# 第3章 资源工作流程

本章将介绍在Unity引擎中使用资源的基本方法。

资源是在您游戏或者项目中使用的任何内容表示。 资源可能来自于Unity引擎之外创建的文件，例如3D模型，音频文件，图像或Unity支持的任何其他类型的文件。 Unity中还可以创建一些资产类型，例如Animator控制器，音频混合器或渲染贴图。

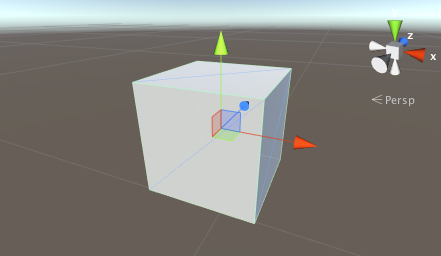


一些可以被导进Unity的资源类型

## 内置的基础物体

Unity可以使用3D建模工具创建的任意形状的3D模型。而且Unity也有许多自带的可以直接生成的原始物体，例如**立方体**，**球体**，**胶囊体**，**圆柱体**，**平面**和**四边形**。这些物体自身通常都很常用（例如平面就常常被用来做成平坦的地表），但是它们也可以用来当成替代品和做出原型以此来提供测试。这些物体可以通过菜单**GameObject -> 3D Object**上选择相应物体添加到场景中。

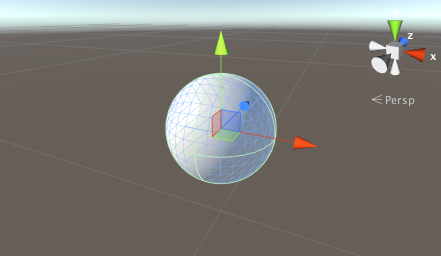
### 立方体



立方体

这是一个边长为1单位的简单立方体，如果给它进行贴图的话，贴图纹理会重复出现在每一个面上。一个标准的立方体在许多游戏中并不是会经常使用。但是只要改变它的大小，它就可以变成墙壁、柱子、箱子，台阶和其他类似的物体。当成品模型还没制作出来时，它也在开发时被程序员当成一个便捷的替代品。例如一辆汽车就可以用大致相似的尺度的细长的立方体盒子代替。尽管这不适合最终的游戏，但是作为测试车辆控制代码的代替品就很合适了。既然标准立方体的边长是一单位，那么您也可以添加立方体到场景中去，用它来检验导入场景的网格的比例是否正确。

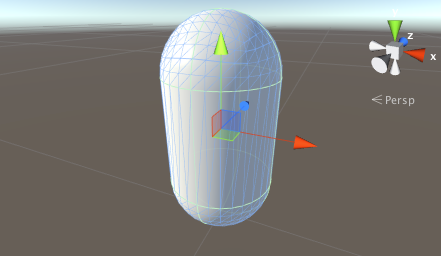
### 球体



球体

这是一个单位直径的球体（半径为0.5单位），如果贴图的话，整个贴图纹理会依据上下两个极点环绕包裹整个球体表面。球体可以作为各种球，行星，炮弹。但是半透明的球体也可以制作为不同半径很漂亮的GUI设备。

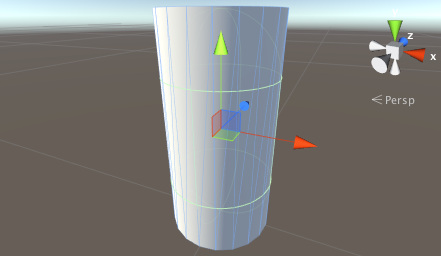
### 胶囊体



胶囊体

一个胶囊体是中间一个柱体和两端半球体的组合。整个物体的大小是直径为一单位长度，高度为两单位长度（中间柱体的高是一单位长度，两个半球是半径为0.5单位长度）。它被贴图的话，贴图纹理会依据上下两端半球的顶端，收缩包裹整个胶囊体。现实中这个形状的物体并不多见，但胶囊体却是一个在原型中很实用的替代品。特别是对一些工作来说，使用了胶囊体物理效果表现要比立方体更好。

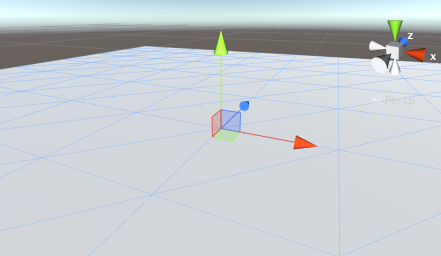
### 柱体



柱体

这是一个直径为一单位长度，高度为两单位长度的简单柱体，被贴图的话，贴图纹理会覆盖到柱体的表面，包括上下两个底面。柱体创建柱子，杆，轮子这些物体十分方便，但是您需要注意，碰撞盒子的形状要选择胶囊体（Unity并没有柱体形状的碰撞盒子）。如果您想要在一些物理实验中需要准确的柱体碰撞盒子，您需要在建模工具中创建一个合适形状的网格，并把它导入到网格碰撞体中。

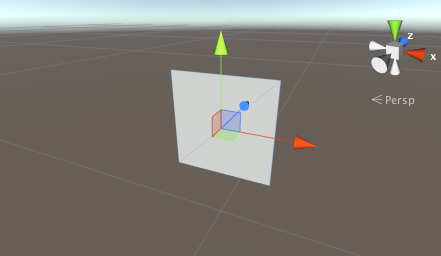
### 平面



平面

这是一个边长为十个单位的，且只能在本地坐标空间XZ平面中调整的平面正方形。被贴图的话，那么整张贴图纹理都会出现在这个正方形上。平面可以适用于很多类型的扁平表面，比如说地板墙壁。一个处在GUI中的表面在某些时候还需要显示一些图片和电影等特殊效果。尽管一个平面可以这么用，但更简单的物体**四边形**在这些问题上却更适合。

### 四边形



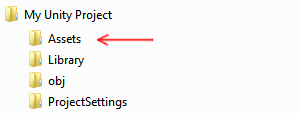
四边形

四边形这个基础物体类似于上文介绍的平面，但是四边形的边长只有一单位长度，并且表面只能在本地坐标系的XY平面调整。此外一个四边形只可以被分成两个三角形，但是一个平面可以被分成两百个三角形。当场景中的物体必须要显示图片和视频时，那么四边形就很合适了。简单的GUI和信息显示都可以用四边形实现，还有在远处显示粒子、图片精灵、伪图片也是可以使用四边形实现。

## 资源导入

在Unity引擎之外创建的资源有两种导入到Unity的方法，一是直接保存到工程的“Asset”目录，或者直接复制到相应的目录。对于许多常见格式的资源文件直接保存到“Asset”文件夹，Unity都能识别。当其内容改变时，Unity会读取它，并且编辑器界面将会提示您正在重新导入资源。

您创建的Unity工程实际上是一个您项目命名文件夹,在此目录下面有这些子文件夹：

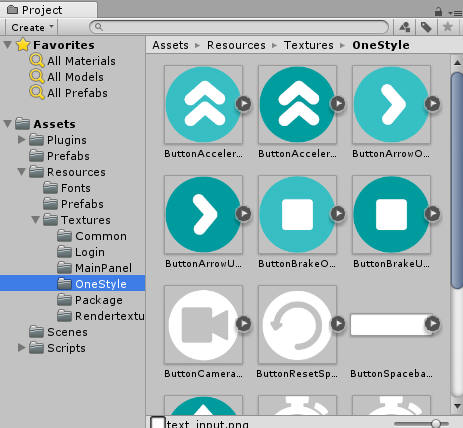


Unity项目的基本文件结构

资源文件夹是您保存或者复制项目所需文件的地方。

Unity项目窗口的内容就是显示您的资源文件夹内的所有东西。所以如果您保存或者复制了文件到您的资源文件夹，它会重新导入，并且显示在您的项目窗口上。

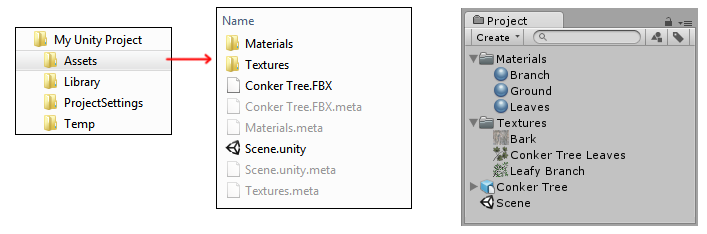
Unity会自动检测您添加到资源文件夹的文件，或者是修改了的文件。当您放一些资源到资源文件夹时，您将发现这些资源自动出现在您的项目窗口中。



被导入到项目中的项目资源

如果您拖一个文件（例如冲Mac的finder或者从Windows的资源管理器）到项目窗口中，它会被复制到资源文件夹内，然后出现在项目窗口中。

出现在您项目窗口的文件代表了（大多数情况下）在您电脑中存在的真实文件，如果您从Unity里删除了它们，也删除了电脑里的本地文件。



电脑上资源目录和在Unity编辑器内项目窗口的关系

上图表示了Unity项目资源文件夹里的一些文件和文件夹的一个例子。您可以创建相关的文件夹来使您的项目看起来井井有条更加组织化。

您会发现上图中存在于文件系统中的.meta文件，但是却没出现在您的Unity里的项目窗口中。Unity为每一个资源和文件夹创建一个.meta文件，并且默认隐藏它们。所以您在文件浏览器也看不见它们。

这些.meta文件包含了相关资源在项目中如何使用的重要的信息，并且必须和同名文件放在同一个路径下。所以如果您在文件浏览器中移动或重命名了这些文件，您也要对.meta文件执行相同的操作。

操作这些文件最好的方法就是在Unity编辑器下操作它们。这样的话，Unity将会自动移动或重命名相对应的.meta文件。

如果您想往您的项目工程中添加一组资源，您可以使用资源包Asset Packages。

### 一些常见资源类型

#### 图片文件

Unity支持大部分常见的图片格式，例如BMP、TIF, TGA, JPG, 和PSD格式。如果您直接以.psd格式（Photoshop）保存到您的资源文件夹，它们会被导入成平面图像。

#### 3D模型文件

如果您从常见的主流3D建模软件以它们的自有格式（如.max，.blend，.mb，.ma）导入3D模型到您的Unity资源目录下，那么Unity会回调3D软件的FBX导出插件，使它们以FBX格式导入，当然您也可以手动将它们以FBX格式从您的3D建模软件导出到Unity中去。

#### 网格和动画

无论您使用哪一个3D包，Unity都会从包中导入网格和动画。

您的网格文件不需要导入动画。如果您需要动画的话，那么您可以选择从一个单独的文件导入所有的动画，或者导入单独的只有一个动画的网格文件。

#### 音频文件

如果您在资源目录下导入未压缩的音频文件，它们将会被自动以压缩导入。

#### 其他资源类型

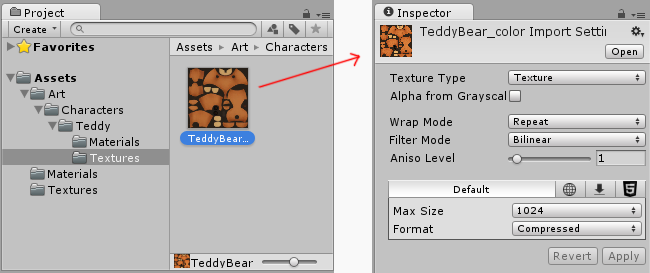
在任何的Unity项目里，您可以对资源进行压缩修改或其他操作，但Unity不会修改您的原始源文件。导入资源操作会读取您的资源文件夹，同时在内部对资源进行一个准备就绪的标识，并且您选择的导入设置进行相关设置。如果您修改了资源文件的导入设置，或改变了资源文件夹下的源文件，Unity会将其重新导入。

注意：导入3D建模软件的自有3D格式的文件，需要在安装了Unity的电脑上同时安装这个3D软件。因为Unity需要使用3D软件的FBX导出插件来读取这个文件。您也可以选择直接从3D软件中直接导出为FBX文件然后保存到项目目录中。

## 资源导入设置

每一种Unity支持的资源类型都有相关的导入设置，这些设置将会影响资源在Unity内的显示或行为。在项目窗口内选中资源，即可在检视窗口查看资源的导入设置，其所显示的选项将会因选中的资源类型的不同而有所不同。

例如图像的导入设置可以让您选择是以贴图，2D精灵图片或法线贴图的形式导入。FBX文件的导入可以让您调整模型的比例大小，生成法线或光照贴图坐标，还可以分割裁剪FBX中的动画片段。



在项目窗口内选中图像资源，检视窗口显示相关导入设置

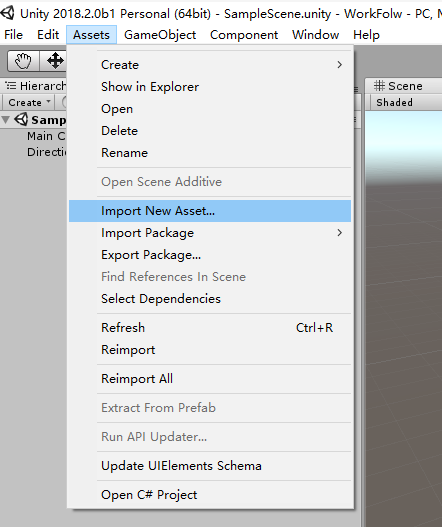
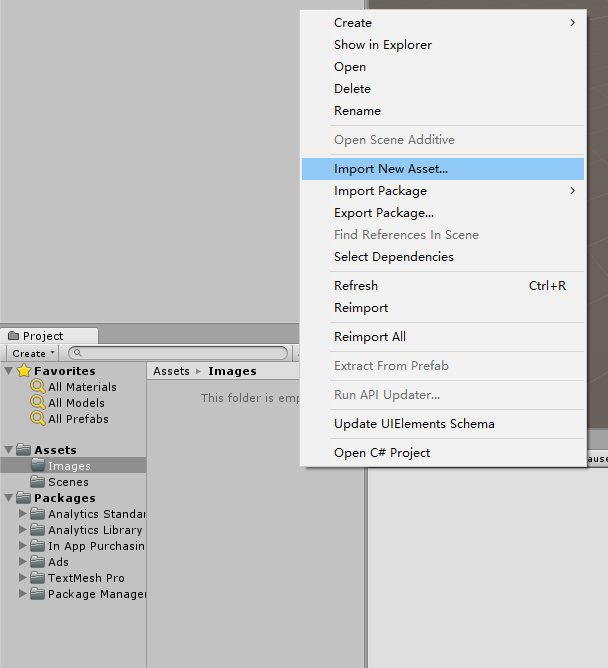
## 图片资源导入设置

Unity引擎只是大部分图片资源格式的导入，包括BMP、EXR、GIF、HDR、IFF、JPG、PICT、PNG、PSD、TGA、TIFF格式，导入后根据不同的设备，不同的平台进行不同的设置。

### 图片资源导入方式

导入方式一：如上面3.2小节介绍一样您可以选择把图片资源复制到您所创建的Unity项目工程目录的Assets路径或者子目录下。

导入方式二：如下图（左），在项目窗口Assets下的相应子目录点击右键打开Import New Asset... 选项（同理如下图（右）所示，点击菜单栏打开Import New Asset... 选项），选择一张本地图片进行导入，图片资源就会在项目窗口的相应文件夹内。

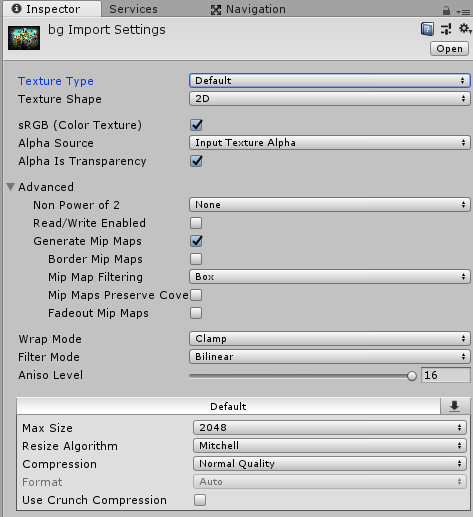


右键点击项目窗口 菜单栏选择导入资源

### 图片纹理类型

您可以通过纹理导入器将不同的纹理类型导入Unity编辑器。

#### 默认贴图

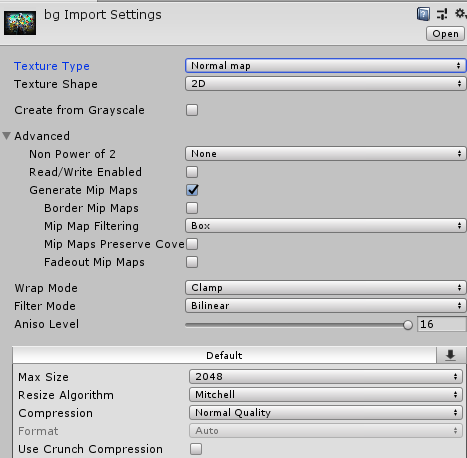


图片Inspector窗口-**纹理类型：默认**

**默认贴图选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | **默认选项**是所有纹理使用的最常用设置，它提供对纹理导入的大多数属性的访问。 |
| **Texture Shape（纹理形状）** | 您可以用它来选择纹理的形状。 |
| **sRGB (Color Texture) （彩色纹理）** | 勾选此选项可指定纹理存储在伽马空间中，其始终检查非HDR颜色纹理（如反照率和高光色）。如果纹理存储具有特定含义的信息，并且您需要Shader中的确切值（例如光滑或金属），请取消勾选。此选项在默认情况下被选中。 |
| **Alpha Source（Alpha通道来源）** | 使用此选项生成纹理的alpha通道，默认为**None**。 |
| None | 无论导入的图片是否拥有Alpha通道，默认为没有Alpha通道。 |
| Input Texture Alpha（图片Alpha通道） | 如果图片有Alpha通道，就使用图片的 |
| From Gray Scale（从灰度生成） | 根据图片的RGB值得平均值生成Alpha值 |
| **Alpha is Transparency（Alpha代表透明度）** | 如果勾选此选项，则启用Alpha透明度可以扩大颜色并避免在图片边缘上过滤。 |
| **Advanced（高级选项）** | |
| **Non Power of 2（2的非幂次）** | 如果图片具有两个非幂次（NPOT）尺寸大小，则会定义导入时的缩放行为。默认设置为**None**。 |
| None | 图片尺寸大小保持不变。 |
| To nearest（最近） | 图片在导入时缩放到最接近的2的幂次尺寸。例如，257x511像素缩放到256x512像素。请注意，PVRTC格式要求纹理为正方形（即宽度等于高度），因此最终尺寸大小会放大到512x512像素。 |
| To larger（更大） | 图片在导入时缩放到最大尺寸值的2的幂次。例如，257x511像素纹理缩放到512x512像素。 |
| To smaller（更小） | 图片在导入时缩放到最小尺寸值的2的幂次。例如，257x511像素纹理缩放到256x256像素。 |
| **Read/Write Enabled（读写使能）** | 勾选此选项以启用从脚本（例如Texture2D.SetPixels，Texture2D.GetPixels和其他Texture2D函数）访问Texture数据。请注意，制作了Texture数据的副本，将Texture Assets所需的内存量加倍，因此如果绝对必要，请不要使用此选项。这仅适用于未压缩和DXT压缩纹理; 其他类型的压缩纹理无法读取。该选项默认是禁用的。 |
| **Generate Mip Maps（生成MipMap）** | 此选项用于生成MipMap，MipMaps是在图片在屏幕上非常小时使用的图片的较小版本。 |
| Border Mip Maps（MipMap边框） | 此选项可避免颜色溢出到较低MIP级别的边缘。此选项在默认情况下不勾选。 |
| Mip Map Filtering（MipMap过滤） | 有两种可用于优化图像质量的mipmap过滤方法。默认选项是**Box**。 |
| Box | 这是淡出mipmap的最简单方法。随着尺寸的减小，MIP级别变得更加平滑。 |
| Kaiser | 锐化算法在维度大小下降时在mipmap上运行。如果您的纹理在距离上太模糊，试试这个选项。 |
| Mip Maps Preserve Coverage（纹理贴图保留覆盖） | 选择此选项在Alpha测试时Alpha通道值会保留覆盖。 |
| Fadeout Mip Maps（淡出MipMap） | 启用此选项可使MIP图层随着MIP级别的进展而渐变为灰色。这用于详细地图。最左边的滚动条是第一个开始淡出的MIP级别。最右侧的滚动条定义了纹理完全变灰的MIP级别。 |
| **Wrap Mode（平铺方式）** | 选择平铺时Texture的行为方式。默认选项是**Clamp**。 |
| Repeat（重复） | 图片以Tile重复。 |
| Clamp（拉伸） | 图片的边缘被拉伸 |
| **Filter Mode（过滤模式）** | 选择图片在被3D变形拉伸时如何被过滤。默认选项是**Point（无过滤器）**。 |
| Point (no filter)（点（无过滤器）） | 图片成紧密块状。 |
| Bilinear（双线性） | 图片近似模糊 |
| Trilinear（三线性） | 像Bilinear一样，但图片也在不同的MIP级别之间模糊。 |

#### 法线贴图

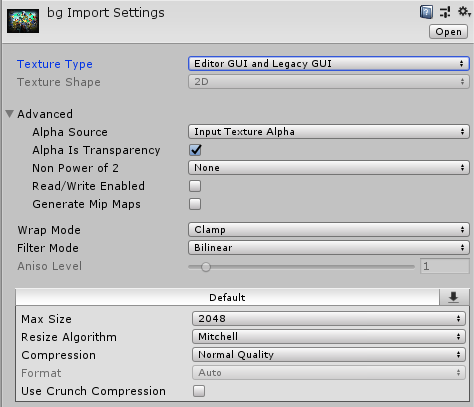


图片Inspector窗口 - **纹理类型:法线贴图**

**法线贴图选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 选择**法线贴图选项**将颜色通道转换为适合实时法线贴图的格式。 |
| **Texture Shape（纹理形状）** | 您可以用它来选择纹理的形状。 |
| **Create from Greyscale（从灰度值生成）** | 这将从灰度高度图创建法线贴图。选中此选项启用它查看**颠簸**和**过滤**。该选项默认是未选中的。 |
| **Bumpiness（颠簸）** | 控制颠簸的量。低凹凸值意味着即使在高度图中形成鲜明的对比度也会转化为平缓的角度和颠簸。较大的值产生夸张的颠簸和非常高对比度的颠簸效果。该选项仅在从灰度生成被选中才可见。 |
| **Filtering（过滤）** | 怎样计算颠簸的量 |
| ***Smooth（平滑的）*** | 生成具有标准（前向差异）算法的法线贴图。 |
| ***Sharp（尖锐的）*** | 也称为索贝尔过滤器，这会生成比标准更尖锐的法线贴图。 |

#### 编辑器GUI 、传统GUI

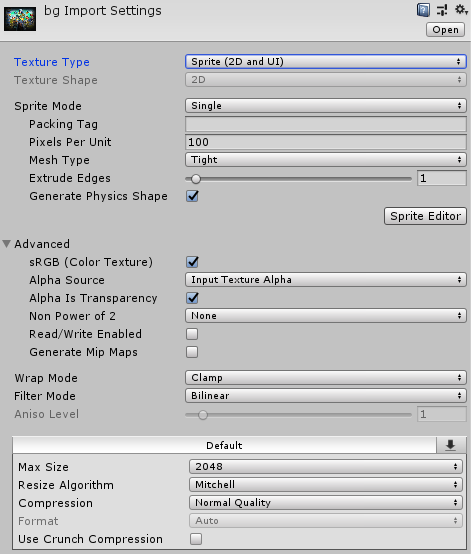


图片Inspector窗口 - **纹理类型:编辑器GUI和传统GUI**

**编辑器GUI和传统GUI选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 当您在HUD或GUI空间使用Texture，您可以选择选择**Editor GUI和Legacy GUI**选项。 |
| **Texture Shape（纹理形状）** | 您可以用它来选择纹理的形状。 |

#### 图片精灵 (2D 和 UI)

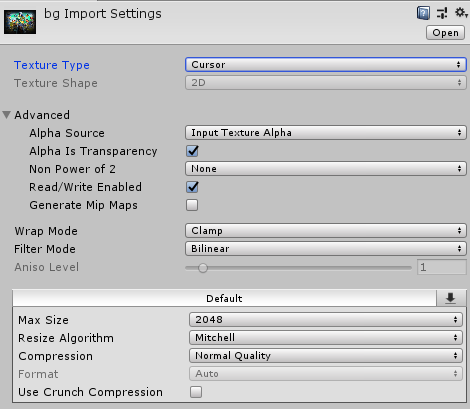


图片Inspector窗口 - **纹理类型:精灵图片（2D和UI）**

**精灵图片选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 当您在2D游戏使用Sprite时，您需要选择为**Sprite（2D和UI）**。 |
| **Texture Shape（纹理形状）** | 您可以用它来选择纹理的形状。 |
| **Sprite mode（精灵模式）** | 此选项如何从图像中提取Sprite图形。此选项的默认值为**Single**。 |
| Single（单个） | 只有一个Sprite。 |
| Multiple（多个） | 将多个相关的精灵集中在同一个图片中（如属于单个游戏角色的动画帧或单独的Sprite元素）。 |
| **Packing Tag（包的标签）** | 按名称指定要将此纹理打包到的Sprite图集。 |
| **Pixels Per Unit（每个单元的像素量）** | Sprite图像中对应于世界空间中的一个距离单位的宽度/高度的像素数量，。 |
| **Mesh Type（网格类型）** | Sprite生成的Mesh类型，默认为**Tight**。 |
| Full Rect | 创建一个四边形来将Sprite填充到上面。 |
| Tight | This generates a Mesh based on pixel alpha value. The Mesh generated generally follows the shape of the Sprite.  **Note:** Any Sprite that is smaller than 32x32 uses **Full Rect**, even when **Tight** is specified.  根据像素阿尔法值生成一个Mesh，生成的网格通常遵循Sprite的形状。 **注意：**即使指定了**Tight**，任何小于32x32的Sprite都会使用**Full Rect**。 |
| **Extrude Edges（边缘）** | 使用滑块确定在生成的Mesh中Sprite周围有多少区域。 |
| **Pivot（中轴点）** | 图像中Sprite的相对坐标坐标系所在位置。选择一个预设选项，或选择**自定义**以设置您自己的中轴点位置。 |
| Custom（自定义） | 设置Pivot在图像中的X和Y值 |

#### 光标

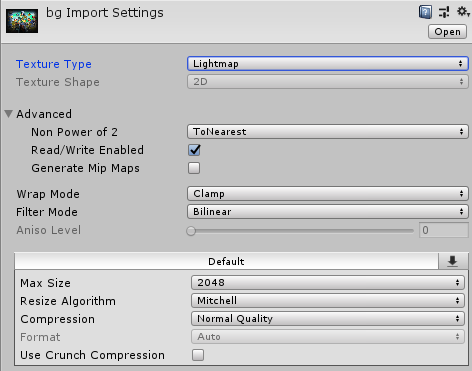


图片Inspector窗口 - **纹理类型:光标**

**光标选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 把图片做为一个光标资源使用。 |

#### 光照贴图

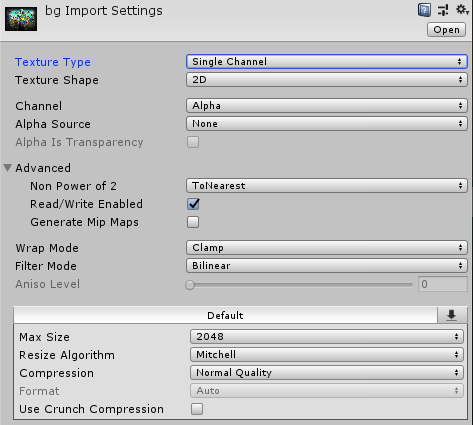


图片Inspector窗口 - **纹理类型:光照贴图**

**光照贴图选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 光照贴图选项使编码成特定格式（RGBM或dLDR，取决于平台）和纹理数据（推挽扩张通道）上的后处理步骤。 |

#### 单通道



图片Inspector窗口 - **纹理类型:单通道**

**单通道选项：**

| **属性:** | **功能介绍:** |
| --- | --- |
| **Texture Type（纹理类型）** | 此选项为只需要图片的一个通道 |
| **Texture Shape（纹理形状）** | 您可以用它来选择纹理的形状。 |
| **Alpha Source（Alpha通道来源）** | 使用此选项生成纹理的alpha通道，默认为**None**。 |
| None | 无论导入的图片是否拥有Alpha通道，默认为没有Alpha通道。 |
| Input Texture Alpha（图片Alpha通道） | 如果图片有Alpha通道，就使用图片的 |
| From Gray Scale（从灰度生成） | 根据图片的RGB值得平均值生成Alpha值 |
| **Alpha is Transparency（Alpha代表透明度）** | 如果勾选此选项，则启用Alpha透明度可以扩大颜色并避免在图片边缘上过滤。 |

## 模型资源导入流程

**注意：**导入流程需要一个可导入的模型文件工作流。如果没有，可以在从第三方的3D建模软件导出文件。

模型文件可以包含各种数据，如角色网格，动画绑定和剪辑，以及材质和纹理。如果导入文件不包含所有这些元素，但可以按照相关的任何工作流程部分进行操作。

无论您想从模型文件中提取什么样的数据，您需要：1.打开**项目**窗口和**Inspector，**以便您可以一次看到两者。2. 在“**项目”**窗口的**资产**文件夹中选择要导入的**模型**文件。“**导入设置”**窗口在**检查器中**打开，默认显示“**模型”标签**。

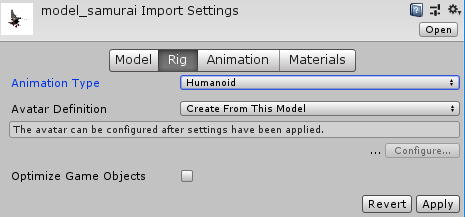
### 导入人形动画

当Unity导入包含**Humanoid** Rigs和Animation的模型文件时，它需要将模型的骨骼结构与其动画进行协调。它通过将文件中的每个骨骼映射到一个人形生物Avatar来实现这一点，以便它可以正确地播放动画。因此，在导入Unity之前仔细准备您的Model文件非常重要。

1. 定义Rig类型并创建Avatar。
2. 更正或验证Avatar的映射。
3. 完成骨骼映射后，您可以使用Muscles & Settings选项来调整Avatar的肌肉配置。
4. 您可以选择save the mapping of your skeleton’s bones to the Avatar为人体模板（.ht）文件。
5. 您可以通过定义一个Avatar Mask来限制在特定骨骼上导入的动画。
6. 在“Animation”选项中，启用“Import Animation”选项，然后设置其他特定于资产的属性。
7. 如果文件包含多个动画或动作，则可以将特定的动作范围定义为动画片段。
8. 对于文件中定义的每个动画片段，您可以做以下操作：
   1. 更改Pose和Root的Transform
   2. 优化循环
   3. 镜像人形骨骼两侧的动画。
   4. 将 Curves添加到Clips以便动画更加生动。
   5. 将Events添加到Clips以便在动画中及时触发某些操作
   6. 放弃部分动画，类似于运行时的Avatar Mask，但是这个是在导入时应用
   7. 选择一个不同的根运动节点来驱动动作
   8. 从Unity中读取关于导入Clip的任何消息
   9. 观看动画片段的预览
9. 要保存更改，请单击“**导入设置”**窗口底部的“Apply**”**按钮或“Revert**”**以放弃更改。

#### 设置Avatar

点击Inspector窗口的“Rig”标签，将“动画类型”设置为Humanoid。默认情况下，“头像定义”属性设置为“Create From This Model”。如果保留此选项，Unity会尝试将文件中定义的一组骨骼映射到人形Avatar。



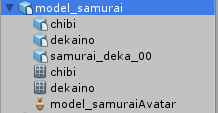
人形绑定

**注意：**在某些情况下，您可以将此选项更改为**Copy From Other Avatar**来使用其他模型定义的Avatar。例如，如果您在3D建模软件中使用几种不同的动画创建网格（Skin），则可以将网格导出为一个FBX文件，并将每个动画导出为其自己的FBX文件。将这些文件导入到Unity中时，只需要为导入的第一个文件（通常是Mesh）创建一个头像。只要所有文件使用相同的骨骼结构，就可以重新使用该头像来处理其余文件（如所有动画）。

如果启用此选项，则必须通过设置**Source**属性来指定要使用的头像。

点击**Apply**按钮。Unity会尝试将现有骨骼结构与Avatar骨骼结构进行匹配。在许多情况下，它可以通过分析动画中骨骼之间的连接自动完成。

如果匹配成功，**Configure** 菜单旁边会出现复选标记。Unity还会为模型资产添加一个Avatar的子资源，您可以在项目视图中找到它。



导入模型的Avatar资源

匹配成功仅仅意味着Unity能够匹配所有必需的骨骼。为了获得更好的结果，还需要匹配可选骨骼并将模型设置为适当的**T形Pose**。

如果Unity无法创建Avatar，则在**配置**按钮旁会出现一个×，并且在项目视图中不会出现Avatar子资源。

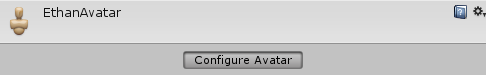
由于Avatar是动画系统的一个重要方面，为**模型进行**适当配置非常重要。为此，无论自动化Avatar创建是否成功，您都应该始终检查Avatar是否有效并正确设置。

#### 配置Avatar

如果Unity不能给模型创建Avatar，则必须单击“**Rig”标签**上的“**配置...”**按钮以打开“Avatar”窗口并修复Avatar。

如果匹配成功，您可以单击“Rig**”**标签上的“**配置...”**按钮或从“项目”视图打开窗口：

1. 在**项目**视图中单击Avatar资源。Inspector窗口显示头像和名称**配置头像**按钮。
2. 点击**配置头像**按钮。



Avatar资源的Inspector窗口

如果您尚未保存Avatar，则会出现一条消息，要求您保存场景， 原因是在**配置**模式下，场景视图用于单独显示所选模型的骨骼，肌肉和动画信息，而不显示场景的其余部分。

一旦保存了场景，**Avatar**窗口将出现在**Inspector中，**显示任何骨骼映射。

确保骨骼映射是正确的，并且映射Unity没有分配的任何可选骨骼。

Skeleton需要至少含有所需的骨头，以便Unity产生有效的匹配。为了提高Avatar匹配的机会，请以反映他们所代表的身体部位的方式命名骨骼。例如，“LeftArm”和“RightForearm”清楚表达这些骨骼是如何控制的。

#### 创建Avatar Mask

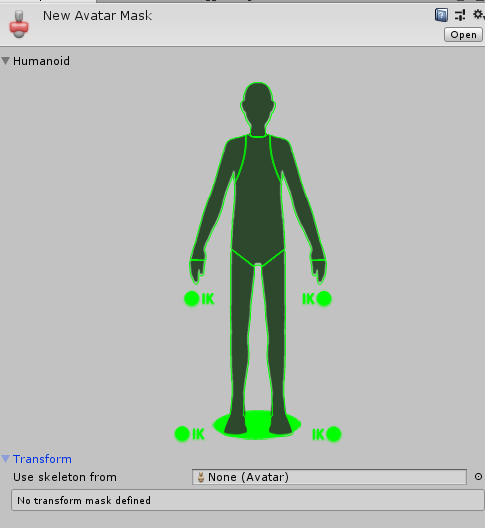
遮罩允许您放弃剪辑中的一些动画数据，允许剪辑仅为对象或角色的一部分而不是整个物体制作动画。如可能有一个标准的步行动画，包括手臂和腿部动作，但如果一个角色用双手拎着一个大物体，那么当他们走路时，您不会希望他们的手臂摆到一边。但是，您仍然可以在携带对象时使用标准行走动画，并使用遮罩只在行走动画的顶部播放承载动画的上半身部分。

您可以在导入时或运行时将遮罩应用于动画片段。在导入期间遮罩是比较好的，因为它允许从构建中省略丢弃的动画数据，使文件变小并因此使用更少的内存。它还可以加快处理速度，因为运行时混合的动画数据较少。在某些情况下，导入遮罩不能满足需求。则可以在运行时通过创建一个“Avatar Mask”资源来应用遮罩，并在“Animator Controller”的“Layer Settings”中使用它。

创建一个空的Avatar Mask资源的步骤如下：

* 从“**Assets**”菜单中选择“**Create** -> **Avatar Mask**”。
* 在**项目**视图中单击要定义遮罩的模型对象，然后右键单击并选择“**Create** -> **Avatar Mask”**。

新资源出现在**项目**视图中：



Avatar Mask窗口

现在，您可以将身体部分添加到遮罩，然后将遮罩添加到动画层或在动画选项的遮罩部分下添加对其的引用。

### 导入非人形动画

一个人形模型是一个非常特殊的结构，内含至少在某种程度上符合实际的人类骨骼15根骨头。其他所有使用Unity动画系统的属于非人形或其他类别。这意味着Generic 可能是从茶壶到龙的任何东西，所以非人形骨骼可能有很多可能的骨骼结构。

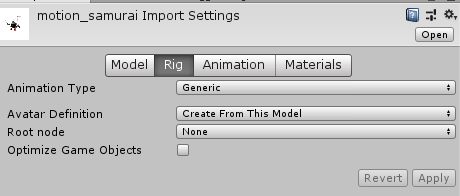
处理这种复杂性的解决方案是Unity只需要知道哪个骨骼是根节点。就通用角色而言，这是对人形角色重心的最佳近似。它有助于Unity确定如何以最最佳方式呈现动画。由于只有一个骨骼需要映射，所以Generic设置不使用Humanoid Avatar窗口。因此，准备将非人形模型文件导入Unity需要的步骤少于人形的模型导入。

1. 设置“Rig”为**“Generic”**。
2. 通过定义“Avatar Mask” 来限制在特定骨骼上导入的动画。
3. 在“**Animation”标签**中，启用“**Import Animation”**选项，然后设置其他特定于资源的属性。
4. 如果文件包含多个动画或动作，则可以将特定的帧范围定义为动画片段。
5. 对于文件中定义的每个动画片段，可以进行以下设置：
   1. 设置Pose和Root Transform
   2. 优化循环
   3. 将 Curves添加到Clips以便动画更加生动。
   4. 将Events添加到Clips以便在动画中及时触发某些操作
   5. 放弃部分动画，类似于运行时的Avatar Mask，但是这个是在导入时应用
   6. 选择一个不同的根动作节点来驱动动作
   7. 从Unity中读取关于导入Clip的任何消息
   8. 观看动画片段的预览
6. 要保存更改，请单击“**导入设置”**窗口底部的“**应用”**按钮或“**恢复”**以放弃更改。

#### 设置Rig

在Inspector window的“Rig”标签，将“Avatar （animation）”类型设置为“Generic”。默认情况下，“Avatar Definition”属性设置为“Create From This Model”，“Root node”选项设置为“None”。

在某些情况下，可以将“Avatar Definition”选项更改为“Copy From Other Avatar”，以使用已为其他模型文件定义的Avatar。例如，如果您在3D建模软件中使用几种不同的动画创建网格（skin），则可以将网格导出为一个FBX文件，并将每个动画导出为其自己的FBX文件。将这些文件导入到Unity中时，只需要为导入的第一个文件（通常是Mesh）创建一个Avatar。只要所有文件使用相同的骨骼结构，就可以重新使用该头像来处理其余文件（如所有动画）。

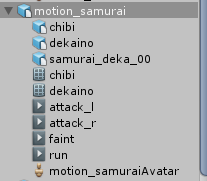


Generic Rig

如果保留“**Create From This Model”**选项，则必须从“**Root node”**属性中选择一个骨骼。

如果您决定将“**Avatar Definition”**选项更改为**“Copy From Other Avatar”**，则需要通过设置“**Source”**属性来指定要使用的头像。

点击**“Apply”**按钮。Unity会创建一个**“Generic** **Avatar”**并在模型资产下添加一个Avatar子资源，可以在项目视图中找到该资源。



模型的Avatar子资源

**注意：**“Generic Avatar”与“Humanoid Avatar”不同，但它确实出现在项目视图中，并且它确实包含根节点映射。但是，如果单击“项目”视图中的“头像”图标以在“**检查器”中**显示其属性，则只会显示其名称，并且没有“**Configure Avatar”**按钮。

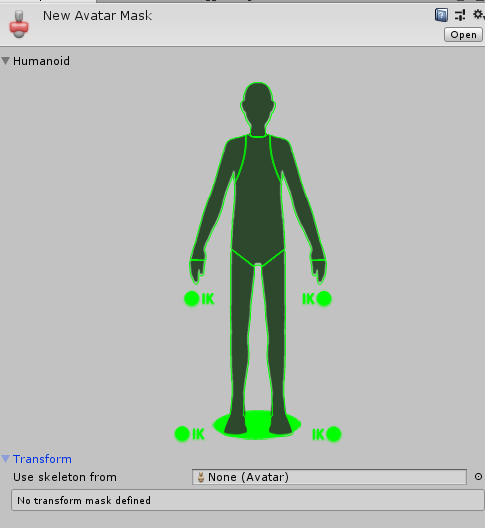
#### 创建Avatar Mask

您可以在导入时或运行时将遮罩应用于动画片段。在导入期间遮罩是最好的，因为它允许从构建中省略丢弃的动画数据，使文件变小并因此使用更少的内存。它还可以加快处理速度，因为运行时混合的动画数据较少。在某些情况下，导入掩蔽可能不适合您的目的。在这种情况下，您可以在运行时通过创建一个**“Avatar Mask”**资产来应用遮罩，并在**“Animator Controller”**的“Layer Settings”中使用它。

创建一个空的Avatar Mask资源的步骤如下：

* 从“**Assets**”菜单中选择“**Create** -> **Avatar Mask**”。
* 在**项目**视图中单击要定义遮罩的模型对象，然后右键单击并选择“**Create** -> **Avatar Mask”**。

现在，您可以选择要将哪些骨骼包含在Transform层次结构中或从Transform层次结构中排除，然后将遮罩添加到动画层或在“Animation”标签的“Mask”部分下添加对其的引用。



Avatar Mask窗口

### 模型资源导入设置

将模型文件放入Unity项目下的Assets文件夹时，Unity会自动导入并将它们存储为Unity资源。要在**Inspector中**查看导入设置，请单击**项目窗口**中的文件。您可以通过在此窗口的四个标签上设置属性来自定义Unity如何导入选定文件：

[Model tab](https://docs.unity3d.com/2018.1/Documentation/Manual/FBXImporter-Model.html)3D 模型可以代表角色，建筑物或家具。在这些情况下，Unity会从一个模型文件中创建多个资源。在项目窗口中，主要导入的对象是模型**预制件**。通常Prefab 还会引用几个**Mesh**对象。

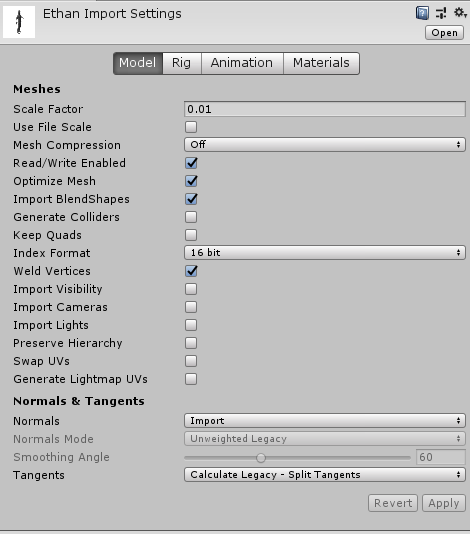
[Rig tab](https://docs.unity3d.com/2018.1/Documentation/Manual/FBXImporter-Rig.html)“Rig”（有时称为骨架）包括一组在3D建模软件中创建的一个动画网格（有时叫Skin）上的一个或多个模型，例如作为3ds Max或Maya）。对于**Humanoid**和**Generic**（非人形）模型，Unity会创建一个**Avatar**来协调导入的Rig及 **GameObject**。

[Animation tab](https://docs.unity3d.com/2018.1/Documentation/Manual/class-AnimationClip.html)此标签可以定义出现在一组任何一系列不同的姿势，例如走路，跑步，甚至是Idle状态（从一只脚移到另一只脚）作为动画剪辑。您可以重复使用具有相同Rig的任何模型的剪辑。通常单个文件包含几个不同的动作，每个动作都可以定义为特定的**动画片段**。

[Materials tab](https://docs.unity3d.com/2018.1/Documentation/Manual/FBXImporter-Materials.html)此标签可以提取材质和纹理，或将它们嵌入到模型中。您还可以调整材质在模型中的映射方式。

#### 模型标签

选择模型时，Inspector窗口将显示模型文件的“**Import Settings**”，其中“**Model”** 标签可以修改**网格**及其**法线并应用**。



导入设置Model标签

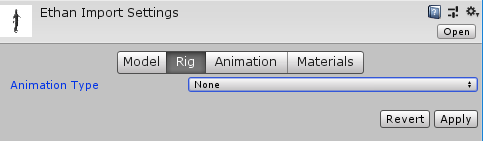
**模型标签选项：**

| **属性** | **功能** |
| --- | --- |
| **Meshes（网格）** | |
| **Scale Factor（缩放因子）** | Unity的物理系统预计游戏世界中的1米在导入的文件中为1个单位。 对于不同的3D封装的默认值如下： .fbx, .max, .jas, .c4d = **0.01** .mb, .ma, .lxo, .dxf, .blend, .dae = **1**  .3ds = **0.1** |
| **Use File Scale（使用文件缩放比例）** | 使用模型文件中定义的模型缩放比例，或取消选中为您的模型设置自定义**缩放因子**值。 |
| **File Scale（文件缩放比例）** | 模型设置自定义比例，仅在选中**使用文件比例**时才可用。 |
| **Mesh Compression（网格压缩）** | 设置压缩比率的级别以减小网格的文件大小。通过使用网格边界和每个组件的较低比特深度来压缩网格数据，增加压缩比会降低网格的精度。 最好将Mesh看起来和未压缩的版本看起来差不多。这对优化游戏包的大小很有用。 |
| **Off（关闭）** | 禁用压缩 |
| **Low（低）** | 使用低压缩比 |
| **Medium（中）** | 使用中压缩比 |
| **High（高）** | 使用高压缩比 |
| **Read/Write Enabled（读写使能）** | 如果启用，Mesh数据将保存在内存中，以便自定义脚本可以读取并更改它。禁用此选项会节省内存，因为Unity可以在游戏中卸载Mesh数据的副本。但是，在某些情况下，当Mesh与Mesh Collider一起使用时，必须启用此选项。这些情况包括： - 负比例缩放（例如，（-1，1，1））。 - 剪切变换（例如，当旋转的网格具有缩放的父变换时）。 |
| **Optimize Mesh（优化网格）** | 让Unity确定三角形在网格中的排列顺序。Unity对顶点和索引进行重新排序以获得更好的GPU性能。 |
| **Import BlendShapes（导入BlendShapes）** | 允许Unity使用Mesh导入BlendShapes。 **注意**导入Blendshape法线需要FBX文件中的smoothing groups。 |
| **Generate Colliders（生成碰撞盒子）** | 启用自动附加的Mesh Colliders导入Meshes。这对于静态物体快速生成碰撞网格非常有用，但对于正在动态物体应该避免使用。 |
| **Keep Quads（保持四边形）** | 启用此功能可停止Unity将具有四个顶点的多边形转换为三角形。例如，如果您正在使用Tessellation Shaders，则可能需要启用此选项，因为细化四边形可能比细化多边形效率更高。 Unity可以导入任何类型的多边形（三角形到N形）。超过四个顶点的多边形会不管此设置总是转为三角形。但是，如果网格同时具有四边形和三角形（或者将N-gons转换为三角形），则Unity会创建两个子网格来分隔四边形和三角形。每个子网格仅包含三角形或仅包含四边形。 **提示：**如果要将四边形导入Unity，则必须在3ds Max中使用可编辑的多边形。 |
| **Index Format（索引格式）** | 定义Mesh索引缓冲区的大小。 **注意：**由于带宽和内存存储容量的原因，您通常希望保留**16位**索引作为默认值，并且在必要时仅使用**32位**，这是**Auto**选项的用途。 |
| **Auto（自动）** | 让Unity根据Mesh顶点数决定是否在导入Mesh时使用16位或32位索引。这是Unity 2017.3及更高版本中添加的资源的默认值。 |
| **16 bit（16位）** | 导入Mesh时使用16位索引。如果Mesh较大，则它被分成<64k个顶点块。在Unity 2017.2或以前版本中制作的项目中已经存在的资源将使用此设置。 |
| **32 bit（32位）** | 导入Mesh时使用32位索引。如果您正在使用基于GPU的渲染管线（例如，使用compute shader triangle裁剪），则使用32位索引可确保所有网格使用相同的索引格式。这减少了着色器的复杂性，因为它们只需要处理一种格式。 |
| **Weld Vertices（优化定点）** | 组合在空间中有相同位置的顶点。这通过减少它们的总数来优化网格上的顶点数。该选项默认启用。 在某些情况下，导入Mesh时可能需要关闭此优化。例如，如果有意地在网格中占据相同位置的重复顶点，使用脚本来读取或操作单个顶点和三角形数据。 |
| **Import Visibility（导入可视）** | 导入定义MeshRenderer组件是否启用（可见）：Unity可以使用Import Visibility属性从FBX文件读取可见性属性。值和动画曲线可以通过控制Renderer.enabled属性来启用或禁用MeshRenderer组件。可见性继承默认为true，但可以被覆盖。例如，如果父级网格上的可见性设置为0，则其子级上的所有Render也将被禁用。在这种情况下，将为每个孩子的Renderer.enabled属性创建一条动画曲线。 |
| **Import Cameras（导入相机）** | 从.FBX文件导入相机时，Unity支持以下属性：**支持正交和透视模式，支持视野和距离调整。**不支持目标摄像机。 |
| **Import Lights（导入灯光）** | 导入FBX文件中的灯光（Omni，Spot，Directional，Area）。 |
| **Preserve Hierarchy（保留层次结构）** | 始终创建一个明确的预制根，即使模型只有一个根节点。通常，FBX Importer会从模型中去除空的根节点来优化。但是，如果有多个具有相同层次结构部分的FBX文件，则可以使用此选项来保留原始层次结构。 例如，file1.fbx包含一个Rig和一个Mesh，file2.fbx包含相同的Rig，但只包含该Rig的动画。如果在不启用此选项的情况下导入file2.fbx，Unity将去除根节点，层次结构就会不匹配，动画会无法使用。 |
| **Swap UVs（交换UV）** | 交换网格中的UV通道。如果您的漫反射纹理使用光照贴图中的UV，请使用此选项。Unity支持多达8个UV通道，但并非所有3D建模软件都能输出两个以上。 |
| **Generate Lightmap UVs（生成光照贴图UV）** | 为光照贴图创建第二个UV通道。 |
| **Normals & Tangents（法线和切线）** | |
| **Normals（法线）** | 如何计算法线。这对优化游戏大小很有用。 |
| **Import（导入）** | 默认选项是从文件导入法线。 |
| **Calculate（计算）** | 根据**Smoothing Angle**（below）计算法线。 |
| **None（没有）** | 禁用法线。如果Mesh既无法线贴图也不受实时照明的影响，请使用此选项。 |
| **Normals Mode（法线模式）** | 当**法线**设置为**计算**时计算法线。 |
| **Unweighted Legacy（传统无权重法）** | 计算法线的传统方法（在版本2017.1之前）。 |
| **Unweighted（无权重法）** | 无权重法。 |
| **Area Weighted（面积权重法）** | 根据每个面的面积权重计算 |
| **Angle Weighted（角度权重法）** | 法线由每个面上的顶角权重计算。 |
| **Area and Angle Weighted（面积与角度权重法）** | 默认选项是由每个面面积和每个面的顶点角度权重计算。 |
| **Smoothing Angle（平滑角度）** | 控制是否为边的分割顶点：通常，较高的值会导致较少的顶点。 **注意：**仅在非常光滑的物体或非常多的面的模型上使用此设置。否则，您最好在3D建模软件中手动设置平滑，然后使用设置为**导入**的“**Normals”**选项**导入**。由于Unity是以单一角度为基础，所以模型的某些部分最终可能会错误地计算。 只有在**Normals**设置为**Calculate**时才可用。 |
| **Tangents（切线）** | 导入或计算顶点切线。只有当**法线**设置为**计算**或**导入**时才可用。 |
| **Import（导入）** | 如果**Normals**设置为**Import，**则从FBX文件导入顶点切线。如果网格没有切线，它将不能用于法线贴图的着色器。 |
| **Calculate Tangent Space（切线空间）** | 默认是使用MikkTSpace计算切线。且**Normals**设置为**Calculate**。 |
| **Calculate Legacy（传统算法）** | 用传统算法计算切线。 |
| **Calculate Legacy - Split Tangent（传统算法-分割切线）** | 使用传统算法计算切线，并在UV图表上分割。当法线光照贴图被网格上的接缝有问题，请使用此选项。这通常只适用于角色。 |
| **None（无）** | 禁止导入顶点切线。网格没有切线，对于法线贴图的着色器不起作用。 |

#### Rig标签

“Rig**”** 标签上的设置定义了Unity如何将模型映射到导入模型中的网格，以便您可以对其进行动画处理。对于人形角色，这意味着分配或创建一个Avatar。对于非人形（Generic）角色，需要识别骨架中的根节点。

默认情况下，当您在**项目**视图中选择模型时，Unity将确定哪个**动画类型**与所选模型最匹配，并将其显示在“RIg**”** 标签中。如果未导入文件，则动画类型设置为**无**：

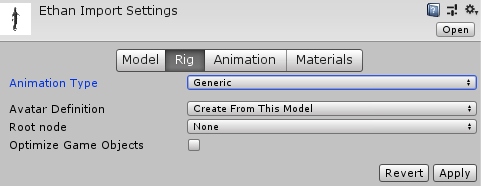


无Rig Animation映射

**Rig标签选项：**

| **属性:** | **功能:** |
| --- | --- |
| **Animation Type（动画类型）** | 指定动画的类型。 |
| **None（无）** | 无动画 |
| **Legacy（老版本）** | 使用老版本的的动画，当导入和使用Unity版本3.x和更早版本的动画时使用。 |
| **Generic（通用非人形）** | 如果Rig是非人形的（四足动物或任何需要动画的实体），请使用通用动画系统。Unity会选取一个根节点，但您可以确定另一个骨骼用作**根节点**。 |
| **Humanoid（人形）** | 如果Rig是人形的（它有两条腿，两条手臂和一个头），则使用人形动画系统。Unity通常会检测到骨架并将其正确映射到Avatar。在某些情况下，您可能需要手动更改“**Avatar Definition”**和“**Configure**”映射。 |

##### 通用动画类型



非人形Rig

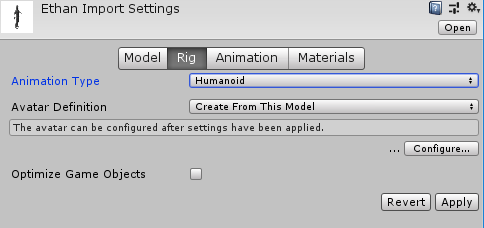
通用动画不使用像人形动画那样的Avatar。由于骨架可以是任意的，所以您必须指定哪个骨骼是**根节点**。根节点允许Unity在通用模型的动画剪辑之间建立一致性，并且在动画之间进行一定的混合。

指定根节点有助于Unity确定骨骼相对于彼此的运动和世界中根节点的运动（由OnAnimatorMove控制）。

**Rig标签的Generic选项：**

| **属性:** | **功能:** |
| --- | --- |
| **Avatar Definition（定义Avatar）** | 选择获取Avatar定义的位置。 |
| **Create from this model（从这个模型创建）** | （根据此模型创建一个Avatar） |
| **Copy from Other Avatar（从其他Avatar复制）** | （指向在另一个模型上设置的Avatar） |
| **Root node（根节点）** | 选择要用作此Avatar的根节点的骨骼。 仅在“**Avatar Definition”**设置为“**Create From This Model”**时才可用。 |
| **Source（资源）** | 使用相同的Rig复制的另一个Avatar以导入其动画片段。 仅在“**Avatar Definition”**设置为**“Copy from Other Avatar”**时才可用。 |

##### 人形动画类型



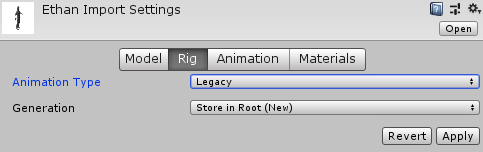
人形（拥有两条腿，两条手臂和一个头）

除极少数例外情况外，人形模型具有相同的基本结构。这个结构代表了身体的主要关节部分：头部和四肢。使用Unity的人形动画功能的第一步是设置和配置一个Avatar。Unity使用Avatar将简化的人形骨骼结构映射到模型骨架中的实际骨骼。

**Rig标签的Humanoid选项：**

| **属性:** | **功能:** |
| --- | --- |
| **Avatar Definition（定义Avatar）** | 选择获取Avatar定义的位置。 |
| **Create from this model（从这个模型创建）** | 根据此模型创建一个Avatar |
| **Copy from Other Avatar（从其他Avatar复制）** | 指向在另一个模型上设置的Avatar |
| **Source（资源）** | 使用相同的Rig复制的另一个Avatar以导入其动画片段。 仅在“**Avatar Definition”**设置为**“Copy from Other Avatar”**时才可用。 |
| **Configure…（配置）** | 打开头像配置。 仅在“**Avatar Definition”**设置为“**Create From This Model”**时才可用。 |
| **Optimize Game Object（优化游戏对象）** | 移除并存储导入角色的GameObject Transform层次结构到Avatar和Animator组件中。如果启用，角色的SkinnedMeshRenderers使用Unity动画系统的内部框架，这可以提高动画角色的性能。 仅在“**Avatar Definition”**设置为“**Create From This Model”**时才可用。 为最终发布的产品启用此选项。 **注意**：在优化模式下，蒙皮网格矩阵提取也是多线程的。 |

##### 老版本（Legacy）动画类型



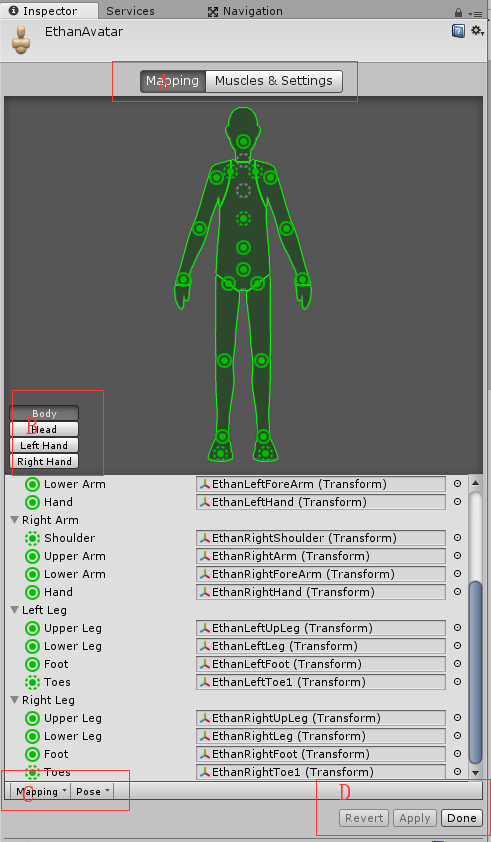
老版本动画系统

**Rig标签的Legacy选项**

| **属性:** | **功能:** |
| --- | --- |
| **Generation（生成）** | 选择动画导入方法 |
| **Don’t Import（不导入）** | 不要导入动画 |
| **Store in Root (New) （存在根节点）** | 默认设置：导入动画并将其存储在模型的根节点中。 |

##### Avatar映射选项

在保存场景之后，“**Avatar Mapping”**选项卡将显示在显示Unity骨骼映射的**Inspector窗口中**：



Avatar的骨骼映射

* （A）在**Mapping**和**Muscles & Settings**选项间切换的按钮。
* （B）在**身体**，**头部**，**左手**和**右手**的部分之间切换的按钮。
* （C）提供各种**Mapping**和Pose工具以帮助您将骨骼结构映射到Avatar的菜单。
* （D）可应用任何更改（Apply），放弃所有更改（Revert），并离开Avatar编辑窗口（Done）。

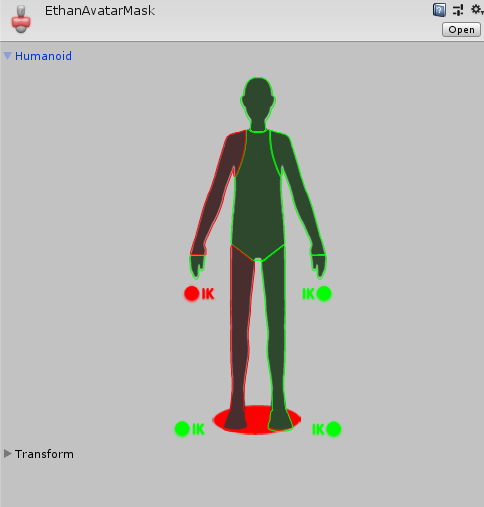
Avatar映射表明哪些骨头是必需的（实心圆圈），哪些是可选的（虚线圆圈）。Unity可以自动插入可选的骨骼。

##### Avatar遮罩

有两种方法可以定义动画的哪些部分应该被遮罩：

* 通过从Humanoid body map选择
* 通过选择应该有哪些骨骼或从Transform hierarchy排除

如果您的动画使用人形Avatar，则可以选择或取消选择简化的人体的某些部分，来标识屏蔽动画的位置：



用人体定义Avatar遮罩

人体图将身体部位分为以下部分：

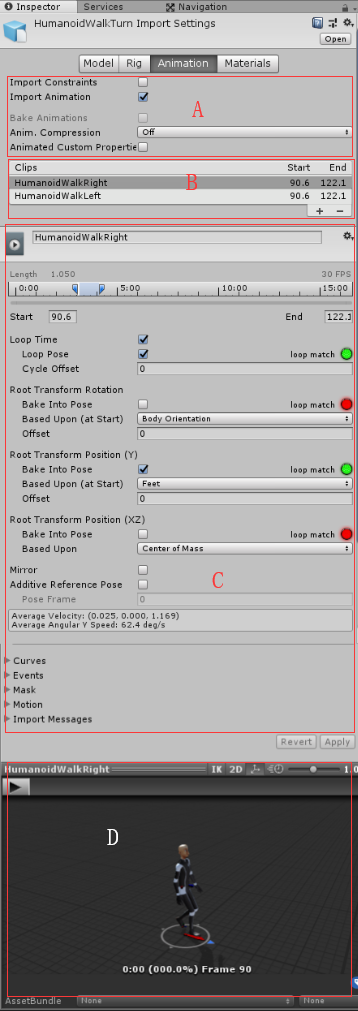
* 头、左臂、右臂、左手、右手、左腿、右腿、根（由脚下的“阴影”表示）

如果动画要包含一个身体的部位，请单击该部位的Avatar图，使其显示为绿色。要排除动画，请点击身体部位，则显示为红色。要包含或排除全部，请双击Avatar周围的空白部分。

您还可以切换手部和脚部的**反向运动学（IK）**，它决定是否在动画混合中包含IK曲线。

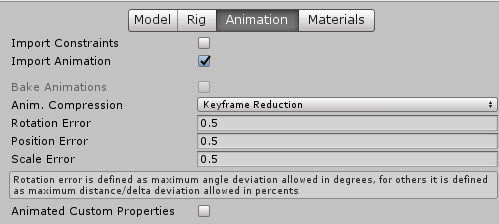
#### 动画标签

**动画片段**是Unity中最小的动画构建块。它们表示一个独立的运动，如RunLeft，Jump或Crouch，并且可以通过各种方式进行操作和组合，以生成活泼的最终效果，可以从导入的FBX数据中选择动画片段。



动画属性窗口

* （A）资源特定属性。这些设置为整个资产定义导入选项。
* （B）剪辑选择列表。您可以从此列表中选择任何项目以显示其属性并预览其动画。您还可以定义新的剪辑。
* （C）特定于剪辑的属性。这些设置为选定的**动画片段**定义导入选项。
* （D）动画预览。您可以播放动画并在此处选择特定的帧。



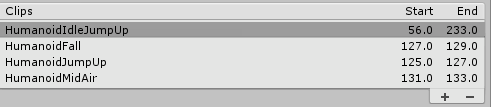
资源的导入设置

**Animation标签选项：**

| **属性:** | **功能:** |
| --- | --- |
| **Import Constraints（导入约束）** | 从资源中导入约束 |
| **Import Animation（导入动画）** | 从资源中导入动画。 **注意：**如果禁用，则此页面上的所有其他选项都将隐藏，并且不会导入动画。 |
| **Bake Animations（烘焙动画）** | 烘焙使用IK或FK关键帧创建的动画。 仅适用于Maya，3ds Max和Cinema 4D文件。 |
| **Resample Curves（采样曲线）** | 将动画曲线重新采样为Quaternion值，并为动画中的每个帧生成一个新的Quaternion关键帧。 该选项默认启用。只有当原始动画中的键之间的插值存在问题时，禁用此选项才能使动画曲线保持原始创作状态。仅当导入文件包含欧拉曲线时才会出现。 |
| **Anim. Compression（动画压缩）** | 导入动画时使用的压缩类型。 |
| **Off（禁用）** | 禁用动画压缩。这意味着Unity不会减少导入时的关键帧数量，所以这是最高精度的动画，但性能会降低，文件和运行时内存容量也会变大。通常不建议使用此选项 - 如果您需要更高精度的动画，则应该启用keyframe reduction并降低允许的 **Animation Compression Error**值。 |
| **Keyframe Reduction（关键帧缩减）** | 减少导入时的冗余关键帧。如果选中，则会显示“**动画压缩错误”**选项。这会影响文件大小（运行时内存）以及曲线。 |
| **Keyframe Reduction and Compression（关键帧缩减压缩）** | 将动画存储在文件中时，减少导入和压缩关键帧上的关键帧。这只影响文件大小 - 运行时内存大小与**关键帧缩减**相同。如果选中，则会显示“**动画压缩错误”**选项。 |
| **Optimal（最佳）** | 让Unity决定如何压缩，无论是通过减少关键帧还是使用密集格式。 仅适用于**Generic**和**Humanoid** 动画类型Rigs。 |
| **Animation Compression Errors（动画压缩错误）** | 仅在启用“**关键帧缩减”**或“**最佳**压缩”时可用。 |
| **Rotation Error（旋转错误）** | 减少旋转曲线多少钱 数值越小，精度越高。 |
| **Position Error（位置错误）** | 减少位置曲线多少钱 数值越小，精度越高。 |
| **Scale Error（缩放错误）** | 减少缩放曲线有多少。数值越小，精度越高。 |
| **Animated Custom Properties（动画自定义属性）** | 导入自定义用户属性的任何FBX属性。 导入FBX文件时（例如平移，旋转，缩放和可视性），Unity仅支持一小部分属性。但是，您可以通过extraUserProperties成员在导入器脚本中将它们命名为用户属性来处理标准FBX属性。在导入过程中，Unity会将所有已命名的属性传递给Asset后处理器，就像'real'用户属性一样。 |

**动画片段（Clips）列表**

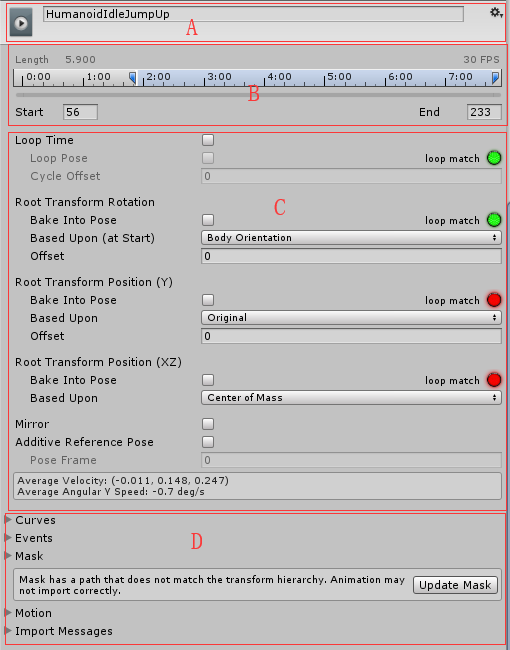
* 从列表中选择一个剪辑以显示其剪辑特定的属性。
* 在动画剪辑预览窗格中播放选定的剪辑。
* 使用add（+）按钮为该文件创建一个新动画剪辑。
* 使用delete（-）按钮删除选定的剪辑定义。



动画剪辑列表

**单个动画剪辑的属性**

* （A）所选剪辑的（可编辑）名称
* （B）动画剪辑时间线
* （C）剪辑属性来控制循环和姿势
* （D）可扩展部分：定义曲线，事件，蒙版和运动根节点; 并查看来自导入过程的消息



单个动画剪辑的属性选项

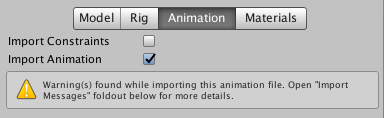
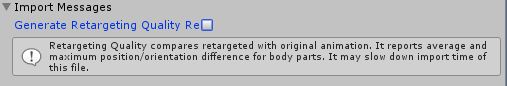
**Animation标签下的单个动画片段选项：**

| **属性:** | | **功能:** |
| --- | --- | --- |
| **Area A (editable name) A区域（动画名称）** | | |
| The name of the source take | | 将源文件作为此动画片段的源使用。 这是在Motionbuilder，Maya和其他3D软件中定义的一组动画。Unity可以将这些需要导入为单独的剪辑。您可以从整个文件或帧的子集创建它们。 |
| **Area B (timeline features) B区域（时间轴）** | | |
| The timeline editor | | 您可以使用拖动时间线周围的开始和结束指示符来为每个剪辑定义帧范围。 |
| **Start （开始）** | | 开始帧 |
| **End （结束）** | | 结束帧 |
| **Area C (looping and pose control) C区域（循环和姿态控制）** | | |
| **Loop Time（循环时间）** | | 循环播放动画片段 |
| **Loop Pose（循环姿势）** | | 无缝循环 |
| **Cycle Offset（周期偏移）** | | 如果它在不同的时间开始，则偏移循环动画的循环。 |
| **Root Transform Rotation（根Transform旋转）** | | |
| **Bake into Pose（烘焙成动作）** | | 骨骼的运动烘焙根旋转。禁用存储为根运动。 |
| **Based Upon（基础）** | | 根部旋转的基础。 |
| **Original（原始的）** | | 使用原始的旋转。 |
| **Root Node Rotation（根节点旋转）** | | 保持上半身朝前。 仅适用于**Generic**  动画类型。 |
| **Body Orientation（身体朝向）** | | 保持上半身朝前。 仅适用于**Humanoid** 动画类型。 |
| **Offset（偏移）** | | 到根旋转的偏移（以度为单位） |
| **Root Transform Position (Y) （Y方向的根位置）** | | |
| **Bake into Pose（烘焙成动作）** | | 将垂直根部运动烘焙到骨骼的运动中。禁用为根运动存储。 |
| **Based Upon (at Start) （基础（开始的时候））** | | 垂直根部位置的基础。 |
| **Original（原始的）** | | 保持源文件的垂直位置。 |
| **Root Node Position（根节点位置）** | | 使用垂直方向根位置。 仅适用于**Generic** 动画类型。 |
| **Center of Mass（重心位置）** | | 保持重心与根变换位置对齐。 仅适用于**Humanoid** 动画类型。 |
| **Feet（脚）** | | 保持脚与根转换位置对齐。 仅适用于**Humanoid** 动画类型。 |
| **Offset（偏移）** | | 偏移到垂直根位置。 |
| **Root Transform Position (XZ) （XZ方向的根位置）** | | |
| **Bake into Pose（烘焙进动作）** | | 烘烤水平的根部运动进入骨骼的运动。禁用为根运动存储。 |
| **Based Upon（基础）** | | 水平根位置的基础。 |
| **Original（原始的）** | | 使用源文件的水平位置。 |
| **Root Node Position（根节点位置）** | | 使用水平根变换位置。 仅适用于**Generic** 动画类型。 |
| **Center of Mass（重心）** | | 保持与根变换位置对齐。 仅适用于**Humanoid** 动画类型。 |
| **Offset（偏移）** | | 偏移到水平根位置。 |
| **Mirror（镜像）** | | 在此剪辑中左右镜像。 只有在动画类型设置为**Humanoid模式时**才会出现。 |
| **Additive Reference Pose（添加到参照动作）** | | 启用为用作添加动画层基准的参照动作设置框架。在时间线编辑器中可以看到蓝色标记：https://docs.unity3d.com/2018.1/Documentation/uploads/Main/AnimationAdditiveReferencePoseTimelineMarker.png |
|  | **Pose Frame（动作框架）** | 输入一个帧数字作为参照动作。您也可以在时间线中拖动蓝色标记以更新此值。 仅在**Additive Reference Pose**启用时才可用。 |
| **Area D (expandable sections) D区域（扩展部分）** 详见后续章节 | | |

创建剪辑基本上定义了动画片段的开始点和结束点。为了使这些剪辑循环，应该尽量修剪它们，以便尽可能匹配所需循环的第一帧和最后一帧。

##### 动画导入警告

如果在动画导入过程中发生任何问题，动画导入检查器的顶部会显示警告：

动画导入警告消息

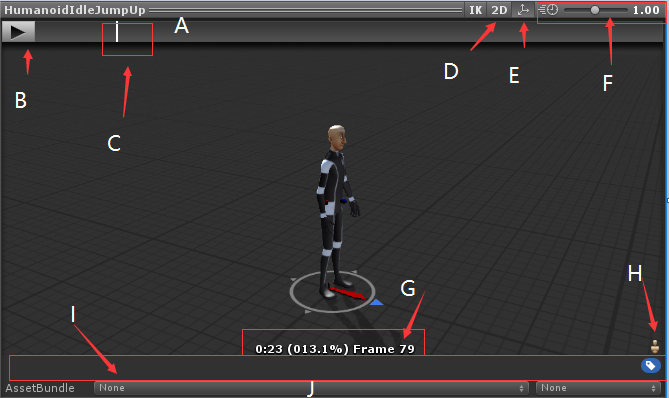
警告并不一定意味着您的动画尚未导入或无法正常工作。这可能意味着导入的动画看起来与源动画略有不同。

在这种情况下，Unity提供了“**Generate Retargeting Quality Report”**选项，您可以通过该选项查看有关重定目标问题的更多具体信息。如：

* 此文件中的默认骨骼长度与在源Avatar中找到的不同。
* 在此文件中找到的骨骼默认旋转不同于源Avatar中找到的旋转。
* 源Avatar层次结构与此模型中找到的不匹配。
* 这部动画有位移，将被丢弃。
* 人形动画具有中间变换和旋转，将被丢弃。
* 具有将被丢弃的缩放动画。

这些消息表明，当Unity导入动画并将其转换为其内部格式时，原始文件中的某些数据会被省略。这些警告基本上告诉您，重定向动画可能不完全匹配源动画。

##### 动画预览



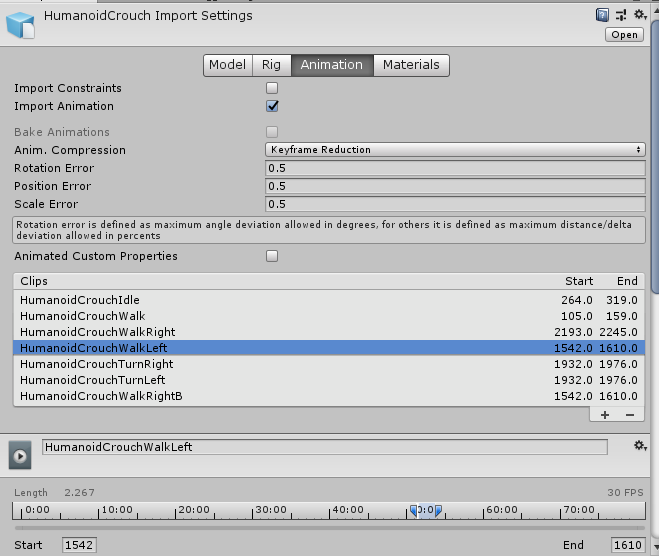
动画预览

* （A）所选剪辑的名称
* （B）播放/暂停按钮
* （C）预览时间线上的播放头（允许前后擦洗）
* （D）2D预览模式按钮（在正交和透视相机之间切换）
* （E）枢轴和质心显示按钮（在显示和隐藏小发明之间切换）
* （F）动画预览速度滑块（向左移动减慢;向右加速）
* （G）播放状态指示器（以秒，百分比和帧数显示播放位置）
* （H）Avatar选择器（更改哪个GameObject将预览动作）
* （I）**标签**栏，您可以在其中定义标签并将其应用于剪辑
* （J）AssetBundles栏，您可以在其中定义AssetBundles和Variants

##### 提取动画剪辑

动画角色通常具有许多不同的动作，这些动作根据游戏不同的情况被激活，称为动画剪辑。例如，我们可能会为走，跑，跳跃，投掷和死亡分别制作动画剪辑。根据美术在3D建模软件中设置动画的方式，这些单独的动作可能会作为不同的动画片段导入，或者作为一个单独的片段导入，其中每个动作只是从前一个动画开始。在只有一个长剪辑的情况下，您可以在Unity内提取动画片段。

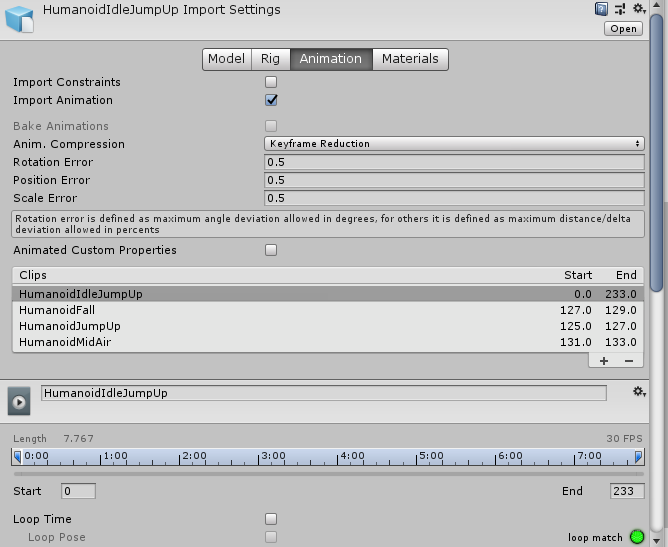
如果您的模型具有多个已经定义为单个片段的动画，则动画选项如下所示：



拥有多个动画的模型

您可以预览出现在列表中的任何剪辑。如果需要，您可以编辑剪辑的时间范围。

如果您的模型将多个动画作为一个连续的动画提供，则**动画**选项如下所示：



一个长动画片段的模型

在这种情况下，您可以定义与每个单独的动画序列（步行，跳跃，跑步和空闲）相对应的时间范围（帧或秒）。您可以按照以下步骤创建新的动画片段：

1. 点击add（+）按钮。
2. 选择它包含的帧或秒范围。
3. 您还可以更改剪辑的名称。

例如，您可以定义以下内容：

* 在帧0 - 83期间空闲动画
* 在帧84 - 192期间跳转动画
* 在第193-233帧期间摆动手臂动画

##### 使用多个模型文件导入动画

导入动画的另一种方法是遵循Unity允许的动画文件的命名方案。您可以创建单独的模型文件并使用约定命名modelName@animationName.fbx。例如，对于一个名为模型goober，您可以导入独立的空闲，走，跳，使用指定的文件walljump动画goober@idle.fbx，goober@walk.fbx，goober@jump.fbx和goober@walljump.fbx。当像这样导出动画时，没有必要在这些文件中包含Mesh，但在这种情况下，您应该启用Preserve Hierarchy Model导入选项。

统一自动导入所有四个文件，并收集所有的动画，而无需在@符号的文件，在上面的例子中，统一导入goober.mb文件，对引用idle，jump，walk和wallJump自动动画。

对于FBX文件，可以将模型文件中的网格导出而不使用动画。然后goober@\_animname\_.fbx通过导出每个所需的关键帧（在FBX对话框中启用动画）来导出四个剪辑。

##### 动画片段的循环优化

处理动画的人员的一项常见操作是确保它们正确循环。例如，如果角色沿着路径行走，则行走动作来自动画剪辑。该动作可能只持续10帧，但该动作连续循环播放。为了使步行运动无缝，它必须以类似的姿势开始和结束。这确保没有脚滑动或奇怪的生涩运动。

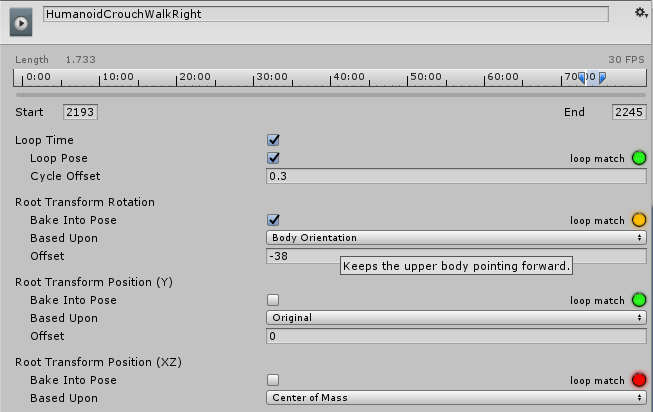
动画片段可以在姿势，旋转和位置上循环播放。使用步行循环的示例，您希望Y中的根变换旋转和Root Transform Position的起点和终点匹配。您不想在XZ中匹配Root Transform Position的开始点和结束点，因为如果它的双脚继续回到它们的水平姿态，您的角色将永远不会到达任何地方。

Unity在动画标签上的特定于剪辑的导入设置下提供了匹配指示符和一组特殊的循环优化图。这些提供了视觉提示，可以帮助您优化每个值的剪辑位置。

为了优化循环运动是否开始并最优地结束，可以查看和编辑循环匹配曲线。

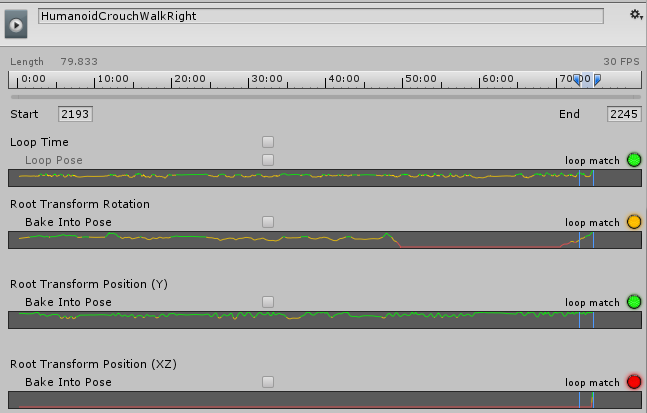
##### 查看循环优化图

在此示例中，循环移动显示剪辑范围的错误匹配，如红色和黄色指示符所示：



红色和黄色指标显示循环的不匹配

要查看循环优化图，请单击并按住时间轴上的开始或结束指示符。所述**基于**和**偏移**值消失并出现每个循环基础一条曲线：



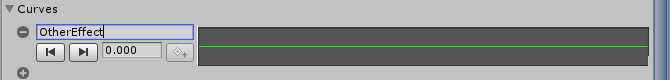
不匹配的循环图

单击并拖动动画片段的起点或终点，直到该点出现在属性为绿色的图形上。Unity以绿色绘制图形，更有可能剪辑可以正常循环。放开鼠标按钮时，图表消失，但指示器仍然存在。

##### 曲线

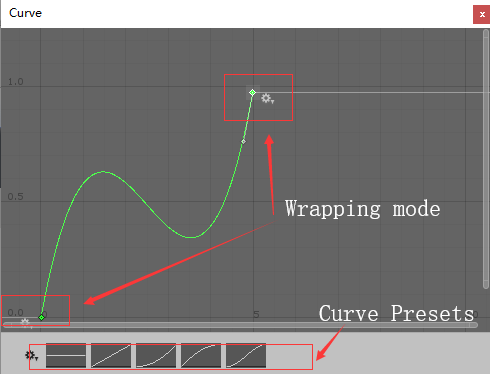
您可以在Animations标签中将动画曲线附加到导入的动画剪辑，可以使用这些曲线将其他动画数据添加到导入的动画片段。可以使用该数据根据动画师的状态动画制作其他项目的时间。例如，在冰冷条件下的游戏中，可以使用额外的动画曲线来控制粒子系统的发射速率，以显示玩家在冷空气中的冷凝呼吸。

要为导入的动画添加曲线，请展开“Animatios” 标签底部的“**Curves”**部分，然后单击加号图标以向当前动画片段添加新曲线，如果导入的动画文件分割为多个动画片段，则每个片段都可以有自己的自定义曲线。



动画剪辑的曲线

曲线的X轴表示标准化时间，并且始终在0.0和1.0之间（分别对应于动画片段的开始和结束，而不管其持续时间）。



Unity曲线编辑器

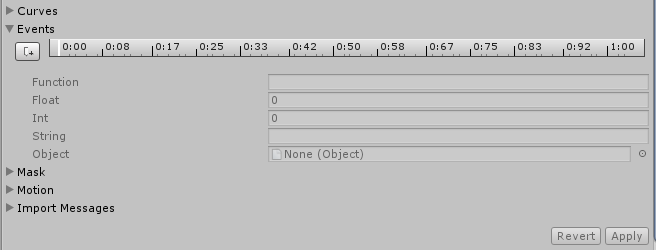
双击动画曲线带来了Unity的标准曲线编辑器，您可以用它来添加**Keys**。Keys是曲线时间轴上的点，它具有由动画制作者明确设置的值，而不仅仅是使用插值。按键对于沿着动画的时间轴标记重要点非常有用。例如，通过步行动画，可以使用键来标记左脚在地面上的点，然后双脚在地面上，右脚在地面上，等等。一旦设置了按键，您可以通过按下上**一个关键帧**和**下一个关键帧**按钮在关键帧之间方便地移动。这将移动垂直红线并显示标准化时间在关键帧。您在文本框中输入的值将驱动当时的曲线值。

##### 动画事件

可以将动画事件附加到动画标签中的动画片段。

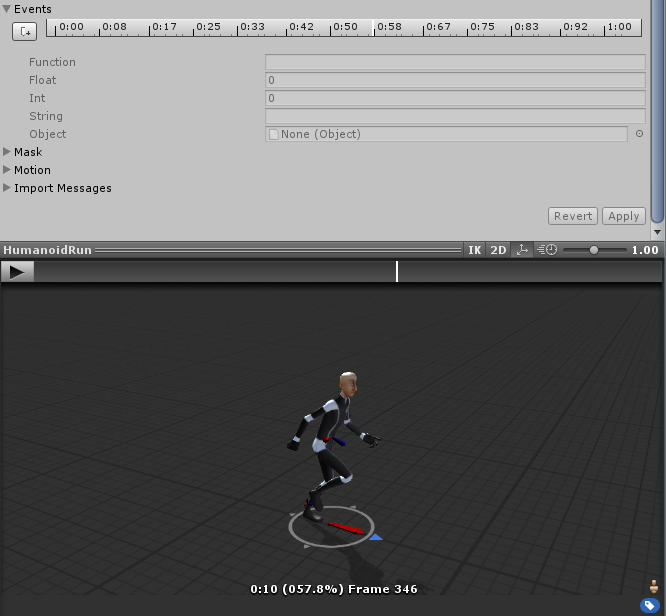
事件允许您将附加数据添加到动画剪辑，该剪辑确定何时应该在动画中及时发生某些操作。例如，对于动画角色，您可能需要添加事件来散步和运行循环，以指示足迹声音何时播放。

要将事件添加到导入的动画，请展开事件部分以显示动画片段的事件时间轴：



添加事件的动画时间轴

要将播放标识移动到时间轴中的其他位置，请在窗口的预览窗口中使用时间轴：

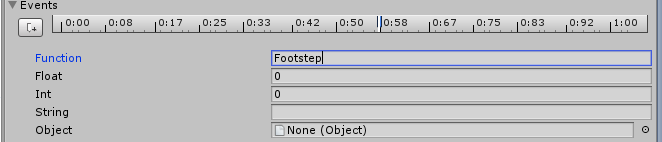


点击预览窗口上的时间轴可以控制事件的位置

将播放标识放在要添加事件的位置，然后单击**添加事件**。出现一个新事件，由时间线上的小白色标记表示。在**Function**属性中，填写要在触发事件时调用的函数的名称。

确保在其动画控制器中使用此动画的任何GameObject都附带了相应的脚本，该脚本包含具有匹配事件名称的函数。

下面的示例演示了一个设置为Footstep在连接到Player GameObject的脚本中调用函数的事件。这可以与AudioSource结合使用来播放与动画同步的脚步声音。



新增事件“Footstep”

您也可以选择指定要发送到事件调用的函数的参数。有四种不同的参数类型：**Float**，**Int**，**String**或**Object**。

通过在其中一个字段中填写值并实现您的函数以接受该类型的参数，可以将事件中指定的值传递给脚本中的函数。

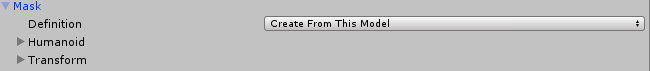
例如，您可能需要传递一个浮点值来指定在不同动作中脚步应该多大，例如步行循环上的安静脚步事件和正在运行的循环上的大声脚步事件。您还可以传递对Prefab效果的引用，从而允许脚本在动画过程中的某些点实例化不同的效果。

#### Mask遮罩

遮罩可以放弃剪辑中的一些动画数据，允许动画片段仅为对象或角色的一部分而不是整个事物制作动画。如果您有一个带有投掷动画的角色。如果您希望能够将投掷动画与各种其他身体动作（例如跑步，蹲伏和跳跃）结合使用，则可以为投掷动画创建一个遮罩，将其限制在右手臂，上身和头部。这部分动画可以在基本跑步或跳跃动画顶部的一层上进行。

导入时遮罩也可以缩小文件大小和内存，它还提高了处理速度，因为在运行时混合的动画数据较少。在某些情况下，导入时遮罩可能不合适，这时可以使用Animator控制器的图层设置在运行时应用遮罩

要将遮罩应用于动画片段，请展开Mask以显示Mask选项。当您打开菜单时，您会看到三个选项：Definition, Humanoid 和Transform。



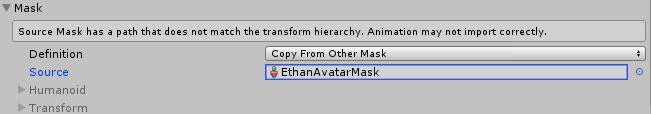
Mask的选项

允许指定是否要在Inspector中专门为此动画片段创建一次性遮罩，或者是否要使用项目中现有的遮罩资源。

如果您想为此剪辑创建一次性蒙版，请选择Create From This Model。

如果要使用相同的蒙版设置多个剪辑，则应选择Copy From Other Mask并使用Mask资源。这允许您为多个动画剪辑重新使用单个Mask的定义。

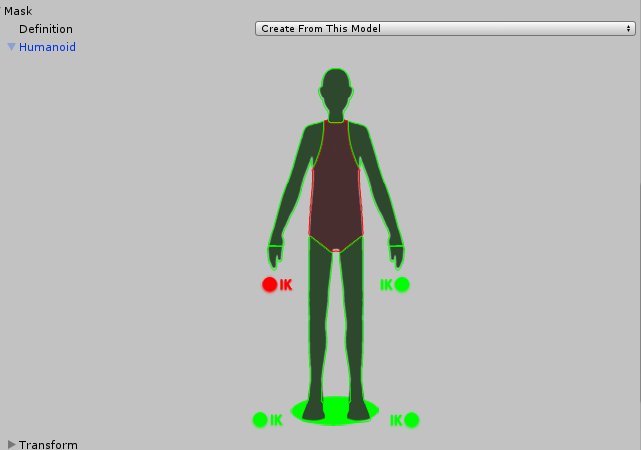
当选择Copy From Other Mask时，Humanoid和Transform选项不可用，因为这些选项与在此剪辑的Inspector中创建一次性遮罩有关。



Copy From Other Mask

##### 人形遮罩

人形选项为您提供了一种通过选择或取消选择人体的身体部位来定义Mask的快速方法。如果动画已被标记为人形并且具有有效的Avatar，则可以使用这些。



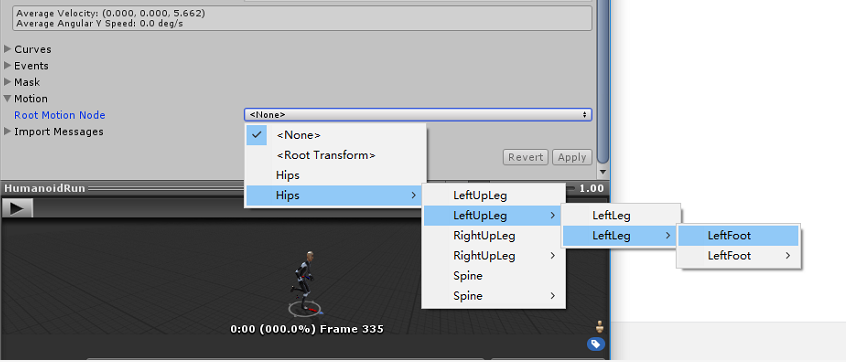
人形遮罩选项

##### 动作

当导入的动画片段包含根动作时，Unity将使用该运动来驱动正在播放动画的GameObject的移动和旋转。然而，有时可能需要在动画文件的层次结构中手动选择不同的特定节点来充当根动作节点。

“Motion”字段允许您使用分层弹出菜单在导入的动画的层次结构内选择任何节点（Transform），并将其用作根动作的源。该对象的动画位置和旋转驱动GameObject播放动画的动画位置和旋转。

要为动画选择根动作节点，请展开Motion部分以显示根动作节点菜单。当您打开菜单时，会显示导入文件层次结构根目录中的所有对象，包括None和Root Transform。这可能是您角色的网格对象，以及它的根骨名称，以及每个具有子对象的项目的子菜单。每个子菜单还包含子对象本身，以及其他子菜单（如果/这些/对象具有子对象）。



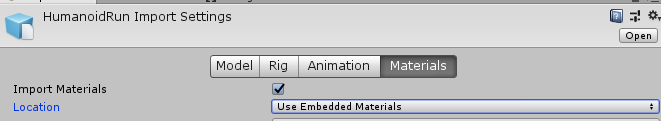
遍历对象的层次结构以选择根动作节点

一旦选择了根动作节点，对象的动画就会驱动它的运动。

#### 材质标签

此标签可以更改Unity在导入模型时如何处理材质和纹理。

当Unity导入没有分配任何材质的模型时，它使用Unity漫反射材质。如果模型具有材质，则Unity将它们作为子资源导入。您可以使用“**Extract Textures”**按钮将嵌入纹理提取到项目中。



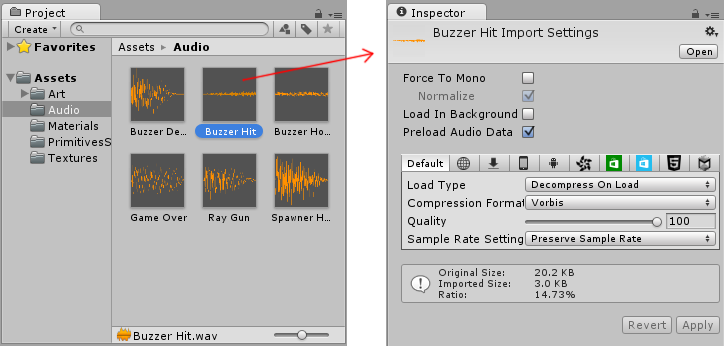
材质标签

**材质标签选项：**

| **属性** | 功能 |
| --- | --- |
| **Import Materials（导入材质）** | 启用导入材质的设置。 |
| **Location（位置）** | 如何访问材质和纹理。根据您选择的这些选项中的哪一个，可以使用不同的属性。 |
| **Use Embedded Materials（使用内嵌材质）** | 选择此选项可将导入的材质保留在资源中。这是Unity 2017.2版本后的默认选项。 |
| **Use External Materials (Legacy) （使用外部材质（老版本））** | 选择此选项可将导入的材质提取为外部资源。这是一种处理材质的老方式，适用于使用2017.1版本及以前的项目。 |

## 声音资源导入设置

对于其他的资源类型来说，导入设置都有所不同。您所看到的导入设置取决于您选定的资源类型。下面是一个音频资源的例子，右侧是其相关导入设置选项。



音频资源的导入设置

如果您正在开发一个跨平台的项目，则可以重写默认设置，为每一种平台设置一种导入设置。

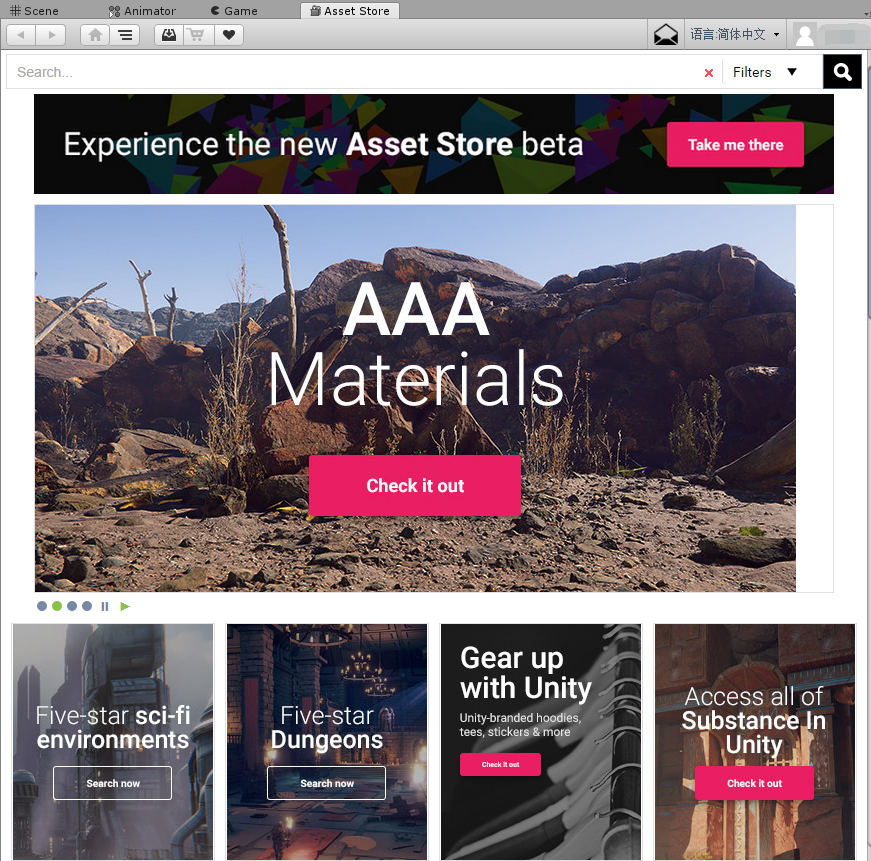
## 从资源商店导入资源

Unity**资源商店**是一个成长中的免费和商业资源库，它由Unity公司和许多社区成员提供的免费或商业资源。您可以找到各种可用资源，包括各种贴图、模型、动画以及完整项目实例，还有教程和编辑器扩展。这些资源通过Unity编辑器的一个简单界面访问下载并直接导入您的项目。

Unity用户可以在资源商店成为发布者，出售他们创建的内容。

### 进入资源商店和选购

您可以通过在主菜单选择 **Window -> Asset Store**打开资源商店。在您第一次访问时,将会被提示创建一个免费账户用于访问商店。



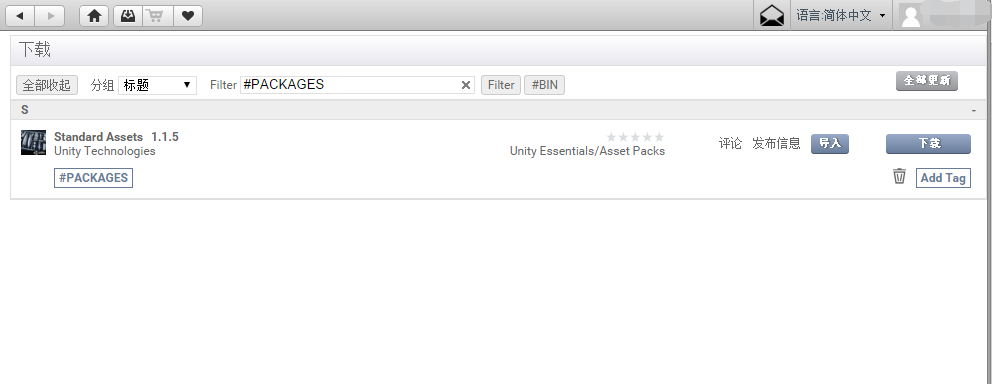
资源商店首页

商店提供了一个类似浏览器的界面，您可以通过自由输入文本搜索或浏览软件包和各种类别进行选择。在主工具栏左侧是浏览按钮用于查看浏览历史，它们的右侧则是查看下载管理器和查看购物车当前内容的按钮以及您收藏的内容按钮。

导航按钮 下载与购物车按钮

您可以通过下载管理器查看购买的包和查找安装任何更新。另外，Unity提供的标准包也可以通过相同的界面查看并添加到您的项目中。



下载管理器

### 下载资源文件位置

一般很少直接访问从资源商店下载的文件，如果需要的话，您可以在下面路径找到它们：

* MacOS: ~/Library/Unity/Asset Store
* Windows: C:\Users\accountName\AppData\Roaming\Unity\Asset Store

这些文件夹包含Asset Store发布者相对应的子目录，实际的资产文件则在相应的子目录中。

## 资源包

Unity包是共享和复用Unity项目和资源集合的一种简便方法。例如Unity标准资源和Unity资源商店上的项目都是以包的形式提供的。

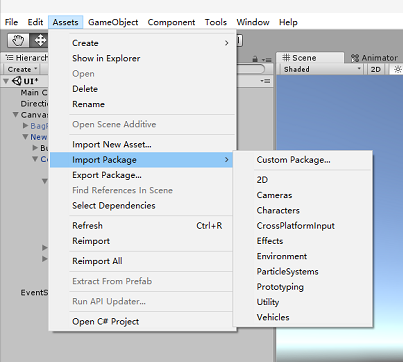
**包**是Unity项目或项目元素的文件和数据的集合，它们被压缩并存储在一个文件中，类似于Zip文件。与Zip文件一样，当它打开时包保持其原有的目录结构和关于资源的元数据(如导入设置和其他资源的链接)。

在Unity中，菜单选项 **Export Package** 压缩和存储资源集，**Import Package**解压包的资源集到您当前打开的项目中。

### 导入包

您可以导入**标准资源包**，它们是由Unity重新制作和提供的资源集，和由Unity用户制作的**自定义包**。

选择**Assets -> Import Package**来导入这两种类型的包.



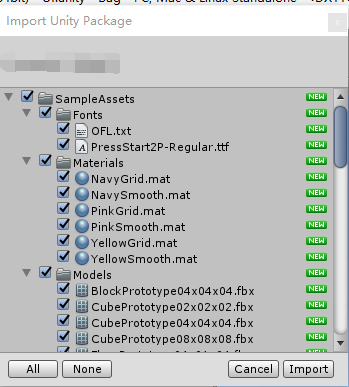
Asset -> Import Package menu

#### 标准资源包

Unity标准资源由多个不同包组成：2D、摄像机、人物、跨平台输入、效果、环境、粒子系统、原型、工具、交通工具。

导入一个新的标准资源包:

1. 打开您想导入资源的项目。
2. 选择 Assets -> Import Package加上您想要导入的包名，然后显示导入Unity包对话框即，预览检测包的所以项目，准备安装。（请参见下图：新安装导入Unity包对话框。）
3. 选择导入，Unity将把包的内容放入标准资源文件夹，您可以在项目视图上访问该文件夹。



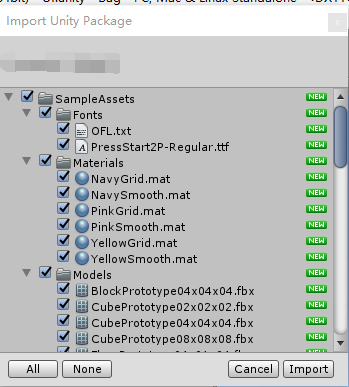
新安装导入Unity包对话框

#### 自定义包

您可以导入从自己项目或别人项目制作导出的自定义包.

导入一个新的自定义包：

1. 打开您要导入资源的项目。
2. 选择 **Assets -> Import Package -> Custom Package…** 打开Windows或者Mac文件管理系统。
3. 在Windows或Mac文件管理系统选择您要导入的包，显示**导入Unity包**对话框，并预览检查包的所以资源，然后准备安装。（请参见下图：安装导入Unity软件包对话框。）
4. 选择**导入**，Unity将把包的内容放入**资源文件夹**，您可以在项目视图上访问它们。



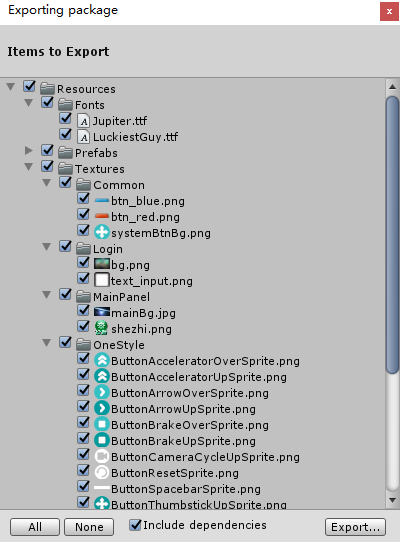
安装导入Unity包对话框

### 导出包

使用**Export Package**选项创建您自己的**自定义包**:

1. 打开您要导出资源的项目。
2. 选择**Assets -> Export Package…**从这种菜单调出**导出包**对话框。（请参见下图：导出包对话框。）
3. 在此对话框，检查并选择勾选您要导出的资源的复选框使其包含在包内。
4. 选中**include dependencies**复选框将自动选择对您选择的资源有用的任何其他资源。
5. 点击**Export** 调出Windows或Mac的文件管理系统，选择您要存放的包的位置。命名然后保存包到任何您想保存的路径下。

**提示：**当导出包时，Unity能同时导出所有依赖项。所以，例如如果您选择一个场景并导出包含的所有依赖项，那么在场景中出现的所有模型、贴图纹理以及其他资源也将同时被导出。这可以快速导出一堆资源，而无需手动查找相关依赖资源。



导出资源对话框

### 导出更新包

有时您可能想更改一个包的内容而创建一个新的，来更新您的资源包版本。其做法如下：

* 在包中选择您想要的资源文件(新增和未更改的文件皆选)。
* 导出文件如上文所述的**Export Package**。

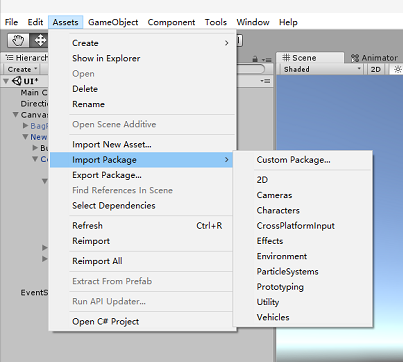
**注意：**您可以重命名一个更新包而且Unity也会识别它为一个更新，所以您可以用增量命名法，例如：MyAssetPackageVer1，MyAssetPackageVer2 (不推荐使用中文命名)。

**提示：**将文件从包中删除并用相同名称命名它们不是一个好习惯：Unity将把它们识别为不同和可能有冲突的文件，在导入它们是会显示警告。如果您已经删除一个文件然后决定替换它，最好给它一个跟原始文件相关但是不同的名字。

## 标准资源

Unity本身包含多种**标准资源**。它们囊括了Unity用户使用最广的一些资源。如：2D、摄像机、角色、跨平台输入、效果、环境、粒子系统、原型、工具、车辆。

Unity使用**Unity包**导入导出标准资源到项目中。



Asset -> Import Package目录

上图为导入标准资源的操作方法。如果您在安装Unity时选择不安装标准资源，您可以从资源商店下载它们，关键字为“Standard Assets”。