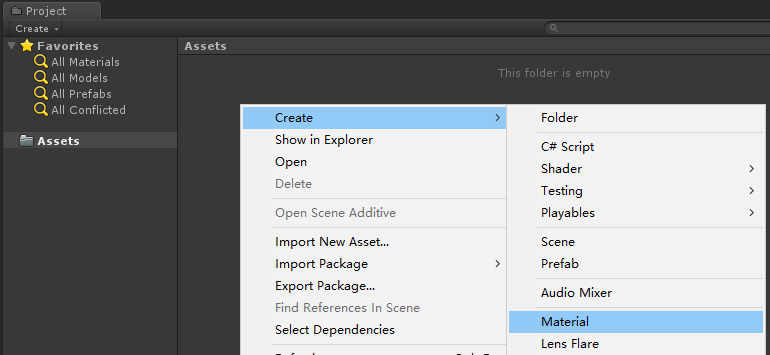
其它两道围墙有两种做法，一是将第一道围墙沿Y轴旋转90度；二是重新创建一个方块并拉伸Z轴。这两种方法都可以达到目的。对于简单游戏来说通常更喜欢不旋转的方法，因为一旦加入了旋转，坐标系就变化了，让问题变得复杂。但是在这个简单的例子中没有太多问题，根据个人喜好来制作即可，如果以后想要改变方法，重新制作也不难。



四道墙搭建好之后的效果如上图所示。可以通过改变围墙的长度或位置让四个角更好看一些。

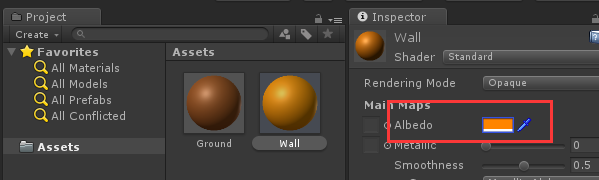
1. **修改物体的颜色**

物体的默认材质是白色的基础材质，要改变默认材质不能直接修改颜色，而要先替换为新的材质，才能修改。首先在工程窗口任意目录新建一个材质文件，在某个资源目录下Create > Material即可新建材质。



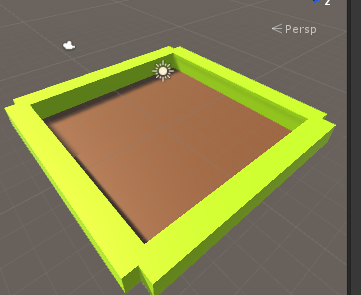
同理，一共需要两个材质，分别将材质文件命名为Ground和Wall，表示地面和墙体的材质。

之后修改材质颜色，只要选中材质文件后在检视窗口中操作即可。



修改检视窗口中Albedo（固有色）的颜色即可。这里甚至可以为Albedo指定一张贴图，实现带有图案的地板和墙面。

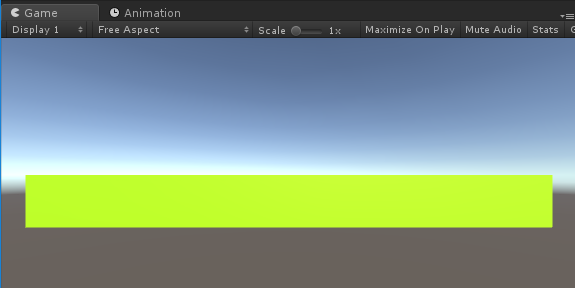
之后用这两个材质替换墙和地板的材质。最简单的方法只需要将材质拖拽到场景视图中的物体上即可；第二种方法是选中物体，将材质拖拽到物体的检视窗口里（要拖到最下面空白部分才可以）。这两种方法都会把默认材质替换为独立的材质文件，并不会添加新的组件。顺便说一句：材质是网格渲染器（Mesh Renderer）组件的参数。



如图，调整后我们有了彩色的地板和墙，方便看到效果。

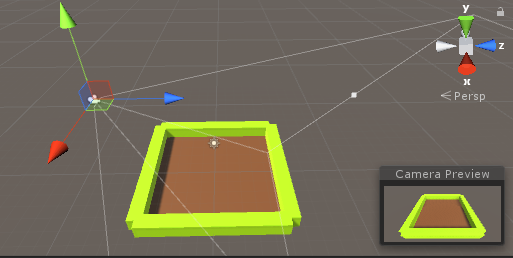
1. **调整摄像机**

虽然目前在场景视图中，看似很美好，但在游戏视图中看到的可能是这样的：

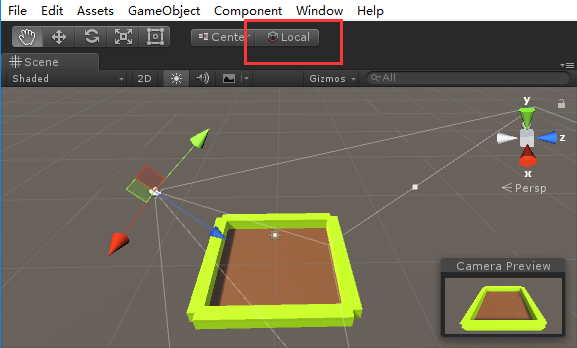


这是因为默认的摄像机是平视向Z轴方向的，而不是向下的角度。接下来调整摄像机，到一个合适的位置。

首先，用旋转工具将摄像机向下旋转45度左右。然后再调节它的位置，先向上移动，，再适当前进，让地板出现在游戏窗口的视野中。这里查看场景有两种方式，一方面，摄像机的预览窗口会出现在场景视图中；但是更推荐的办法是用默认的2 by 3布局，同时看到场景视图和游戏视图，边看效果边调节摄像机。



默认的工具坐标系是Global的，调节摄像机位置是沿着世界坐标的Y轴和Z轴移动。实际上这里有一种更合适的方式，将工具坐标系切换为Local：

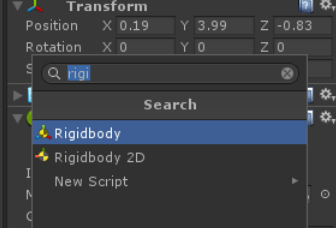


使用本地坐标系之后，你会发现摄像机的Z轴可以更直接地调节从摄像机到地板的距离。往往在搭建场景时需要根据需要反复切换工具的坐标系。

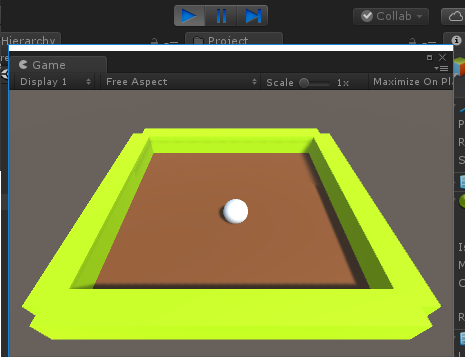
1. **添加一个小球**

创建一个球体并放在合适的位置上，步骤不再赘述。

有趣的是，你可以把小球适当放高一些，然后加上刚体（Rigidbody）组件。只需要选中小球，在检视窗口中点击最下面的Add Component按钮。然后搜索rigid或者选择Physics > Rigidbody即可。



默认刚体是开启了重力的，所以在播放游戏后，小球会落在地面上。



### 2D与3D工程的区别

用Unity开发2D或3D游戏都同样适合。在刚开始创建工程的时候，你可以选择2D或3D模式。也许你已经想好了要建立哪种模式的工程，这里再次对2D和3D项目做一些说明，以加深对二者的理解。

创建2D或3D工程主要是对一些初始设置有影响，比如当导入图片资源时，默认为材质（textures）还是精灵（Sprites）。可以随时将项目在2D或3D之间切换。以下再次对典型的2D和3D游戏做一些介绍。

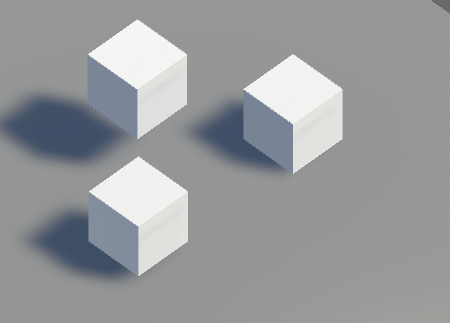
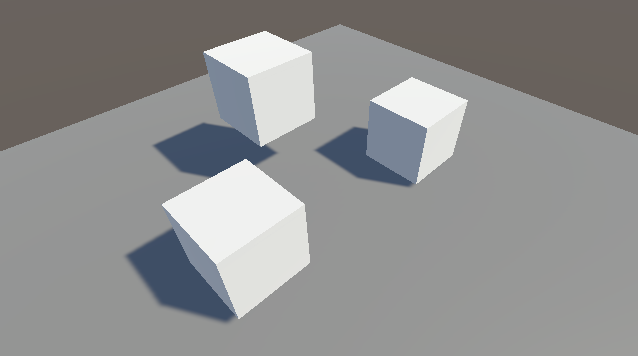
#### 全3D游戏



（一个场景简洁的3D游戏）

全3D游戏通常使用带有模型和材质的几何体来搭建游戏中的场景、角色与其他物体。在全3D游戏中，摄像机往往可以任意移动，不被限制在某个轴或者平面上；灯光与阴影也会用拟真的方式表现。3D游戏通常都采用透视摄像机，即近大远小的效果。

#### 正交3D游戏



（左图为透视摄像机效果，右图为正交摄像机效果。观察同一个简单场景）

某些游戏也是使用3D模型，但是使用正交摄像机代替透视摄像机。正交摄像机没有近大远小的效果，适用于很多鸟瞰视角的游戏，比如表现一个卡通城市，有时这种游戏被称作“2.5D”的。对这种游戏你也应当使用Unity的3D模式创建，只是在创建之后，将摄像机和场景视图（Scene view）改为正交的（Orthographic）即可。

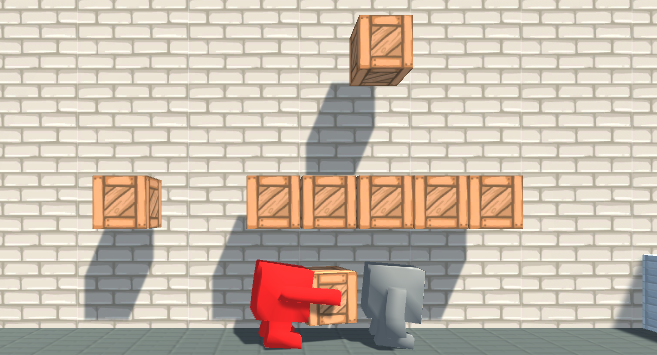
#### 全2D游戏



（纯2D游戏的例子，来自官方示例的截图）

许多2D游戏都使用扁平的图像（有时称之为精灵）来表现游戏画面。这种游戏的摄像机一般都是正交的。对这种游戏来说应当使用Unity的2D模式来制作。

#### 具有3D画面的2D游戏



（一个具有2D玩法，3D画面表现的游戏）

某些游戏使用了3D的角色和场景，但是游戏玩法却依然限制在2D范围内。比方说摄像机始终对准人物侧面，主角也只能左右移动，但是依然使用3D场景和角色，摄像机也采用透视摄像机。对这种游戏来说，3D效果只是一种为了增强表现力的手法，而不是为了实现游戏玩法。这种类型的游戏有时也会被称为“2.5D”。尽管游戏玩法是2D的，但是物体还是有深度效果，在Unity中应当使用3D模式创建。

#### 2D玩法和画面，透视摄像机

这是另一种流行风格的2D游戏，使用2D场景加上透视摄像机，来实现一种多重卷轴的效果。这种游戏中所有的物体都是扁平的，但是离摄像机有着不同的距离。这种游戏完全适合用Unity的2D模式创建，只要在创建以后讲摄像机和场景视图（Scene view）设置为透视的（Perspective）即可。

## 总结

本节的主题是“初识Unity”。我们从Unity的安装开始，初步介绍了Unity编辑的主要组成部分，包括场景视图、工程视图、检视窗口等几大窗口。其中场景窗口的基本使用方法是学习的重点。相信看完本章之后，读者已经安装好了Unity引擎并且开始熟悉它了。

本节的最后补充了一个搭建简单场景的例子，这个例子本身用到的知识点不多，但是作者希望能通过一个非常简单的例子让读者养成动手实践的习惯。书籍作为静态的印刷品，很难平衡“知识点的完备性”和“易于实践”这两点。希望读者能自己平衡好阅读、查询资料、动手实践等等几方面的平衡，更好地使用本书。