



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

# 关于Cinemachine

Cinemachine是Unity的一款模块化相机工具套件，可为您项目中的每台相机提供AAA游戏质量控制。这是一个易于安装的插件，可以让您为已有的相机添加功能，或者以令人惊叹的行为制作新的功能。

Cinemachine已被设计为您项目中的整个统一相机系统，但也可以与现有相机一起使用。如果你已经有一堆相机工作，并且只是想用Cinemachine来制作过场动画或其他特定的东西，那完全没问题。但是，当您在整个项目中使用它时，它可以让您将任何相机以任何其他相机的方式融入游戏玩法 - 过场动画 - 后退 - 无缝的方式。

## 要求

Cinemachine没有外部依赖性。只需安装并开始使用它。如果您还使用后处理堆栈（版本1或2），则会提供适配器模块 - 由ifdefs保护，该ifdefs会自动定义是否检测到后处理堆栈的存在。

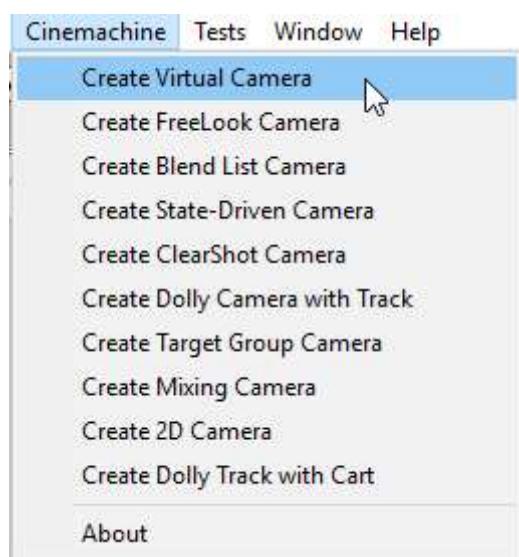
此Cinemachine 2.1.9版与以下版本的Unity编辑器兼容：

- 2017.1及更高版本（推荐）

# 使用Cinemachine

## 创建你的第一个虚拟相机

转到Unity Main菜单，然后选择Cinemachine / Create Virtual Camera。这也是创建额外Cinemachine镜头的方法。



你会注意到你场景的两个变化：



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼



1. 主相机在场景层次结构中旁边有一个小图标。这表示您的相机已使用Cinemachine启用了一个新组件：CinemachineBrain。
2. 带有CinemachineVirtualCamera组件的新GameObject CM vcam1，它是单个相机或镜头。

这两件事是Cinemachine的核心；一切都建立在他们身上。

## Cinemachine虚拟相机

虚拟相机是一个镜头，一个相机。它有孩子的组成部分，你可以打开它来做一些事情，例如程序性地跟踪和组成物体，跟随事物并用程序噪音摇晃。他们在一起是一个非常强大的组合，可以产生无限数量的结果。

这些模块经过设计，重新设计和重新设计，以最少的控制数量提供最广泛的可能性。驾驶这些相机行为的数学是复杂而复杂的，在各种游戏中经历了许多不同的场景。

### 为什么虚拟相机？

“虚拟”相机场景的原因是因为Unity从主相机渲染。开箱即用的Unity一次仅限于一台摄像机作为主动摄像机，这使得两个镜头的混合变得不可能。Cinemachine专为解决这个问题而设计，并同时提供丰富简单而强大的功能，从而实现无与伦比的相机行为，这些行为既快速又易于设置。

## Cinemachine大脑

这是Cinemachine的核心组件，它可以完成所有的魔术。它是Unity摄像头的一个组件，它监视场景中的所有虚拟摄像机，并使用它最喜欢的镜头驱动Unity摄像机。

会发生什么是只有一个Unity相机。根据大脑正在观察的虚拟相机提供的说明，安装在其上的大脑负责定位和配置一台相机在正确的时间在正确的位置。

以下有关如何使用这些新事物的更多信息。不要忘记查看Cinemachine / Examples / Scenes中包含的示例场景，请访问我们的用户论坛，网址 (<https://forum.unity3d.com/forums/cinemachine.136/>) 为 <https://forum.unity3d.com/forums/cinemachine.136/> (<https://forum.unity3d.com/forums/cinemachine.136/>)

## 经常问的问题

**我们是否必须用Cinemachine更换我们的相机系统？** 不，Cinemachine可以与现有的相机系统一起工作，并且可以在两个系统之间来回无缝混合。如果你写了一些你喜欢的相机技术，但是想要Cinemachine来做别的，没问题，把它们混合在一起。

**是不是所有这些相机都沉重？** 不，实际上它非常轻。每台虚拟摄像机都有超低的开销 - 尽可能多地制作。如果您对性能过度敏感，您可以禁用所有相机，只需启用您希望在任何特定时间运行的相机即可获得极佳性能。

**它是怎么运行的？** Cinemachine的作品有点像您现有的主摄像头的牵线木偶系统。它不创造任何新的相机，只有新的相机。Cinemachine的“大脑”组件会自动添加到场景中的主摄像头，Cinemachine会与其所有相机混合，优先级，混合魔术一起使用，以向主摄像头提供一组变换。因为它是一个完全统一的系统，所以您可以从任何相机类型混合到任何其他类型。无缝地从游戏中的freelook系统转移到关键帧过滤相机并返回。任何组合都是可能的，并且选项是无限的。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

**我是否可以将所有调音器保存在Play模式中？** 是！几乎Cinemachine中的所有内容都保存在播放模式中，因此可以在游戏运行时跳入并调整某些相机。当你考虑它时，它是真正做到这一点的唯一方法。我们希望您在游戏中随时随地进入该区域并按摩相机，同时您还可以在其中玩耍。

**我们的工程师花了很长时间在我们的相机系统上，并不想改变它。** 我们听到很多。抛开代码并不是件好事，但你需要考虑项目的大局。如果您能够快速创建和调整相机，而不会进一步困扰您的工程师，是不是一个优势？您怀疑现有的系统在每种情况下的表现如何，因此无论如何还有更多的编码需要完成。

相机是你项目中的重要组成部分，好的相机可以在好的游戏和惊人的相机之间找到差别。尝试在现有相机旁边运行Cinemachine，看看它是什么样子。如果那名工程师仍然不愿意与他们分享这个想法 - 火箭队将完美的助推器部分弹射到太空中。相机代码有助于推动你进入轨道！编辑室的地板上通常会留下很棒的场景。做出最适合您项目的决定。

**我可以在游戏和镜头相机之间进行混合吗？** 你可以在任何相机之间进行混合 从游戏到切换和后退的无缝转换就像打开摄像机或播放带有CM剪辑的时间线一样简单。

**你通常如何配置相机的项目？** 而不是一个大的代码相机试图做任何事情，你会为每一个特定的时刻制作大量的自定义摄像头，并在需要时打开它们。混合或切割到新相机与打开相机一样简单。如果优先级相同或更高，CM将使用新相机并根据您在主相机上CM'Brain'的自定义混合部分中定义默认混合或任何其他混合的方式进行混合。

**我不想在X场景中彻底改变相机，我只是想稍微调整一下** 相机的变化可以非常微妙。您可以复制主摄像机，只需更改FOV或合成图像，然后在玩家进入触发器音量时混合到该摄像机。你可以只用微妙的差异打开vcams，以在任何情况下获得你想要的行为。项目可以有很多vcams，只需稍作更改，随时准备并等待任何游戏场景需求。

**我什么时候可以使用状态驱动相机？为什么有这么多的摄像机？** SDC使得将动画和状态链接到不同相机变得非常容易。拍摄角色的本地运动系统和轨道摄像机。当人物跑步时，相机抖动一点点，相机在外面时拉回到更大的轨道，当人物潜入时相机全部关闭时，它会让您的游戏更加活跃。SDC允许您制作相机以最适合您的玩家正在做的事情。

例如，dev中的一个项目拥有20多个不同的摄像机设备，全都收听动画和游戏事件。站立，步行，冲刺，跳跃，攻击，> 3个敌人到达，内部，外部，触发地图特定区域的体积，健康<20%等。如此多的控制很容易实现。

## 用户手册

本指南以逻辑部分组织，涵盖从简单的基本设置到Cinemachine的高级影院功能。详细描述组成Cinemachine的每个类和行为。

Cinemachine绝不仅限于这里描述的类和行为。它具有开放的管道架构，我们期待看到我们的用户将构建到系统中的令人兴奋的新行为和智能。

## 基本的虚拟相机

每个虚拟相机都是一个“镜头”，或一台相机。您可以直接为它们制作动画或将两者混合在一起以创建相机移动。他们是‘虚拟’相机的原因是允许混合，基于优先级的相机选择以及让Cinemachine在内存中保存多个相机的能力，最终产品呈现给当前正在渲染的活动Unity相机。

Cinemachine允许您创建无限数量的虚拟相机，将它们融合在一起，并将结果呈现给单个Unity相机。它自动完成所有这些。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼



虚拟摄像机的关键组件是：

- 优先级**：这个镜头的优先级。与当前相机优先级相比，相同或更高值的相机优先级将被激活。这允许根据触发量，动画，健康状态等调用相机的相机状态机设置。重要的相机将触发较低优先级的相机。
- 镜头**：这种镜头使用什么样的镜头：FOV，裁剪平面和荷兰语（相机胶卷）。后期处理效果也可以包含在这里。
- 身体**：照相机放置在场景中的位置，以及将照相机“挂载”到任何物体并用它移动照相机的程序性方法。
- 目的**：摄像头所看到的，以及跟踪和撰写任何主题的程序方式。
- 噪音**：程序Perlin多层噪音系统适用于手持行为，摇晃和振动。
- 扩展**：这些是基本Cinemachine管道的附加组件。后处理和碰撞避免是Cinemachine附带的标准扩展的示例。该API是开放的，这样你也可以编写自己的扩展。

虚拟相机检查器面板中的**独奏按钮**将使虚拟相机暂时处于活动状态，以便在调整相机的行为参数时可以在游戏视图中立即获得视觉反馈。

该**游戏窗口指南**复选框切换任何屏幕组成的知名度引导虚拟摄像头可能要在游戏窗口中显示。此切换是一个静态全局，适用于所有虚拟摄像机。

将**保存在播放**复选框可服用而在播放模式下进行相机设置，调整，并传播他们回到现场Cinemachine功能。这可以在游戏中对摄像头进行微调，而无需进行大量的复制/粘贴操作。退出播放模式后，场景将会更新以反映播放过程中所做的更改。按撤销将恢复这些更改。

以下各节介绍组成Cinemachine虚拟相机的行为和类及其一些标准程序组件。

## CinemachineVirtualCamera

此组件旨在附加到空的Transform GameObject，并且它表示Unity场景中的虚拟相机。

虚拟摄像机将根据其CinemachineComponent管道（目标，正文和噪声）中包含的规则对其进行动画处理。当虚拟摄像机是Live时，Unity摄像机将假定虚拟摄像机的位置和方向。

虚拟相机不是相机。相反，它可以被认为是相机控制器，与摄影师不同。它可以驱动Unity Camera并控制其位置，方向，镜头设置和后期处理效果。每个虚拟摄像机都拥有自己的Cinemachine组件管道，通过它您可以提供动态跟踪特定游戏对象的说明。

虚拟相机非常轻便，不会自行渲染。它只是跟踪有趣的GameObjects，并相应地定位自己。一个典型的游戏可以有几十个虚拟相机，每个虚拟相机都有自己的属性，如名称，事件，碰撞事件等。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

- 活动**: 虚拟相机正在主动控制Unity相机。虚拟相机正在跟踪其目标并在每一帧进行更新。
- 待机**: 虚拟相机正在跟踪其目标并在每一帧都进行更新，但没有任何Unity相机正在被它控制。这是在场景中启用的虚拟相机的状态，但可能比Live虚拟相机的优先级低。
- 禁用**: 虚拟摄像机存在但在场景中禁用。它不积极跟踪目标，因此不消耗处理能力。但是，虚拟摄像头可以在时间线上直播。

Unity Camera可以由场景中的任何虚拟相机驱动。根据游戏逻辑，游戏逻辑可以通过操纵虚拟摄像头的启用标志及其优先级来选择虚拟摄像头进行实时拍摄。

为了由虚拟相机驱动，Unity相机必须具有CinemachineBrain行为，该行为将根据其优先级或其他标准选择最符合条件的虚拟相机，并将管理混合。

设置	类型	描述
看着	转变	相机想要看的对象（目标目标）。如果此值为null，则vcam的变换方向将定义相机的方向。
跟随	转变	相机想要移动的物体（人体目标）。如果此值为null，则vcam的变换位置将定义相机的位置。
位置混合	PositionBlendMethod	<p>提示将虚拟相机的位置和虚拟相机进行混合。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li><b>线性</b>: 标准线性位置混合。</li> <li><b>球形</b>: 关于LookAt目标位置的球形混合，如果有LookAt目标。</li> <li><b>圆柱体</b>: 如果有LookAt目标，则围绕LookAt目标位置进行圆柱体混合。垂直坐标是线性插值的。</li> </ul>
镜片	镜头设置	指定此虚拟摄像机的镜头属性。这通常反映了Unity摄像机的镜头设置，并将在vcam处于活动状态时用于驱动Unity摄像头。
优先	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## 镜头设置

描述相机的FOV和剪辑平面。这通常反映了Unity摄像机的镜头设置，并将在vcam处于活动状态时用于驱动Unity摄像头。

设置	类型	描述
视野	单	这是垂直度的相机视图。对于电影人来说，超35毫米传感器上的50毫米镜头等于19.6度FOV。
正交大小	单	当使用正交相机时，它定义了相机视图的世界坐标中的半高。
剪贴板附近	单	这定义了相机平截体的可渲染范围内的近区域。提高这个值将会阻止游戏在相机附近绘制东西，这有时候会派上用场。较大的值也会增加阴影分辨率。
远剪辑平面	单	这定义了相机平截体的可渲染范围的远端区域。通常，您希望将此值设置得尽可能低而不切断所需的远处物体。
荷兰人	单	相机Z以角度滚动或倾斜。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

## CinemachineComposer

这是组件管道目标部分中的CinemachineComponent。它的工作是将摄像头对准vcam的LookAt目标对象，并配置偏移，阻尼和合成规则。

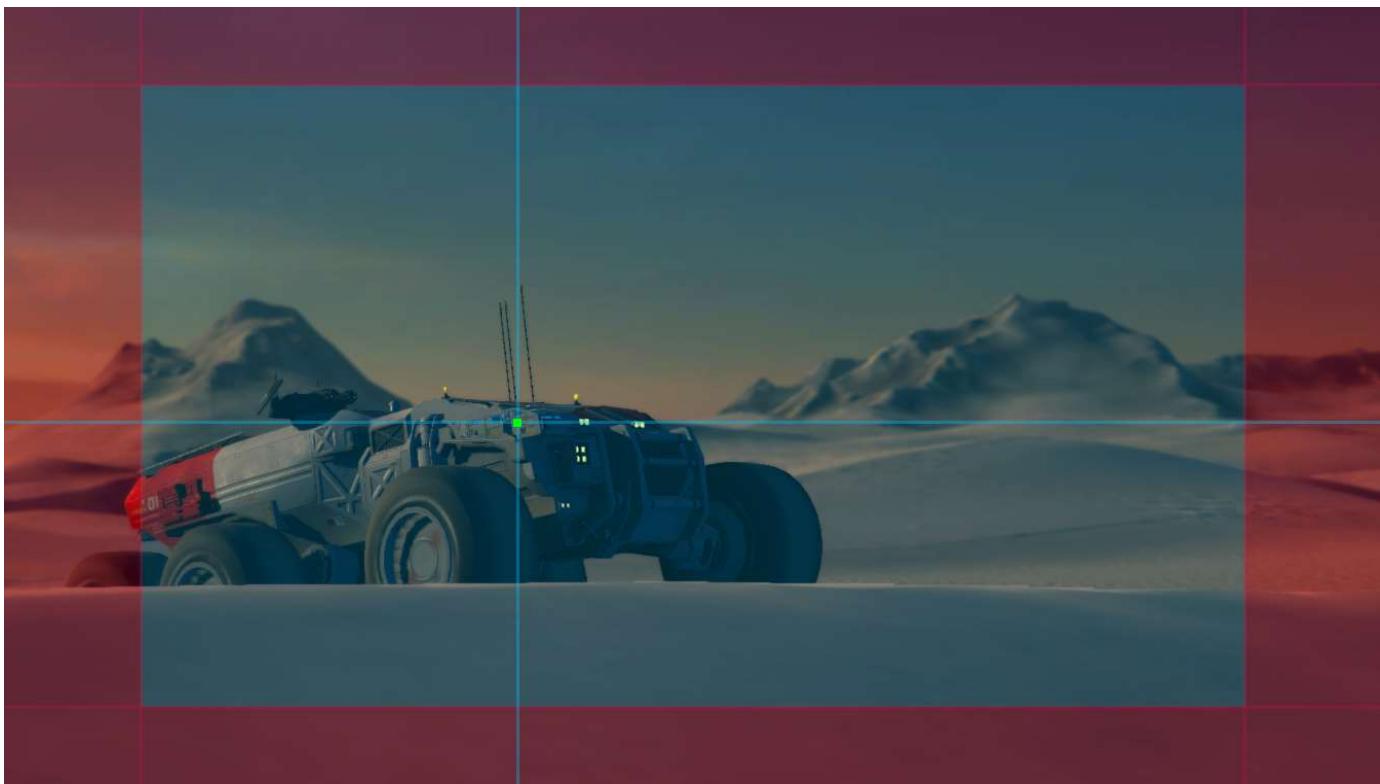
作曲家不改变相机的位置。它只能平移和倾斜摄像机，以获得所需的帧。要移动相机，您必须使用虚拟相机的“正文”部分。

vcam的LookAt目标是作曲家的目标。Common Composer的目标是：将骨骼包括在上部脊柱或头部骨骼等字符上，车辆或以编程方式控制或动画化的虚拟物体。

一旦您设置了vcam的LookAt目标并且正在追踪某个内容，您需要定义您希望它在屏幕上的位置，以及您希望摄像头追踪它的积极程度。滞后程度或相机跟随主体的程度由两个控件定义：

**水平/垂直柔和阻尼：**将这些值设置为零表示相机将难以跟踪拍摄对象，而上面的蓝色区域将用作锁定相机移动的刚性屏障，以将目标保持在这些值内。设置较大的值可以让目标“挤压”到蓝色区域，为您提供一些非常好的相机重量和延迟。这就是真正的摄像机操作员所做的！垂直和水平值是分开的，所以您可以让照相机从左向右挤压或按照上下拉紧等等。

**SoftZone / DeadZone：**对这些进行调整以定义要在屏幕上显示主题的位置。这些控件非常灵活。蓝色区域是基于您拥有多少水平/垂直阻尼的“松软”区域，红色区域定义了相机始终会跟踪拍摄对象的“不通过”限制。打开柔软区域以在中间创建一个“死区”，可以使屏幕区域免受目标运动的影响，适用于动画循环等情况，例如动画循环，如果摄像机不想跟踪目标一点。



设置	类型	描述
跟踪对象偏移	的 Vector3	目标在目标局部空间中偏离目标对象的中心。当期望的区域不是被跟踪物体的中心时，使用它来微调跟踪目标位置。
前瞻时间	单	此设置将指示作曲家根据目标的运动调整目标偏移量。作曲家将着眼于估计目标将在未来数秒内完成的一点。请注意，此设置对有噪音的动画很敏感，并且可能会放大噪音点，导致不合需要的相机抖动。如果摄像机在目标运动时抖动不合格，请关闭此设置，或者更平滑地为目标设置动画。
前瞻平滑	单	控制先行算法的平滑度。较大的值可以消除抖动预测并增加预测滞后时间。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
Lookahead 忽略Y.	布尔	如果选中，则预览计算将忽略沿Y轴的移动。
水平阻尼	单	相机尝试在屏幕水平方向上追踪目标的方式非常积极。小数字响应更快，快速定位相机以将目标保持在死区。数字越大，响应速度越慢。使用不同的垂直和水平设置可以产生各种相机行为。
垂直阻尼	单	相机尝试在屏幕垂直方向上跟随目标的方式有多强烈。小数字响应更快，快速定位相机以将目标保持在死区。数字越大，响应速度越慢。使用不同的垂直和水平设置可以产生各种相机行为。
屏幕X	单	目标的水平屏幕位置。相机将旋转以在此定位追踪对象。
屏幕Y	单	目标的垂直屏幕位置，相机将旋转以在此定位追踪对象。
死区宽度	单	如果目标位于此位置范围内，相机将不会水平旋转。
死区高度	单	如果目标位于此位置范围内，相机将不会垂直旋转。
软区宽度	单	当目标位于此区域内时，摄像机将逐渐水平旋转以根据阻尼速度重新对准所需的位置。
软区高度	单	当目标位于该区域内时，摄像机将逐渐垂直旋转，以根据阻尼速度重新对准所需的位置。
偏见X	单	非零偏差将使目标位置水平移动离开软区域的中心。
偏见Y.	单	非零偏置将使目标位置垂直移动离开软区中心。

## CinemachineTransposer

这是组件管道正文部分的CinemachineComponent。它的工作是将摄像机放置在与vcam的跟踪目标对象具有固定关系的位置，具有偏移和阻尼。

Transposer只会改变相机在太空中的位置。它不会重新定位或以其他方式瞄准相机。为此，您需要在管道的目标部分指示vcam。

Transposer是将摄像头安装到任何物体的组件。与将摄像机置于要让摄像机跟随的物体下方相比，它具有许多优点。遮光罩凸轮，门凸轮，POV凸轮，导弹凸轮 - 如果要将摄像机安装到某物上，请使用转接板。

使用Transposer进行相机身体运动具有以下优点：

- 即使游戏正在运行，位置调整也会保持不变。在游戏模式下调整普通相机，一旦退出，位置就会丢失，而不是转座子。在游戏运行时调整它，并完成。
- 为相机添加阻尼 - 如果您的相机遵循某些内容，阻尼值会给相机带来一定的“重量”，让它们感觉更平滑。
- 组织：将所有相机放在一个地方，而不是将它们隐藏在项目中的各种资产之下 - 选择目标与将相机置于层次结构下。

**按照偏移量：**相机将转到您在转接器相机目标中定位的对象的中心，因此如果您希望相机进一步落后，请输入一些偏移量。我们默认在-10，所以相机无论你的目标是什么。

**阻尼：**每通道阻尼会导致相机落后于目标。您可以控制位置阻尼和方向阻尼。

**绑定模式：**相机与目标对象之间存在关联。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

- 在目标分配时锁定目标分配在摄像机初始化时相对于目标的本地坐标系安装摄像机。这个偏移在世界空间中保持不变，并且相机不会与目标一起旋转。
- 无论目标物体朝向哪个方向，世界空间偏移量都会将相机相对于世界坐标系中的目标进行定位。
- 随着世界的简单跟随：这将跟随目标在所需的距离和高度，同时移动相机的位置尽可能少。这与人类摄像机操作员在指示跟踪目标时将执行的操作非常相似。

这些不同的模式做了令人难以置信的不同的事情，所以尝试一下，无论您的要求是什么，都应该很好地工作。

设置	类型	描述
绑定模式	BindingMode	<p>解释来自目标的偏移量时使用的坐标空间。这也用于设置相机的向上矢量，这将在瞄准相机时保持不变。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>锁定目标位置指定</b>：使用由虚拟相机启用时的目标本地框架组成的参照系，或者指定目标时，相机将被绑定到跟随目标。</li> <li>- <b>锁定目标与世界上</b>：相机将绑定到跟踪目标使用一个由目标的本地框架组成的参考框架，倾斜和横滚归零。</li> <li>- <b>锁定目标没有卷</b>：使用由目标本地框架组成的参照系将相机绑定到跟随目标，并将滚轮置零。</li> <li>- <b>锁定目标</b>：相机将使用目标的本地框架绑定到跟随目标。</li> <li>- <b>世界空间</b>：使用世界空间偏移将相机绑定到跟随目标。</li> <li>- <b>随着世界的简单追踪</b>：偏移量将使用相机局部轴相对于目标进行计算。</li> </ul>
按照偏移量	的Vector3	转换器将尝试从Follow目标保持的距离向量。
X阻尼	单	相机尝试保持X轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的x轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
Y Damping	单	相机尝试保持Y轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的y轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
Z阻尼	单	摄像机试图保持Z轴的偏移量。小数字响应更快，快速平移相机以保持目标的Z轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
音调阻尼	单	相机试图追踪目标旋转的X角度的积极性。小数字响应更快。数字越大，响应速度越慢。
偏航阻尼	单	相机尝试追踪目标旋转的Y角度的积极性。小数字响应更快。数字越大，响应速度越慢。
滚动阻尼	单	相机试图追踪目标旋转的Z角度的积极性。小数字响应更快。数字越大，响应速度越慢。

## CinemachineBasicMultiChannelPerlin

作为实现Noise阶段的Cinemachine Pipeline的一部分，此组件将Perlin Noise添加到CameraState的Correction通道中的Camera状态。

噪音是使用预定义的噪音档案资产创建的。这定义了噪音随时间的形状。您可以按幅度或时间对其进行缩放，以使用相同的配置文件生成大量不同的噪音。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

程序噪音是一个复杂的事情，让看起来真实。令人信服的手持运动是低频，中频和高频摆动的混合物，它们共同组成可信的东西。

**位置/方向大小：**这是您想要为位置或方向或两者混合多少个噪音功能。将至少3个方向通道混合在一起以实现一些逼真的手持动作。

**振幅**以度为单位定义噪音量。为了“看到”震动，更宽的镜头需要更大的度数值。远摄镜头使用较小的值，因为通过较窄的FOV镜头可以放大较小的设置。

**频率**以Hz为单位定义噪声的速度。通常，“低”频率值可能在0.1左右。考虑到你的游戏运行在30或60hz，因此高于此设置的设置将会在Nyquist频率的“另一侧”，这意味着它们不会被直接跟踪。100的设置将高于相机可以“追随”的水平，因为您的游戏仅在60hz运行。由于相机无法追踪比游戏运行速度更快的样品，因此它可能看起来有点波动。它也可以看起来很酷，但很少。实验。通常情况下，对于大多数手持式设置，低位约为0.1-0.5，中位数可能为0.8-1.5，高位约为3-4。这是3-4每秒来回摇动。

最令人信服的相机抖动通常是使用方向噪声完成的，因为这是相机瞄准的地方。手持式摄像机操作人员倾向于摇摆得比他们的位置更旋转，但当然可以随意混合某些位置噪音，请记住，最好从方向开始。

我们在Cinemachine / Examples / Presets / Noise下包含了许多预设，您可以根据需要添加尽可能多的预设，只需右键单击资产窗口Create-> Cinemachine-> Noise，然后将该资源拖到该虚拟摄像机下的“噪波设置”窗口中。您还可以通过幅度增益和频率增益设置使噪音动画，以上下调整效果。

设置	类型	描述
噪音档案	NoiseSettings	包含噪音配置文件的资产。定义那里的频率和幅度来制作特征噪声曲线。制作您自己的或仅使用众多预设中的一种。
幅度增益	单	获得应用于NoiseSettings资产中定义的幅度。1是正常的。将其设置为0可以完全消除噪音。
频率增益	单	比例因子适用于NoiseSettings资产中定义的频率。1是正常的。噪点越大，噪音震动越快。

## NoiseSettings

这是定义噪音配置文件的资产。噪声曲线是作为时间函数的噪音形状。您可以通过在不同幅度处组合不同基底Perlin噪音频率来构建任意复杂的形状。

应谨慎选择频率和幅度，以确保有趣的噪音质量，这种质量并不明显重复。

作为一个数学方面的说明，任何任意周期曲线都可以分解为一系列固定幅度的正弦波加在一起。这被称为傅里叶分解，并且是许多信号处理的基础。它与这项资产并无太大关系，但它超级有趣！

设置	类型	描述
位置	TransformNoiseParams[]	这些是虚拟相机位置的噪音通道。令人信服的噪音设置通常将低频，中频和高频混合在一起，所以从3开始。
方向	TransformNoiseParams[]	这些是虚拟相机定位的噪音通道。令人信服的噪音设置通常将低频，中频和高频混合在一起，所以从3开始。

## NoiseSettings.NoiseParams

描述噪音通道的行为。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
振幅	单	此通道的噪声幅度。更大的数字振动得更高。
频率	单	此频道的噪音频率。更高的数值振动得更快。

## NoiseSettings.TransformNoiseParams

包含相机所有3个主轴的噪声模块的噪声行为。

设置	类型	描述
X	NoiseParams	X轴的噪声定义。
和	NoiseParams	Y轴的噪声定义。
同	NoiseParams	Z轴的噪声定义。

## 大脑和混合

CinemachineBrain是Cinemachine的核心组件，可以完成所有的魔术。它是Unity摄像头的一个组件，它监视场景中的所有虚拟摄像机，并使用它最喜欢的镜头驱动Unity摄像机。

当从一个镜头转换到另一个镜头时，您可以选择如何完成此转换。它可以是相机切割或Cinemachine混合。Cinemachine混合物是特殊的：它们不是褪色，或擦拭或溶解。相反，它们是将一台相机的设置插值到另一台相机的设置。想像一下混合，就好像一个摄影师顺利地将摄像机传给另一个摄像师在不同的位置，以优雅的方式过渡镜头而没有切割。

## CinemachineBrain

CinemachineBrain是Unity Camera和场景中的Cinemachine虚拟相机之间的链接。它会监视优先堆栈以选择当前的虚拟相机，并在必要时与另一台相融合。最后也是最重要的一点，它将虚拟摄像机状态应用于所连接的Unity摄像机。

CinemachineBrain也是虚拟相机之间混合规则定义的地方。相机混合是一个虚拟摄像机位置和状态随时间的插值。如果您将虚拟相机视为摄影师，那么混合就像一位摄影师顺利地将相机传递给另一位摄影师。您可以指定混合时间以及混合曲线形状。请注意，相机切割只是零时间混合。

大脑拥有以下关键设置：

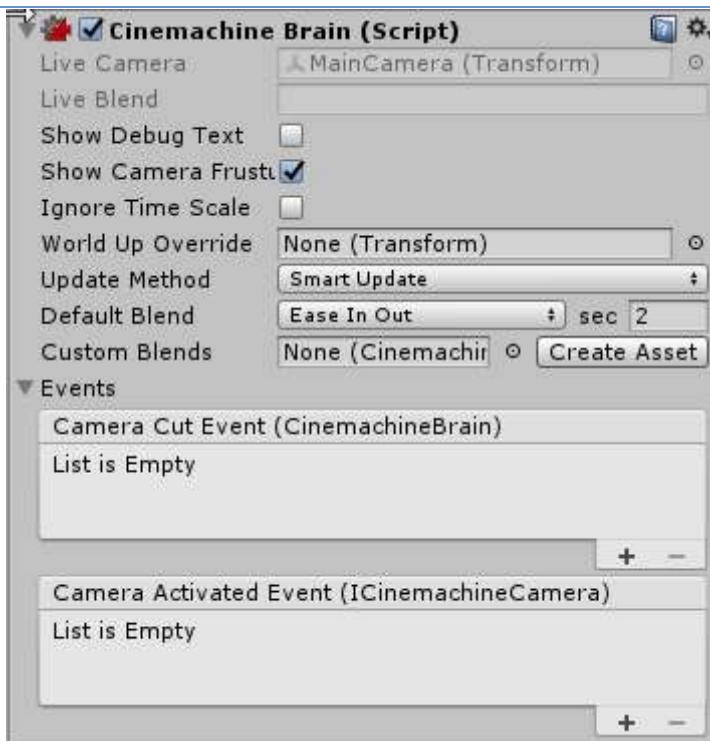
- Blend Settings (混合设置)**：这是定义任何镜头与任何其他镜头混合的阵列。从CameraA到CameraB可以有4秒的混合，但是从CameraB到CameraA的混合时间为1秒。这在使用状态机类型设置时非常强大。如果未定义两台相机之间的特定混合，则会使用默认混合，默认情况下是平滑的2秒混合（但您可以更改此设置）。
- 图层滤镜**：Brain只会注意那些通过其Unity摄像机剔除掩模的虚拟相机。您可以通过将不同的虚拟相机分配给不同的层来设置分屏环境，并使用剔除掩码对其进行过滤。
- 事件调度**：当击球发生变化时，大脑会触发事件，以防您需要监控这一点。在每次虚拟摄像机上线时发生的事件以及摄像机切断时的其他事件（以便可以重置时间后置效果）。



已翻译为以下语言: 中文(简体)

显示原文

选项 ▼



设置	类型	描述
显示调试文本	布尔	启用后，当前摄像机和混合将在游戏窗口中显示，以进行调试。
显示相机Fr	布尔	启用时，相机的平截头体将始终显示在场景视图中。
忽略时间比例	布尔	启用时，即使游戏以慢动作运行，相机也会始终实时响应用户输入和阻尼。
世界超越	转变	如果设置，该对象的Y轴将为所有虚拟摄像机定义世界空间向上矢量。这在例如自上而下的游戏环境中是有用的。如果没有设置，Up是世界空间Y.恰当地设置这一点非常重要，因为虚拟相机不喜欢直直看或直下。
更新方法	UpdateMethod	<p>如果您在固定更新期间（例如RigidBodies）为所有目标设置了动画效果，则在正常更新循环期间所有目标均为动画效果的LateUpdate;如果您希望Cinemachine在每个目标上执行相应操作，则使用SmartUpdate。SmartUpdate是推荐的设置。</p> <p><b>可能的值:</b></p> <ul style="list-style-type: none"> <li>- <b>修正更新:</b> 虚拟摄像机与固定更新中的物理模块同步更新。</li> <li>- <b>最新更新:</b> 虚拟摄像机在MonoBehaviour LateUpdate中更新。</li> <li>- <b>智能更新:</b> 根据目标更新的方式更新虚拟摄像机。</li> </ul>
默认混合	CinemachineBlendDefinition	在没有明确定义两台虚拟相机之间混合的情况下使用的混合。
自定义混合	CinemachineBlenderSettings	这是包含场景中特定虚拟相机之间混合自定义设置的资产。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
相机切割事件	BrainEvent	每当虚拟相机上线并且没有混合时，此事件就会启动。
相机激活的事件	VcamEvent	每当虚拟相机上线时，此事件都会触发。如果涉及混合，则该事件将在混合的第一帧上触发。

## CinemachineBlendDefinition

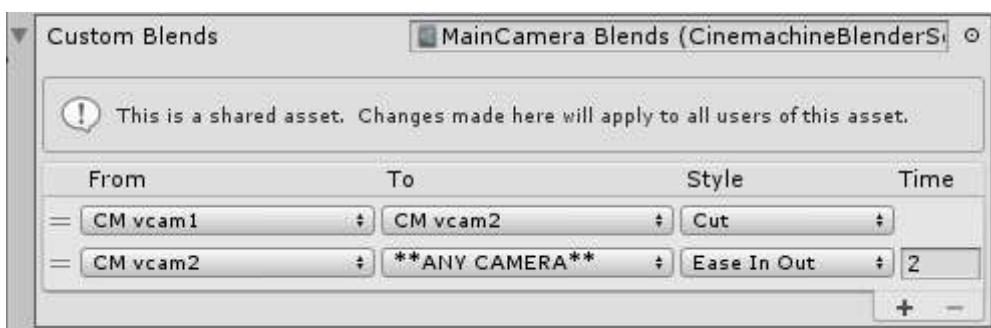
相机混合的定义。该结构保存为Cinemachine Blend生成合适的AnimationCurve所必需的信息。

设置	类型	描述
样式	样式	<p>混合曲线的形状。 可能的值：</p> <ul style="list-style-type: none"> <li>- <b>切割</b>: 零长度混合。</li> <li>- <b>轻松输入</b>: S形曲线，平滑过渡。</li> <li>- <b>轻松输出</b>: 线性输出镜头，并易于进入。</li> <li>- <b>轻松输出</b>: 轻松输出输出，并线性输入输入。</li> <li>- <b>硬入</b>: 容易出局，并且很难入局。</li> <li>- <b>硬出</b>: 硬出，容易进入。</li> <li>- <b>线性</b>: 线性混合。机械好看。</li> <li>- <b>自定义</b>: 自定义混合曲线。</li> </ul>
时间	单	混合持续时间，以秒为单位。

## CinemachineBlenderSettings

定义虚拟相机之间混合规则的资产。

“发件人”和“收件人”设置是基于名称的，这意味着通过将相机的名称与设置进行匹配可以找到相机。他们没有链接到特定的游戏对象。



**Blender支持名为\*\* ANY CAMERA \*\*的通配符**，您可以在混合的输入或输出上使用这种**通配符**，因此无论您使用何种相机，只需要一次混合就可以指定一个混合进入或退出相同的方式。

设置	类型	描述
自定义混合	CustomBlend []	该数组包含两个虚拟摄像机之间明确定义的混合。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

容器指定两台特定的Cinemachine虚拟摄像机如何混合在一起。

设置	类型	描述
从	串	从本相机混合时。
至	串	混合到本相机时。
混合	CinemachineBlendDefinition	混合曲线定义。

## 关注Cams和玩家输入

在手册的前几节中，我们讨论了基本的虚拟相机。它们非常适合电影和构图，但很难在移动目标的路径上设置相机，到目前为止，还没有考虑相机的位置以响应玩家的输入。

因为Cinemachine是一个模块化系统，所以很容易用不同的组件替换Body部分中的Transposer - 考虑Follow目标的速度，以及来自各种HID的玩家输入。**OrbitalTransposer**就是这样一个组件。

基于OrbitalTransposer，**CinemachineFreeLook**相机是一款复杂的装备，可让玩家在两个方向上定位相机：水平使用OrbitalTransposer，垂直使用3个相互关联的儿童装备的超强自定义混合。

玩家输入是通过在Unity输入管理器中设置轴来完成的，或者您可以使用任何自定义或专有输入系统直接驱动这些值。

## CinemachineOrbitalTransposer

这是组件管道正文部分中的CinemachineComponent。它的工作是将摄像机放置在与vcam的跟踪目标对象具有可变关系的位置，并具有偏移和阻尼。

该组件通常用于实现跟随其目标的摄像头。它可以接受来自输入设备的玩家输入，从而允许玩家动态控制摄像头和目标之间的关系，例如使用游戏杆。

OrbitalTransposer引入了**Heading**的概念，这是目标移动的方向，OrbitalTransposer将尝试将摄像头放置在与标题相关的位置上，默认情况下直接位于目标后面。您可以通过调整标题偏差设置来控制默认关系。

如果您将输入控制器连接到OrbitalTransposer，则玩家还可以控制摄像机相对于目标方位的位置。这使相机可以移动到目标周围任何位置。您可以配置OrbitalTransposer从输入管理器中以前设置的任何轴输入其输入，或者可以使用任何自定义输入系统直接控制该值。

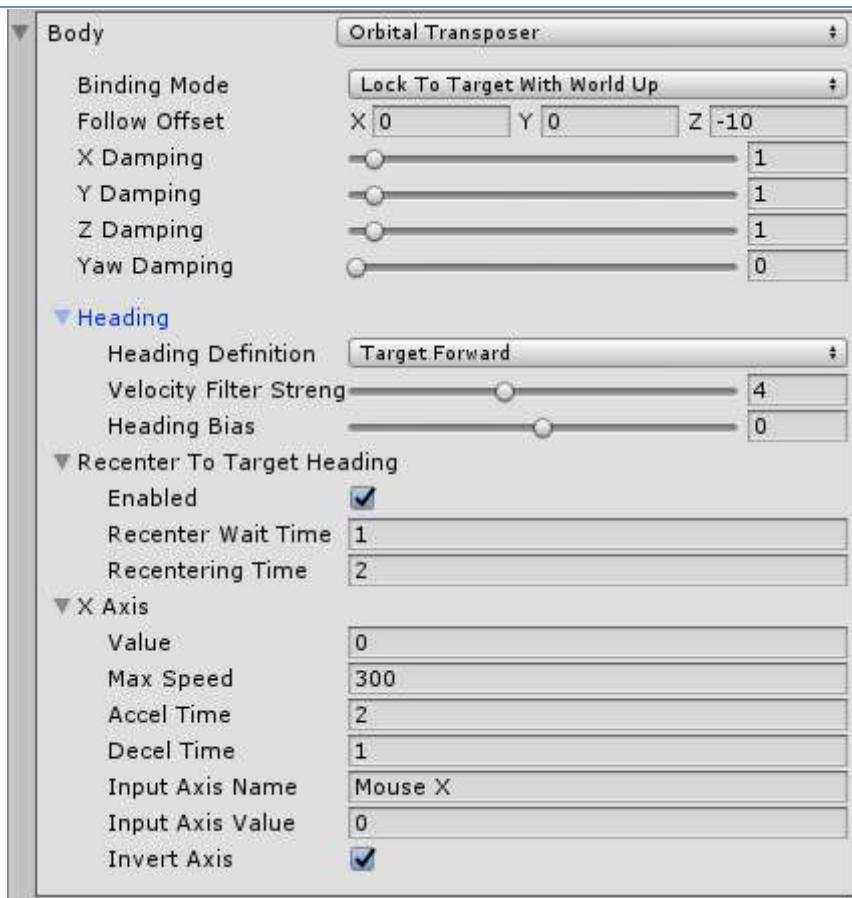
OrbitalTransposer的另一个特点是自动重新搜索。启用后，在指定时间内没有检测到用户输入后，它会自动将相机移回默认目标标题。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼



设置	类型	描述
标题	标题	Forward的定义。相机会跟在后面。
接收者目标标题	Recentering	自动标题重新标记。这里的设置定义了相机如何在没有玩家输入的情况下重新定位自己。
X轴	AxisState	标题控制。这里的设置控制相机响应玩家输入的行为。
绑定模式	BindingMode	<p>解释来自目标的偏移量时使用的坐标空间。这也用于设置相机的向上矢量，这将在瞄准相机时保持不变。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>锁定目标位置指定</b>：使用由虚拟相机启用时的目标本地框架组成的参照系，或者指定目标时，相机将被绑定到跟随目标。</li> <li>- <b>锁定目标与世界上</b>：相机将绑定到跟踪目标使用一个由目标的本地框架组成的参考框架，倾斜和横滚归零。</li> <li>- <b>锁定目标没有卷</b>：使用由目标本地框架组成的参照系将相机绑定到跟随目标，并将滚轮置零。</li> <li>- <b>锁定目标</b>：相机将使用目标的本地框架绑定到跟随目标。</li> <li>- <b>世界空间</b>：使用世界空间偏移将相机绑定到跟随目标。</li> <li>- <b>随着世界的简单追踪</b>：偏移量将使用相机局部轴相对于目标进行计算。</li> </ul>
按照偏移量	的Vector3	转换器将尝试从Follow目标保持的距离向量。
X阻尼	单	相机尝试保持X轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的X轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
Y	单	相机尝试保持Y轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的Y轴偏移。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
Z阻尼	单	摄像机试图保持Z轴的偏移量。小数字响应更快，快速平移相机以保持目标的Z轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
音调阻尼	单	相机试图追踪目标旋转的X角度的积极性。小数字响应更快。数字越大，响应速度越慢。
偏航阻尼	单	相机尝试追踪目标旋转的Y角度的积极性。小数字响应更快。数字越大，响应速度越慢。
滚动阻尼	单	相机试图追踪目标旋转的Z角度的积极性。小数字响应更快。数字越大，响应速度越慢。

## CinemachineOrbitalTransposer.Heading

如何定义“前进”方向。轨道偏移与前进方向有关。

设置	类型	描述
标题定义	HeadingDefinition	<p>如何“向前”被定义。相机将默认放置在目标后面。PositionDelta会认为‘前进’是目标移动的方向。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>位置增量：</b>根据上次更新位置与当前帧之间的差异计算出的目标标题。</li> <li>- <b>速度：</b>从其刚体速度计算出的目标航向。如果不存在刚体，它将回退到 HeadingDerivationMode.PositionDelta。</li> <li>- <b>目标转发：</b>根据目标转换的欧拉Y角计算出的目标航向。</li> <li>- <b>世界转发：</b>默认标题是一个恒定的世界空间标题。</li> </ul>
速度过滤器强度	INT32	目标航向滤波器的速度采样窗口的大小。这可以滤除目标运动中的不规则性。仅在从目标移动中获取标题时才使用 (PositionDelta或Velocity)。
标题偏见	单	当X轴值为零时放置相机的位置。这是围绕Y轴旋转的度数。当此值为0时，相机将被放置在目标后面。非零偏移将旋转目标周围的零位。

## AxisState

轴状态用于定义如何对玩家输入作出反应。此处的设置控制着轴对玩家输入的响应度。

设置	类型	描述
值	单	轴的当前值。
最大速度	单	该轴的最大速度，单位/秒。
加速时间	单	用提供的轴以其最大值加速到MaxSpeed所需的时间量（以秒为单位）。
减速时间	单	如果提供的轴处于空档位置，则使轴减速到零所需的时间量（以秒为单位）。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
输入轴名称	串	在Unity输入管理器中指定的该轴的名称。设置为空字符串将禁用该轴的自动更新。
输入轴值	单	输入轴的值。值为0意味着没有输入。您可以直接从自定义输入系统驱动该轴，或者您可以设置轴名称并使其由内部输入管理器驱动。
反转输入	布尔	如果选中，则输入轴的原始值在使用前会被反转。
最小值	单	轴的最小值。
最大值	单	轴的最大值。
包	布尔	如果选中，则轴将以最小/最大值环绕，形成一个循环。

## AxisState.Recentering

帮助器自动重新坐标轴。

设置	类型	描述
启用	布尔	如果选中，将启用轴的自动重新调整。如果未选中，则近期被禁用。
等待时间	单	如果没有在轴上检测到用户输入，轴将在重新编辑之前等待几秒钟。
接收时间	单	重新调整的最大角速度。将加速进入并减速。

## CinemachineFreeLook

Cinemachine相机适合第三人称相机体验。相机围绕其拍摄对象旋转三个独立的相机装置，定义目标周围的环。每个钻机都有自己的半径，高度偏移，作曲家和镜头设置。根据连接这三个钻台的样条线上相机的位置，可以插入这些设置以给出最终的相机位置和状态。

玩家输入沿着2个轴提供：控制轨道位置的X轴（参见CinemachineOrbitalTransposer）和控制连接3个儿童装备的样条线上的垂直位置的Y轴。

每个儿童装备都可以有自己的镜头设置，或者可以使用共同的共享值。此外，每个儿童装备都可以有自己的Composer和Noise设置。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

**Orbits**

Binding Mode: Simple Follow With World Up

Spline Curvature: 0.2

TopRig	Height: 4.5	Radius: 1.75
MiddleRig	Height: 2.5	Radius: 3
BottomRig	Height: 0.4	Radius: 1.3

**TopRig**

Look At Override: None (Transform)

- Body: Orbital Transposer
- Aim: Composer
- Noise: none

**MiddleRig**

Look At Override: None (Transform)

- Body: Orbital Transposer
- Aim: Composer
- Noise: none

**BottomRig**

Look At Override: None (Transform)

- Body: Orbital Transposer
- Aim: Composer
- Noise: none

设置	类型	描述
看着	转变	对象为照相机孩子看（目标目标）。
跟随	转变	儿童想要随身携带的物体（身体目标）。
<b>共同的镜头</b>	布尔	如果启用，此镜头设置将适用于所有三个儿童装备，否则将使用儿童装备镜头设置。
镜片	镜头设置	指定此虚拟摄像机的镜头属性。这通常反映了Unity摄像机的镜头设置，并将在vcam处于活动状态时用于驱动Unity摄像头。
和轴	AxisState	垂直轴。值是0..1。选择如何混合儿童装备。
X轴	AxisState	水平轴。值是0..359。这被传递到钻塔的OrbitalTransposer组件。
标题	标题	Forward的定义。相机会跟在后面。
<b>接收者目标标题</b>	Recentering	控制如何自动重新X轴的完成。



已翻译为以下语言： 中文(简体)

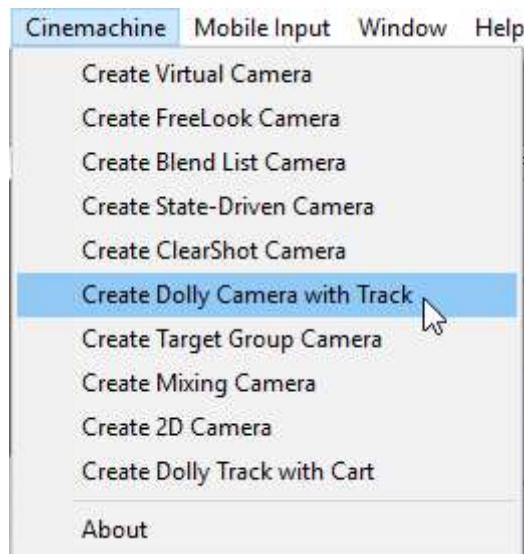
显示原文

选项 ▼

设置	类型	描述
绑定模式	BindingMode	<p>解释来自目标的偏移量时使用的坐标空间。这也用于设置相机的向上矢量，这将在瞄准相机时保持不变。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>锁定目标位置指定</b>: 使用由虚拟相机启用时的目标本地框架组成的参照系，或者指定目标时，相机将被绑定到跟随目标。</li> <li>- <b>锁定目标与世界上</b>: 相机将绑定到跟踪目标使用一个由目标的本地框架组成的参考框架，倾斜和横滚归零。</li> <li>- <b>锁定目标没有卷</b>: 使用由目标本地框架组成的参照系将相机绑定到跟随目标，并将滚轮置零。</li> <li>- <b>锁定目标</b>: 相机将使用目标的本地框架绑定到跟随目标。</li> <li>- <b>世界空间</b>: 使用世界空间偏移将相机绑定到跟随目标。</li> <li>- <b>随着世界的简单追踪</b>: 偏移量将使用相机局部轴相对于目标进行计算。</li> </ul>
样条曲率	单	控制连接钻机轨道的线条，这决定了Y轴上的最终位置。
轨道	轨道[]	三个轨道钻机的半径和高度。
优先	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## 跟踪的多莉

如果您需要将相机的位置限制在小车轨道上，则可以使用轨道的CinemachinePath和虚拟相机的车身组件的TrackedDolly组合完成。在Cinemachine菜单中创建这个对有一个快捷方式：



路径本身是通过定位航点来定义的。CinemachinePath检查器为此提供了一个简单的界面。航点用bezier连接，您可以通过操纵航点处的曲线切线来控制插值的方式。这样做的结果是一个摄像机轨道，以类似于铁轨的方式在场景视图中绘制。它不会在游戏视图中绘制。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼



## CinemachineTrackedDolly

Cinemachine虚拟相机机身组件，将相机运动限制在CinemachinePath或CinemachineSmoothPath上。相机可以沿着路径移动。

该组件可以在两种模式下操作：手动定位和自动多边形定位。在手动模式下，相机的位置通过动画设置路径位置字段来指定。在自动移动摄影模式下，通过查找最接近虚拟摄像机跟随目标的路径上的位置，每个帧都会自动为路径位置字段设置动画。

设置	类型	描述
路径	CinemachinePathBase	相机将被约束的路径。这必须是非空的。
路径位置	单	照相机将放置在路径上的位置。这可以直接进行动画制作，也可以通过Auto-Dolly功能自动设置，以尽可能接近Follow目标。该值根据位置单位设置进行解释。
位置单位	位置单位	<p>如何解释路径位置。如果设置为路径单位，则值如下所示：0代表路径上的第一个航点，1代表第二个航点，依此类推。中间值是航点之间的路径上的点。如果设置为距离，则路径位置表示沿路径的距离。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>路径单位</b>：使用PathPosition单位，其中0是第一个航点，1是第二个航点等。</li> <li>- <b>距离</b>：使用沿路径的距离。路径将根据其分辨率设置进行采样，并且距离查找表将在内部高速缓存。</li> <li>- <b>标准化</b>：标准化单位，其中0是路径的开始，1是结束。路径将根据其分辨率设置进行采样，并且距离查找表将在内部高速缓存。</li> </ul>



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
路径偏移	的Vector3	相机相对于路径位置的位置。X垂直于路径，Y向上，Z平行于路径。这样相机就可以从路径本身偏移（例如，就像在三脚架上一样）。
X阻尼	单	相机尝试在垂直于路径的方向上保持其位置的积极性。小数字响应更快，快速平移相机以保持目标的x轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
Y Damping	单	摄像机试图保持其在路径局部向上方向上的位置。小数字响应更快，快速平移相机以保持目标的y轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
Z阻尼	单	摄像机试图保持其在平行于路径的方向上的位置有多积极。小数字响应更快，快速平移相机以保持目标的Z轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
相机了	CameraUpMode	<p>如何设置虚拟相机的向上向量。这将影响屏幕组成，因为相机Aim行为将始终尝试尊重Up方向。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>默认值</b>：保持摄像机的向上矢量。它将根据Brain's WorldUp进行设置。</li> <li>- <b>路径</b>：从当前点的路径向上向量中取出向上向量。</li> <li>- <b>路径否滚动</b>：从当前点的路径向上矢量中取出向上矢量，但滚动为零。</li> <li>- <b>追踪目标</b>：从追踪目标的向上向量中取出向上向量。</li> <li>- <b>追踪目标没有滚动</b>：从跟随目标的向上向量中取出向上矢量，但卷的零点已经清零。</li> </ul>
音调阻尼	单	相机试图追踪目标旋转的X角度的积极性。小数字响应更快。数字越大，响应速度越慢。
偏航阻尼	单	相机尝试追踪目标旋转的Y角度的积极性。小数字响应更快。数字越大，响应速度越慢。
滚动阻尼	单	相机试图追踪目标旋转的Z角度的积极性。小数字响应更快。数字越大，响应速度越慢。
汽车小车	AutoDolly	控制如何自动推动发生。追踪目标对于使用此功能是必需的。

## CinemachineTrackedDolly.AutoDolly

控制如何自动推动发生。

设置	类型	描述
启用	布尔	如果选中，将启用自动小车，该小车选择尽可能接近跟随目标的路径位置。注意：这可能会对性能产生重大影响。
位置偏移	单	以当前位置单位从路径上的最近点偏移到跟随目标。
搜索半径	INT32	搜索当前位置两侧有多少个航点。整个路径使用0。
搜索分	INT32	我们通过将航段分成许多直路段来在航点之间进行搜索。数字越高，结果越准确，但对更



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

## CinemachinePath

定义一个世界空间路径，由一组航点组成，每个航点都有位置，切线和滚转设置。Bezier内插在航点之间执行，以获得平滑和连续的路径。

设置	类型	描述
环	布尔	如果选中，则路径结尾会连接起来形成一个连续的循环。
航点	航点[]	定义路径的路标。它们将使用贝塞尔曲线进行插值。
解析度	INT32	每个航点的路径样本。这用于计算路径距离。
出现	出现	控制路径将如何显示在编辑器场景视图中的设置。

## CinemachinePath.Waypoint

沿着路径的一个航点。

设置	类型	描述
位 置	的 Vector3	在路径局部空间中的位置。
切 线	的 Vector3	从位置偏移，该位置定义了航点处曲线的切线。切线的长度编码贝塞尔手柄的强度。在航点的两侧对称使用相同的手柄，以确保平滑。
滚	单	定义此航点上的路径滚动。其他定向轴由切线和世界向上推断。

## CinemachineSmoothPath

定义一个世界空间路径，由一系列航点组成，每个航点都有位置和滚转设置。Bezier内插在航点之间执行，以获得平滑和连续的路径。路径将通过所有路点，并且（与CinemachinePath不同）保证一阶和二阶连续性。

设置	类型	描述
环	布尔	如果选中，则路径结尾会连接起来形成一个连续的循环。
航点	航点[]	定义路径的路标。它们将使用贝塞尔曲线进行插值。
解析度	INT32	每个航点的路径样本。这用于计算路径距离。
出现	出现	控制路径将如何显示在编辑器场景视图中的设置。

## CinemachineSmoothPath.Waypoint

沿着路径的一个航点。

设置	类型	描述
位置	的Vector3	在路径局部空间中的位置。
滚	单	定义此航点上的路径滚动。其他定向轴由切线和世界向上推断。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

该类保存控制路径在编辑器场景视图中的显示方式的设置。该路径在游戏视图中不可见。

设置	类型	描述
路径颜色	颜色	路径本身在编辑器中处于活动状态时的颜色。
非活动路径颜色	颜色	路径本身在编辑器中处于非活动状态时的颜色。
宽度	单	绘制的轨道的宽度代表路径。

## 国家驱动的相机

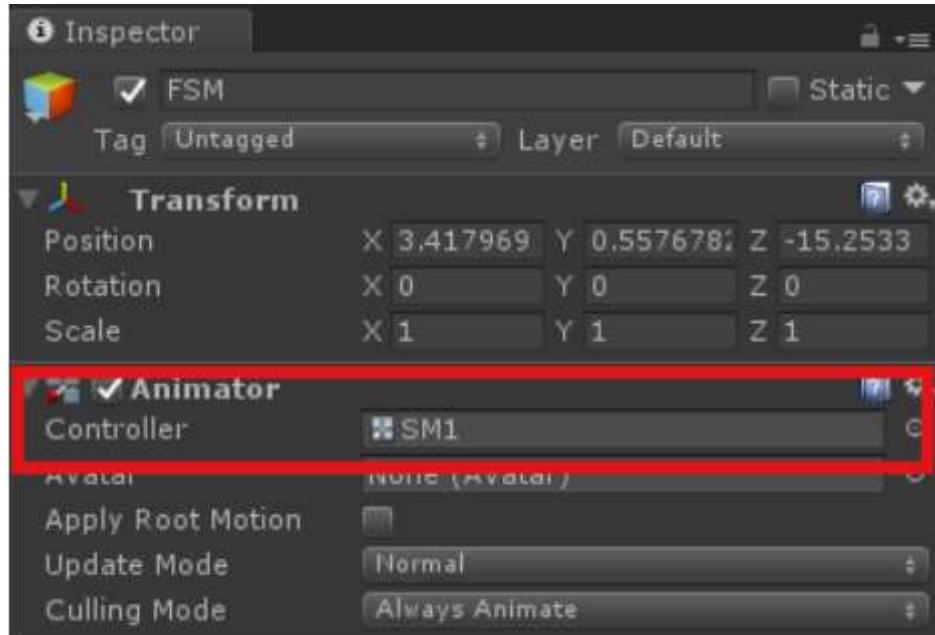
状态驱动相机是一个简单而强大的概念的实现：当主体进入特定的动画状态时激活特定的镜头。

更好的是，状态驱动平台内镜头之间的混合可以进行定制和调整，以最大限度地发挥过渡的情感冲击。

使用Cinemachine可以建立世界级的第三人称动作冒险相机系统。例如，您可以为这样的事件创建一个独特的自由观看相机：Stand / Walk, Run, Sprint等。将相应的Cinemachine自由观看相机链接到该动画状态，以便在动画状态触发时转动在那台Cinemachine相机上。

这个想法是简单地将每个动画状态混合到正确的自由摄像机中。您的'Sprint'自由曲线靠得更近，镜头更宽，并且存在大量的手持噪音。繁荣，你现在有一个'Roadie Run'相机。这个概念已被用于建立一个复杂的相机设备，拥有超过40个自由视角相机，用于角色的局部运动和战斗系统，所有这些都设置得如此之快并且可以实时调整。

您首先从一个动画目标开始，即某个动画组件由FSM控制：



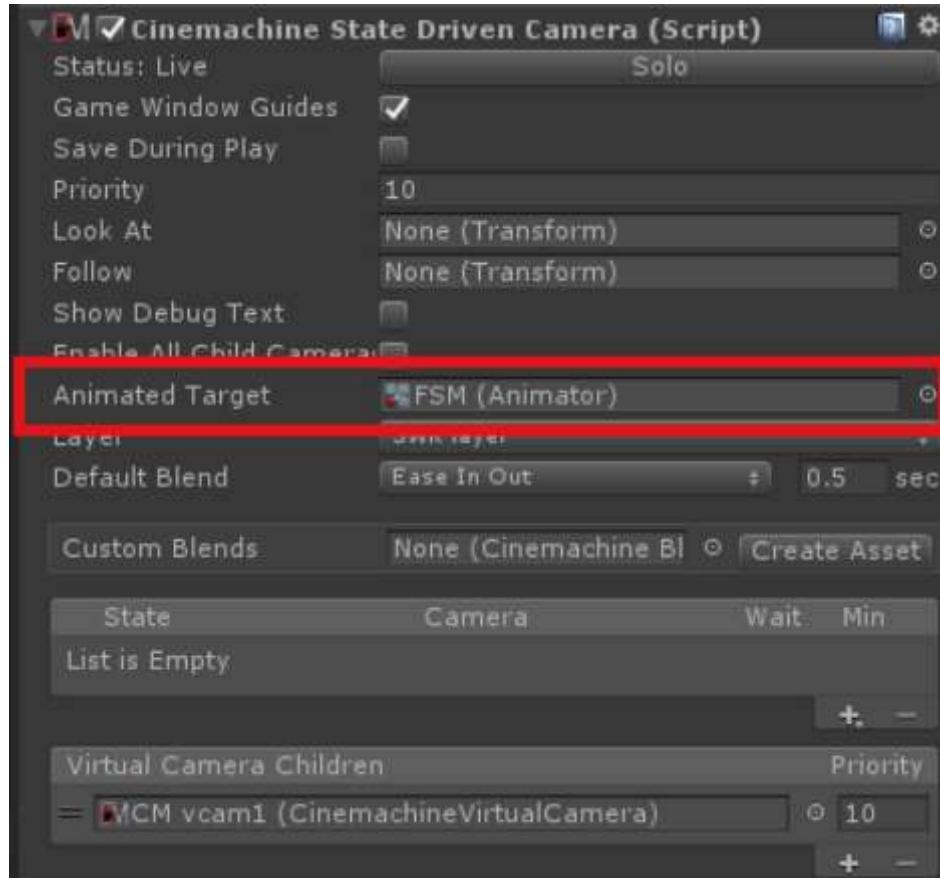


已翻译为以下语言: 中文(简体)

显示原文

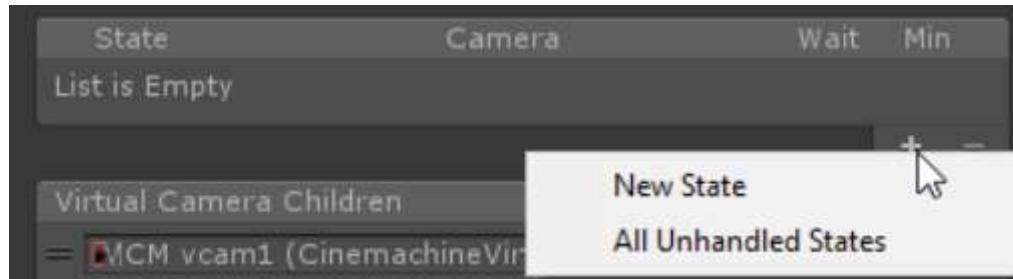
选项 ▼

然后，您创建一个CinemachineStateDrivenCamera并将FSM连接到其动画目标：

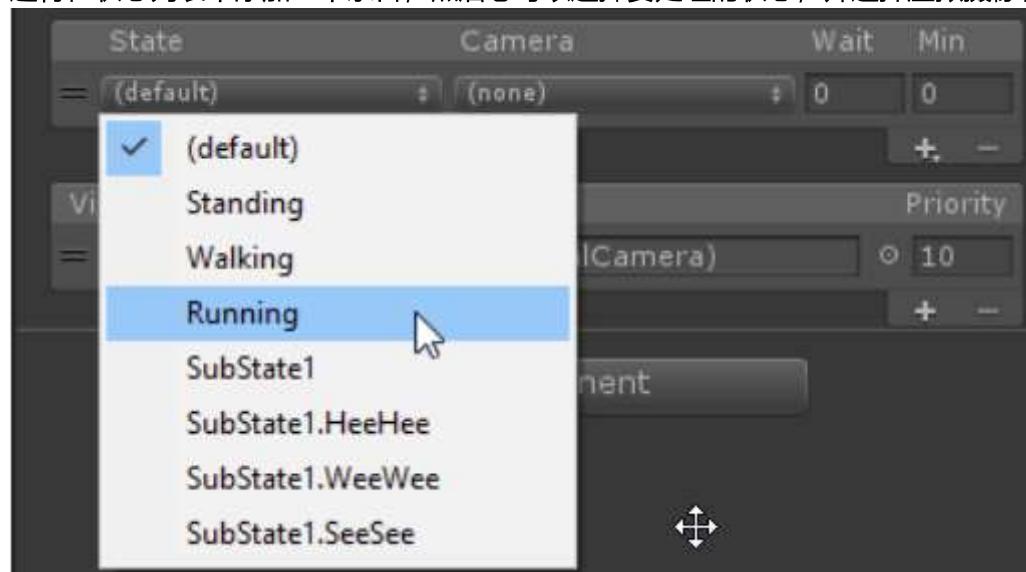


将“动画目标”设置指向具有动画状态的目标对象后，检测器编辑器将扫描目标对象以获取动画状态并填充其菜单，以便快速轻松地将可用动画状态映射到子vcams。

接下来，创建一些状态到镜头的映射。你可以通过点击状态列表中的小图标来实现：



这将在状态列表中添加一个条目，然后您可以选择要处理的状态，并选择虚拟摄像机子图将其映射到。



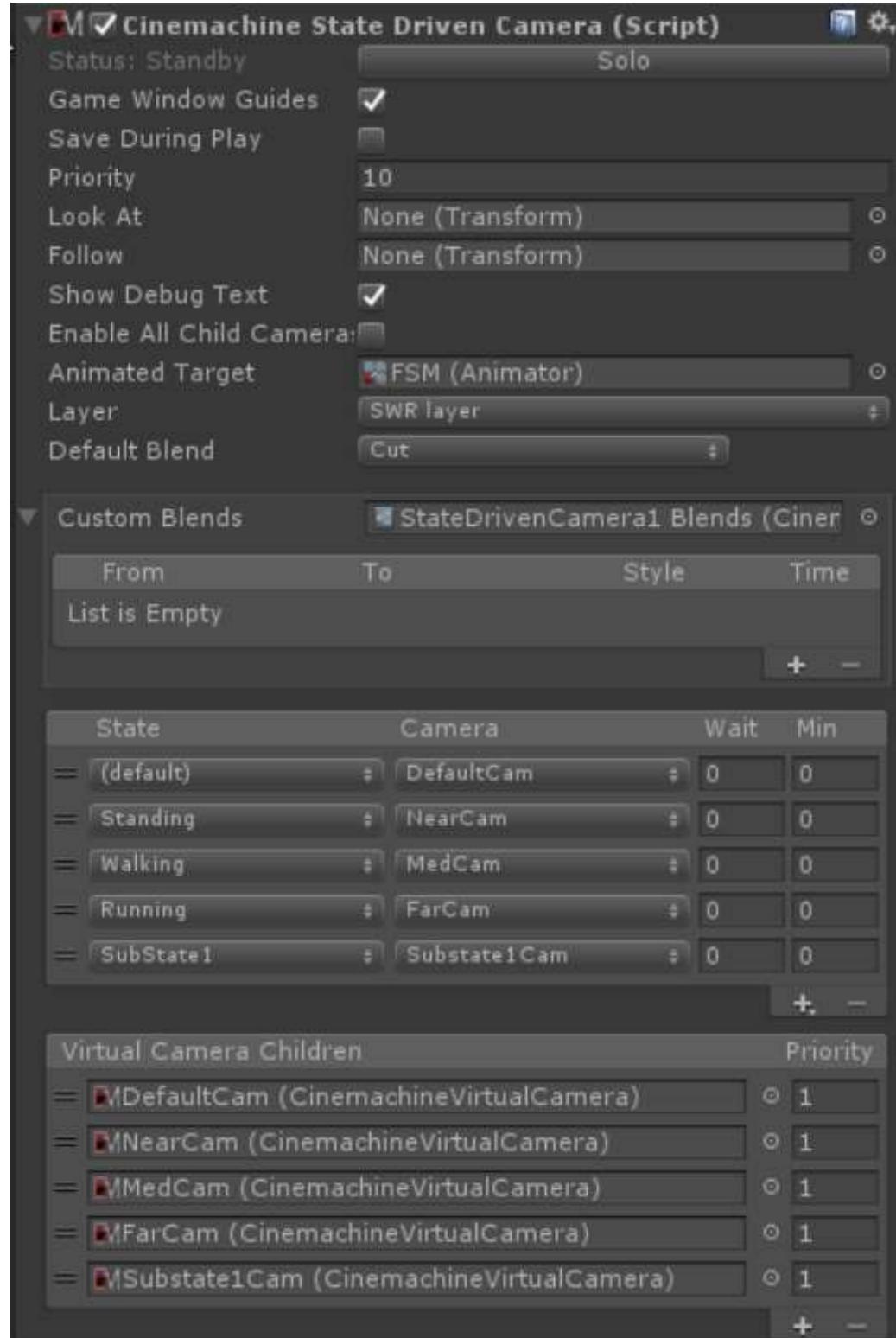


已翻译为以下语言： 中文(简体)

显示原文

选项 ▾

可用的摄像机是StateDrivenCamera的虚拟摄像机子。根据你的喜好创建你需要的品种。如果您喜欢，您可以使用层次结构视图将外部创建的vcams重新映射到StateDrivenCamera，或者只需按相机列表上的小节点以创建标准视频。当你这样做时，这些相机将可用于映射。这些虚拟摄像头只对状态驱动摄像头父级可见，并且不会消耗任何处理资源，直到它们被父级激活。



你最终会得到这样的结果：

所以现在，当列表中的状态变为活动状态时，相关的子摄像机也将变为活动状态。他们将使用StateDrivenCamera中定义的默认混合设置进行混合。您还可以通过在StateDrivenCamera上创建自定义混合资源来为特定对创建自定义混合。

## CinemachineStateDrivenCamera



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

您可以定义儿童相机之间的自定义混合和转换。

为了使用这种行为，你必须有一个动画目标（即用状态机动画的对象）来驱动行为。

设置	类型	描述
看着	转变	如果没有在儿童相机中指定，则相机儿童可以查看的默认对象（目标目标）。如果所有的孩子都定义他们自己的目标，则可能是空的。
跟随	转变	如果未在子摄像机中指定，则相机儿童的默认对象需要与（身体目标）一起移动。如果所有的孩子都定义他们自己的目标，则可能是空的。
动画目标	动画制作者	状态改变的状态机将驱动这台摄像机选择活动孩子。
层索引	INT32	在目标状态机中观察哪一层。
显示调试文本	布尔	启用后，当前的儿童相机和混合将在游戏窗口中显示，供调试。
启用所有儿童相机	布尔	强制启用所有子摄像头。如果在时间轴中对它们进行动画处理，这会非常有用，但会消耗额外的资源。
说明	指令[]	将虚拟相机与状态相关联的指令集。这些说明用于在任何特定时刻选择活着的孩子。
默认混合	CinemachineBlendDefinition	如果您未明确定义两个虚拟摄像机子项之间的混合，则使用该混合。
自定义混合	CinemachineBlenderSettings	这是包含特定子混合的自定义设置的资产。
优先	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## 防撞和射击评估

当角色在复杂的环境中移动时，场景中的障碍物有时会出现在相机与其主体之间。同样，也可能发生这样的情况，即障碍物会不方便地发现自己恰好处于相机想要的位置。Cinemachine提供了处理这些情况的机制。

Collider模块可以添加到任何Cinemachine虚拟摄像机（包括复杂的摄像机类型，如FreeLook，State-Driven-Camera和ClearShot）。它可以完成以下几项或全部内容：

- 将相机推出场景中相交的障碍物。
- 将相机放置在摄像头与其LookAt目标之间的障碍物前方。
- 评估**拍摄质量**（相机是否已经远离其想要的位置？相机是否清晰地看到了它的目标？相机距目标的距离是否合适？）。此镜头评估成为vcam状态信息的一部分，可供基于镜头质量执行操作的模块使用（例如，请参阅ClearShot）。
- 相机位置的高斯平滑。

对撞机使用物理Raycasts来做这些事情，因此场景中的障碍物必须有对撞体积才能被CinemachineCollider看到。此外，还有与此相关的相关性能成本。如果你的游戏使用物理的成本太高，那么你可能更喜欢以不同的方式实现这个功能。

## CinemachineCollider



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

Cinemachine虚拟摄像机的附加模块，用于后期处理虚拟摄像机的最终位置。根据提供的设置，对撞机将尝试通过远离妨碍视图的物体保留虚拟摄像机的LookAt目标的视线。

此外，对撞机可用于评估拍摄质量，并将其报告为相机状态中的字段。

设置	类型	描述
<b>碰撞</b>	LayerMask	Collision将对其进行光线投射的Unity层掩码。
<b>距目标的最短距离</b>	单	靠近目标的障碍物将不会被看到。
<b>避免障碍</b>	布尔	启用后，将尝试解决目标视线被障碍物阻挡的情况。
<b>距离限制</b>	单	检查相机目标视线是否清晰时的最大光线投射距离。如果设置为0或更小，则将使用到目标的当前实际距离。
<b>相机半径</b>	单	相机会尽量保持与任何障碍物的距离。尽量保持这个值小。由于摄像头上的FOV较大，如果您看到内部障碍物，请将其增大。
<b>战略</b>	ResolutionStrategy	<p>对撞机将试图保持对目标的看法。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>向前拉摄像机</b>：摄像机将沿其Z轴向前拉，直到其位于最近的障碍物的前方。</li> <li>- <b>保留相机高度</b>：除了向前拉相机之外，还会努力将相机恢复到原始高度。</li> <li>- <b>保留摄像机距离</b>：除了向前拉摄像机外，还需要努力将摄像机恢复到距目标的原始距离。</li> </ul>
<b>最大的努力</b>	INT32	上限处理多少障碍命中。数字越高可能会影响性能。在大多数环境中，4就足够了。
<b>减震</b>	单	碰撞解决的渐进性。较高的数字会使摄像机逐渐远离障碍物。
<b>最佳目标距离</b>	单	如果大于零，则当目标更接近此距离时，将给予更高分数。将其设置为零以禁用此功能。

## CinemachineConfiner

Cinemachine虚拟摄像机的附加模块，用于后期处理虚拟摄像机的最终位置。它会将虚拟相机的位置限制在“边界体积”字段中指定的体积。

这比CinemachineCollider资源密集程度要低，但它不会执行镜头评估。

confiner可以在2D或3D模式下运行。该模式将影响它将接受的边界形状。在3D模式下，相机在3D中的位置将局限于音量。这将适用于2D游戏，但您必须考虑深度。在2D模式下，您不必担心深度。

如果您的相机是拼写正字，则会有一个额外的选项来限制屏幕边缘，而不仅仅是相机点。这确保整个屏幕区域保持在边界形状内。

设置	类型	描述
<b>限制模式</b>	模式	<p>confiner可以使用2D边界形状或3D边界体积进行操作。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>限制2D</b>：需要Collider2D边界形状</li> <li>- <b>限制3D</b>：需要3D Collider绑定体积</li> </ul>



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
包围形状 2D	Collider2D	相机所在的2D形状。
限制屏幕边缘	布尔	如果相机是正交图像，屏幕边缘将被限制在音量范围内。如果没有选中，那么只有相机中心被限制。
减震	单	如果摄像机超出边界，则逐渐将摄像机返回到边界体积。更高的数字更渐进。

## CinemachineClearShot

Cinemachine ClearShot是一个“经理相机”，拥有和管理一系列虚拟相机gameObject的孩子。当Live时，ClearShot会检查孩子，并选择质量最好的镜头并使其生效。

这可能是一个非常强大的工具。如果儿童摄像机具有CinemachineCollider扩展功能，他们将分析场景中的目标障碍物，最佳目标距离和其他物品，并将其拍摄质量评估报告给ClearShot父母，然后他们将选择最佳的一个。您可以使用它来设置复杂的多摄像头覆盖场景，并确保始终可以清晰地拍摄目标。

如果多个子摄像机具有相同的镜头质量，则将选择具有最高优先级的摄像机。

您还可以定义ClearShot子项之间的自定义混合。

设置	类型	描述
看着	转变	如果没有在儿童相机中指定，则相机儿童可以查看的默认对象（目标目标）。如果所有孩子都指定他们自己的目标，则可能是空的。
跟随	转变	如果未在子摄像机中指定，则相机儿童的默认对象需要与（身体目标）一起移动。如果所有孩子都指定他们自己的目标，则可能是空的。
显示调试文本	布尔	启用后，当前的儿童相机和混合将在游戏窗口中显示，供调试。
激活后	单	在启动新的儿童相机之前等待几秒钟。
最短时间	单	主摄像头必须至少在这几秒钟内激活，除非较高优先级的摄像头需要激活。
随机选择	布尔	如果选中，如果同时需要多台摄像机，则会随机选择相机。否则，将使用子列表顺序和子摄像头优先级。
默认混合	CinemachineBlendDefinition	如果您未明确定义两台虚拟相机之间的混合，则使用该混合。
优先	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## 多个目标对象

如果您有多个LookAt目标，并且想要将它们全部放在框架中，Cinemachine也可以这样做。这是一个两步骤的过程：

1. 定义一个目标组。这是一个目标对象列表，每个目标对象都有一个权重和半径。重量说明团队成员的重要性，半径是其物理尺寸的粗略指示。

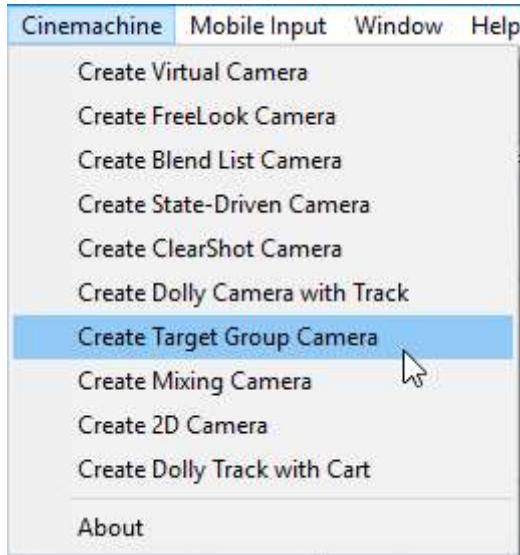


已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

Cinemachine菜单中有一个快捷方式来设置它并让你走。



## CinemachineTargetGroup

定义一组目标对象，每个对象都有一个半径和一个权重。计算目标组的平均位置时使用权重。该组的高级成员将计数更多。边界框是通过考虑成员位置，重量和半径来计算的。

设置	类型	描述
位置模式	PositionMode	<p>该团队的职位如何计算。选择GroupCenter作为边界框的中心，GroupAverage选择成员位置的加权平均值。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>组中心：</b>组位置将是该组的轴对齐边界框的中心。</li> <li>- <b>集团平均值：</b>集团头寸将是成员头寸的加权平均值。</li> </ul>
旋转模式	RotationMode	<p>如何计算小组的轮换。选择手动以使用组转换中的值，并使用组平均作为成员方向的加权平均值。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>手动：</b>在组的变换中手动设置。</li> <li>- <b>组平均值：</b>其成员方向的加权平均值。</li> </ul>
更新方法	UpdateMethod	<p>何时根据组成员的位置更新组的变换。</p> <p><b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>更新：</b>在正常的MonoBehaviour更新中更新。</li> <li>- <b>更新更新：</b>与固定更新中的物理模块同步更新。</li> <li>- <b>最新更新：</b>在MonoBehaviour LateUpdate中更新。</li> </ul>
目标	目标[]	目标对象以及它们的权重和半径将有助于组的平均位置，方向和大小。

## CinemachineTargetGroup.Target

持有代表该组成员的信息。

设置	类型	描述
目标	转变	目标对象。该物体的位置和方向将根据其重量对该组的平均位置和方向作出贡献。
重量	单	平均时赋予目标多少权重。不能是负面的。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

# CinemachineGroupComposer

这是组件管道目标部分中的CinemachineComponent。它的工作是将摄像机瞄准目标物体，并具有可配置的偏移，阻尼和合成规则。

另外，如果目标是CinemachineTargetGroup，则该行为将调整FOV和相机距离以确保整个目标组被正确构筑。

设置	类型	描述
<b>组框尺寸</b>	单	目标的边界框应该占据这个量的屏幕空间。1表示填满整个屏幕。0.5表示填满屏幕的一半等
<b>取景模式</b>	FramingMode	框架时需要考虑什么屏幕尺寸。可以是水平，垂直或两者。 <b>可能的值：</b> <ul style="list-style-type: none"><li>- <b>水平</b>: 仅考虑水平尺寸。垂直框架被忽略。</li><li>- <b>垂直</b>: 只考虑垂直尺寸。水平取景被忽略。</li><li>- <b>水平和垂直</b>: 水平和垂直尺寸中的较大者将占主导地位，以获得最佳配合。</li></ul>
<b>框架阻尼</b>	单	相机试图构建组合的方式非常积极。小数字响应更快，可以快速调整摄像头以将组保留在画面中。数字越大，响应速度越慢。
<b>调整模式</b>	AdjustmentMode	如何调整相机以获得所需的成帧。你可以放大，放大/缩小，或者两者兼而有之。 <b>可能的值：</b> <ul style="list-style-type: none"><li>- <b>仅缩放</b>: 不要移动相机，只调整FOV。</li><li>- <b>Dolly Only</b>: 只需移动相机，不要更改FOV。</li><li>- <b>多莉然后变焦</b>: 尽可能多地移动相机，然后根据需要调整FOV进行拍摄。</li></ul>
<b>Max Dolly In</b>	单	朝向目标的最大距离，此行为被允许移动相机。
<b>Max Dolly Out</b>	单	该行为被允许移动相机的目标的最大距离。
<b>最小距离</b>	单	设置此限制摄像机可以达到的目标距离。
<b>最大距离</b>	单	设置此项可限制摄像机可以达到的目标距离。
<b>最小FOV</b>	单	如果调整FOV，则不会将FOV设置为低于此值。
<b>最大FOV</b>	单	如果调整FOV，则不会将FOV设置得高于此值。
<b>最小Ortho 尺寸</b>	单	如果调整字体大小，不会将其设置为低于此值。
<b>最大Ortho 尺寸</b>	单	如果调整字体大小，不会将其设置为高于此值。
<b>跟踪对象偏移</b>	的Vector3	目标在目标局部空间中偏离目标对象的中心。当期望的区域不是被跟踪物体的中心时，使用它来微调跟踪目标位置。
<b>前瞻时间</b>	单	此设置将指示作曲家根据目标的运动调整目标偏移量。作曲家将着眼于估计目标将在未来数秒内完成的一点。请注意，此设置对有噪音的动画很敏感，并且可能会放大噪点，导致不合需要的相机抖动。如果摄像机在目标



已翻译为以下语言： 中文(简体)

显示原文

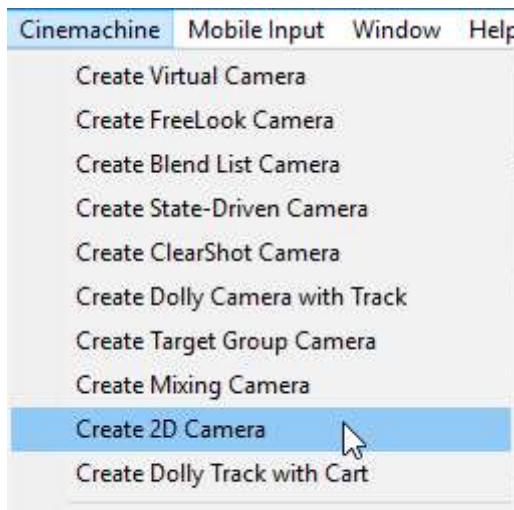
选项 ▼

设置	类型	描述
<b>前瞻平滑</b>	单	控制先行算法的平滑度。较大的值可以消除抖动预测并增加预测滞后时间。
<b>Lookahead 忽略Y.</b>	布尔	如果选中，则预览计算将忽略沿Y轴的移动。
<b>水平阻尼</b>	单	相机尝试在屏幕水平方向上追踪目标的方式非常积极。小数字响应更快，快速定位相机以将目标保持在死区。数字越大，响应速度越慢。使用不同的垂直和水平设置可以产生各种相机行为。
<b>垂直阻尼</b>	单	相机尝试在屏幕垂直方向上跟随目标的方式有多强烈。小数字响应更快，快速定位相机以将目标保持在死区。数字越大，响应速度越慢。使用不同的垂直和水平设置可以产生各种相机行为。
<b>屏幕X</b>	单	目标的水平屏幕位置。相机将旋转以在此定位追踪对象。
<b>屏幕Y</b>	单	目标的垂直屏幕位置，相机将旋转以在此定位追踪对象。
<b>死区宽度</b>	单	如果目标位于此位置范围内，相机将不会水平旋转。
<b>死区高度</b>	单	如果目标位于此位置范围内，相机将不会垂直旋转。
<b>软区宽度</b>	单	当目标位于此区域内时，摄像机将逐渐水平旋转以根据阻尼速度重新对准所需的位置。
<b>软区高度</b>	单	当目标位于该区域内时，摄像机将逐渐垂直旋转，以根据阻尼速度重新对准所需的位置。
<b>偏见X</b>	单	非零偏差将使目标位置水平移动离开软区域的中心。
<b>偏见Y.</b>	单	非零偏置将使目标位置垂直移动离开软区中心。

## 2D环境

Cinemachine支持Orthographic相机。当您将Unity相机的样式设置为Orthographic时，Cinemachine会进行调整以适应它。透镜设置中的FOV将被替换为正交大小。请注意，如果摄像机是正交图像，与FOV相关的设置和FollowZoom等特定于FOV的行为不会产生任何影响。

在拼写正确的环境中，旋转相机通常没有意义。因此，Cinemachine有一个特殊的转换器，可以在不旋转相机的情况下处理取景和构图。





已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

# CinemachineFramingTransposer

这是组件管道正文部分中的Cinemachine组件。它的工作是将相机放置在与vcam的跟踪目标物体固定的屏幕空间关系中，并具有偏移和阻尼。

照相机将首先沿照相机Z轴移动，直到跟随目标位于距照相机XY平面所需的距离处。然后摄像机将在其XY平面内移动，直到跟随目标位于相机屏幕上的所需点。

FramingTransposer只会改变相机在太空中的位置。它不会重新定位或以其他方式瞄准相机。

为了使此组件正常工作，vcam的LookAt目标必须为空。Follow目标将定义相机正在看什么。

如果Follow目标是CinemachineTargetGroup，则可以使用其他控件来动态调整摄像头的视图以构建整个组。

虽然这个组件是专为正交相机而设计的，但它与具有相机功能的相机同样适用，并且可以在3D环境中使用。

设置	类型	描述
<b>前瞻时间</b>	单	此设置将指示作曲家根据目标的运动调整目标偏移量。作曲家将着眼于估计目标将在未来数秒内完成的一点。请注意，此设置对有噪音的动画很敏感，并且可能会放大噪点，导致不合需要的相机抖动。如果摄像机在目标运动时抖动不合格，请关闭此设置，或者更平滑地为目标设置动画。
<b>前瞻平滑</b>	单	控制先行算法的平滑度。较大的值可以消除抖动预测并增加预测滞后时间。
<b>Lookahead</b>	布尔	如果选中，则预览计算将忽略沿Y轴的移动。
<b>忽略Y.</b>		
<b>X阻尼</b>	单	相机尝试保持X轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的x轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
<b>Y Damping</b>	单	相机尝试保持Y轴偏移的积极性。小数字响应更快，快速平移相机以保持目标的y轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
<b>Z阻尼</b>	单	摄像机试图保持Z轴的偏移量。小数字响应更快，快速平移相机以保持目标的Z轴偏移。数字越大，响应速度越慢。每个轴使用不同的设置可以产生各种相机行为。
<b>屏幕X</b>	单	目标的水平屏幕位置。相机将移动到这里定位被跟踪的物体。
<b>屏幕Y</b>	单	目标的垂直屏幕位置，相机将移动到此处定位被跟踪对象。
<b>相机距离</b>	单	摄像机轴线上的距离将根据跟踪目标进行维护。
<b>死区宽度</b>	单	如果目标位于此位置范围内，相机将不会水平移动。
<b>死区高度</b>	单	如果目标位于此位置范围内，相机将不会垂直移动。
<b>死区深度</b>	单	如果跟随目标在指定的相机距离的这个距离内，相机将不会沿着其z轴移动。
<b>无限软区</b>	布尔	如果选中，则软区的大小将不受限制。
<b>软区宽度</b>	单	当目标位于该区域内时，摄像机将逐渐水平移动以根据阻尼速度重新对准所需的位置。
<b>软区高度</b>	单	当目标位于该区域内时，摄像机将逐渐垂直移动以根据阻尼速度重新对准所需的位置。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

设置	类型	描述
偏见X	单	非零偏差将使目标位置水平移动离开软区域的中心。
偏见Y.	单	非零偏置将使目标位置垂直移动离开软区中心。
组合模式	FramingMode	框架时需要考虑什么屏幕尺寸。可以是水平，垂直或两者。 <b>可能的值：</b> <ul style="list-style-type: none"><li>- <b>水平</b>: 仅考虑水平尺寸。垂直框架被忽略。</li><li>- <b>垂直</b>: 只考虑垂直尺寸。水平取景被忽略。</li><li>- <b>水平和垂直</b>: 水平和垂直尺寸中的较大者将占主导地位，以获得最佳配合。</li><li>- <b>无</b>: 不要做任何取景调整。</li></ul>
调整模式	AdjustmentMode	如何调整相机以获得所需的成帧。你可以放大，放大/缩小，或者两者兼而有之。 <b>可能的值：</b> <ul style="list-style-type: none"><li>- <b>仅缩放</b>: 不要移动相机，只调整FOV。</li><li>- <b>Dolly Only</b>: 只需移动相机，不要更改FOV。</li><li>- <b>多莉然后变焦</b>: 尽可能多地移动相机，然后根据需要调整FOV进行拍摄。</li></ul>
组框尺寸	单	目标的边界框应该占据这个量的屏幕空间。1表示填满整个屏幕。0.5表示填满屏幕的一半等
Max Dolly In	单	朝向目标的最大距离，此行为被允许移动相机。
Max Dolly Out	单	该行为被允许移动相机的目标的最大距离。
最小距离	单	设置此限制摄像机可以达到的目标距离。
最大距离	单	设置此项可限制摄像机可以达到的目标距离。
最小FOV	单	如果调整FOV，则不会将FOV设置为低于此值。
最大FOV	单	如果调整FOV，则不会将FOV设置得高于此值。
最小Ortho 尺寸	单	如果调整字体大小，不会将其设置为低于此值。
最大Ortho 尺寸	单	如果调整字体大小，不会将其设置为高于此值。

## 后处理堆栈V1

如果你的Unity后期处理堆栈是你项目中的一个资产（如果你不这样做，你应该这样做，因为这是一个惊人的工具，可以提供世界级的结果），你可以轻松地为任何镜头指定自定义后期效果在你的场景中，即使它是一个虚拟相机，而不是一个真实的相机。

本节介绍后处理堆栈V1的使用。如果您有Post Processing V2，请参阅下面的部分。

只需为您的vcam定义一个后期处理配置文件，然后通过将CinemachinePostFX组件连接到您的vcam来连接它。要完成连接，您还需要Unity摄像头上的CinemachinePostFX组件，以及CinemachineBrain。这就是你必须做的。现在，无论何时激活vcam，其自定义后期处理配置文件都将应用于Unity相机。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

## CinemachinePostFX

这种行为是Cinemachine与Post-Processing v1模块之间的联系。您必须安装后处理V1堆栈资产商店软件包才能使用此行为。

它有两种使用方式：

- 作为Unity相机的一个组件：它用作相机的CinemachineBrain和相机的后期处理行为之间的联络。它侦听摄像机剪切事件，并在发生后处理堆栈时重置它。如果您正在使用后期处理，那么您应该将此行为始终与CinemachineBrain一起添加到相机中。
- 作为虚拟摄像机的组件：在此功能下，它拥有后处理配置文件资产，只要虚拟摄像机处于运行状态，该资产就会应用于Unity摄像机。它还具有（临时）可选功能，可将摄像机状态的Focus Distance和DepthOfField属性动画化，并将它们应用于当前的后处理配置文件。

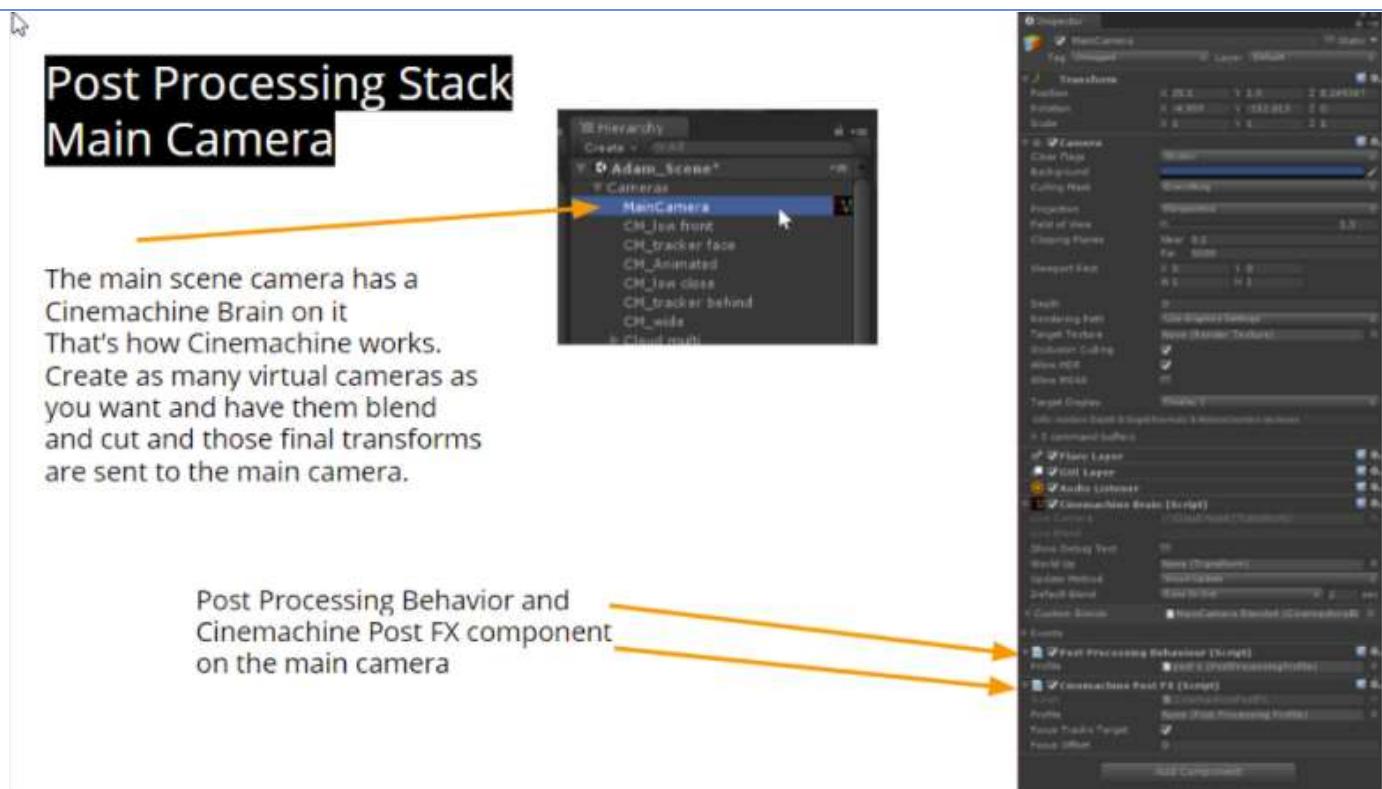
设置	类型	描述
轮廓	后处理配置文件	当此行为位于Unity相机上时，此设置是相机的默认后期处理配置文件，并且只要未被虚拟相机覆盖即可应用。当行为在虚拟相机上时，则这是后处理配置文件，只要此虚拟相机处于活动状态，该配置文件就会激活。
重点追踪目标	布尔	如果选中，则焦距设置为相机与LookAt目标之间的距离。
焦点偏移	单	偏离目标距离，与焦点轨道目标一起使用。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

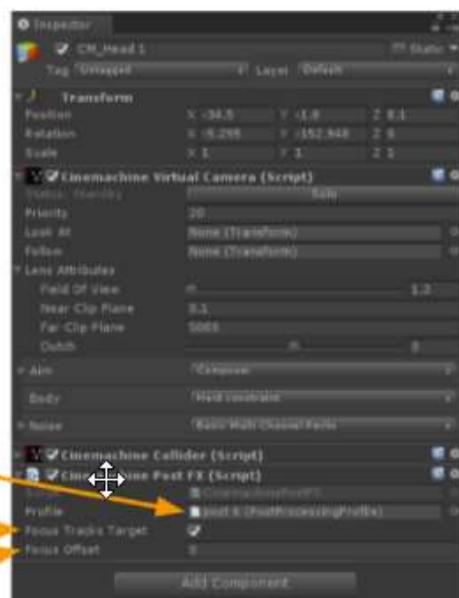


## Post Processing Stack Cinemachine cameras

You can have custom Post FX on each/any Cinemachine camera by simply adding a Cinemachine Post FX component and using a different PostProcessingProfile

Set the focus distance to automatically track the target and adjust focus plane

Animate the Focus Offset on Timeline for nice rack focuses or some human error micro adjust mojo



## 后处理堆栈V2

如果您安装了Unity的后处理堆栈V2软件包，则可以为虚拟相机添加后处理配置文件。配置文件将与相机一起混合使用。

为您的vcam定义后期处理配置文件，并通过将**CinemachinePostProcessing**扩展程序附加到**vcam**进行连接。要完成连接，您还需要Unity摄像机上的PostProcessingLayer组件，以及CinemachineBrain。

这就是你必须要做的。现在，无论何时激活vcam，其自定义后期处理配置文件都将应用于Unity相机。如果相机与另一个vcam混合，则混合权重也将应用于后期处理效果。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

## CinemachinePostProcessing

此行为是Cinemachine与后处理V2模块之间的联系。您必须安装后处理堆栈V2统一软件包才能使用此行为。

作为虚拟相机的一个组件，它拥有一个后处理配置文件资产，只要虚拟相机处于运行状态，该配置文件就会应用于相机。它还具有动画处理摄像机状态的Focus Distance和DepthOfField属性的可选功能，并将它们应用到当前的后处理配置文件。

设置	类型	描述
轮廓	后处理配置文件	当此行为位于Unity相机上时，此设置是相机的默认后期处理配置文件，并且只要未被虚拟相机覆盖即可应用。当行为在虚拟相机上时，则这是后处理配置文件，只要此虚拟相机处于活动状态，该配置文件就会激活。
重点追踪目标	布尔	如果选中，则焦距设置为相机与LookAt目标之间的距离。
焦点偏移	单	偏离目标距离，与焦点轨道目标一起使用。

## 其余

本节介绍了Cinemachine附带的其他组件。

## CinemachinePOV

这是组件管道目标部分中的CinemachineComponent。它的工作是将摄像机对准用户的鼠标或操纵杆输入。

作曲家不改变相机的位置。它只能平移和倾斜摄像机，以获得所需的帧。要移动相机，您必须使用虚拟相机的“正文”部分。

设置	类型	描述
垂直轴	AxisState	垂直轴。值是-90..90。控制垂直方向。
垂直重新打包	Recentering	控制垂直轴的自动重新调整是如何完成的。
横轴	AxisState	水平轴。值是-180..180。控制水平方向。
水平重新纳入	Recentering	控制如何自动重新调整水平轴的完成。

## CinemachineSameAsFollowTarget

这是组件管道目标部分中的CinemachineComponent。它的工作是匹配Follow目标的方向。

## CinemachineSmoothPath

定义一个世界空间路径，由一系列航点组成，每个航点都有位置和滚转设置。Bezier内插在航点之间执行，以获得平滑和连续的路径。路径将通过所有路点，并且（与CinemachinePath不同）保证一阶和二阶连续性。

设置	类型	描述
✓	布尔	启用此功能后，将更改此行为在按顺序排列的航点之间形成一个连续的循环。



已翻译为以下语言: 中文(简体)

显示原文

选项 ▼

设置	类型	描述
航点	航点[]	定义路径的路标。它们将使用贝塞尔曲线进行插值。
解析度	INT32	每个航点的路径样本。这用于计算路径距离。
出现	出现	控制路径将如何显示在编辑器场景视图中的设置。

## CinemachineFollowZoom

Cinemachine虚拟摄像机的附加模块，无论摄像机和目标位置如何，都可以调整镜头的FOV以将目标物体保持在屏幕上不变的大小。

设置	类型	描述
宽度	单	以世界单位维持目标距离的射门宽度。
减震	单	增加此值可以减轻后续缩放的攻击性。小数字响应速度更快，数字越大，响应速度越慢。
Min FOV	单	此行为将生成的FOV的下限。
最大FOV	单	该行为将生成的FOV的上限。

## CinemachineDollyCart

这是一个非常简单的行为，它将其转换为CinemachinePath。它可用于为路径上的任何物体制作动画，或作为Cinemachine虚拟相机的追踪目标。

设置	类型	描述
路径	CinemachinePathBase	遵循的道路。
更新方法	UpdateMethod	<p>何时移动购物车，如果速度不为零。  <b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>更新</b>: 在正常的MonoBehaviour更新中更新。</li> <li>- <b>更新更新</b>: 与固定更新中的物理模块同步更新。</li> </ul>
位置单位	位置单位	<p>如何解释路径位置。如果设置为路径单位，则值如下所示：0代表路径上的第一个航点，1代表第二个航点，依此类推。中间值是航点之间的路径上的点。如果设置为距离，则路径位置表示沿路径的距离。  <b>可能的值：</b></p> <ul style="list-style-type: none"> <li>- <b>路径单位</b>: 使用PathPosition单位，其中0是第一个航点，1是第二个航点等。</li> <li>- <b>距离</b>: 使用沿路径的距离。路径将根据其分辨率设置进行采样，并且距离查找表将在内部高速缓存。</li> </ul>
速度	单	沿着路径以这个速度移动购物车。该值根据位置单位设置进行解释。
位置	单	沿着购物车将被放置的路径的位置。这可以直接设置动画，或者如果速度不为零，则会自动更新。该值根据位置单位设置进行解释。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

CinemachineMixingCamera是一款“经理相机”，它可以呈现其子虚拟相机状态的加权平均状态。

固定数量的插槽可用于摄像机，而不是动态阵列。我们这样做是为了支持时间轴上的权重动画。时间轴不能动画数组元素。

设置	类型	描述
<b>重量0</b>	单	第一台跟踪摄像机的重量。
<b>重量1</b>	单	第二台跟踪摄像机的重量。
<b>重量2</b>	单	第三台跟踪摄像机的重量。
<b>重量3</b>	单	第四台跟踪摄像机的重量。
<b>重量4</b>	单	第五台跟踪摄像机的重量。
<b>重量5</b>	单	第六台跟踪摄像机的重量。
<b>重量6</b>	单	第七台跟踪摄像机的重量。
<b>重量7</b>	单	第八台跟踪摄像机的重量。
<b>优先</b>	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## CinemachineBlendListCamera

这是一个虚拟相机“经理”，拥有并管理一系列儿童虚拟相机。当照相机启动后，这些小孩摄像头会一个接一个地启用，并将每个摄像头保持指定的时间。指定相机之间的混合。最后一台相机无限期地举行。

设置	类型	描述
<b>看着</b>	转变	如果没有在儿童相机中指定，则相机儿童可以查看的默认对象（目标目标）。如果所有的孩子都定义他们自己的目标，则可能是空的。
<b>跟随</b>	转变	如果未在子摄像机中指定，则相机儿童的默认对象需要与（身体目标）一起移动。如果所有的孩子都定义他们自己的目标，则可能是空的。
<b>显示调试文本</b>	布尔	启用后，当前的儿童相机和混合将在游戏窗口中显示，供调试。
<b>启用所有儿童相机</b>	布尔	强制启用所有子摄像头。如果在时间轴中对它们进行动画处理，这会非常有用，但会消耗额外的资源。
<b>说明</b>	指令[]	用于启用儿童相机的一组说明。
<b>优先</b>	INT32	优先级将根据其他相机和本相机的状态确定哪个相机变为活动状态。数字越大，优先级越高。

## CinemachineExternalCamera

这个组件将把一台非电影摄影机暴露给Cinemachine系统，允许它参与混合。只需将它作为一个组件与现有的Unity Camera组件一起添加即可。

设置	类型	描述
<b>看着</b>	转变	相机正在看的物体。设置此项可以提高本相机混合物的质量。



已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

# CinemachineStoryboard

Cinemachine虚拟摄像机的附加模块，可将图像放在摄像机输出的屏幕空间中。

设置	类型	描述
<b>显示图像</b>	布尔	如果选中，指定的图像将显示为覆盖虚拟相机的输出。
<b>图片</b>	质地	要显示的图像。
<b>方面</b>	FillStrategy	如何处理图像方面和屏幕方面的差异。 可能的值： - <b>最适合</b> ：图像在屏幕上尽可能大，而不会被裁剪。 - <b>适合裁剪图像</b> ：如果需要，图像将被裁剪，以便屏幕完全填满。 - <b>拉伸以适应</b> ：图像将被拉伸以覆盖任何与屏幕不匹配的方面。
<b>A</b>	单	图像的不透明度。0是透明的，1是不透明的。
<b>中央</b>	Vector2	显示图像的屏幕空间位置。零是中心。
<b>回转</b>	的Vector3	要应用于图像的屏幕空间旋转。
<b>规模</b>	Vector2	屏幕空间缩放应用于图像。
<b>同步比例</b>	布尔	如果选中，则X和Y比例同步。
<b>静音相机</b>	布尔	如果选中，相机变换将不受此虚拟相机控制。
<b>拆分视图</b>	单	水平擦拭图像。

## SaveDuringPlay

相机需要在上下文中进行调整 - 通常这意味着游戏正在播放。通常情况下，一旦播放模式退出，Unity不会将这些更改传播到场景中。Cinemachine已经实现了一项特殊功能来保存游戏中进行的参数调整。它不会保存结构变化（如添加或删除行为），但它会保持调整。Cinemachine行为有一个特殊的属性[SaveDuringPlay]来启用此功能。如果您需要，也可以在自己的脚本上随意使用它。

它通过在Play退出后扫描场景并将任何更改的参数应用回场景来工作。这退出后一秒左右踢。您始终可以点击撤消来恢复更改。

如果您想要从Play-Mode保存中排除的行为中存在特定设置，则可以向它们添加[NoSaveDuringPlay]属性，并且它们将被跳过。

此功能可通过选中任何虚拟摄像机检查器上的保存播放复选框启用。这是一个静态的全局设置，而不是每个摄像头，所以你只需要检查一次。

## 时间轴集成

1. 将启用Cinemachine的Unity相机（即带有CinemachineBrain组件的相机）拖到时间轴上，从弹出菜单中选择创建Cinemachine轨道。
2. Cinemachine轨道是启用/禁用样式轨道，这意味着片段将在时间线中处于活动状态时启用Cinemachine虚拟相机。
3. 重叠Cinemachine拍摄剪辑将产生混合。
4. 右击Cinemachine轨道并选择Create Cinemachine Shot Clip创建一个Cinemachine镜头。
5. 将场景中的虚拟相机拖到Cinemachine Shot Clip检查器上。这将使镜头剪辑与虚拟相机相关联。
6. 虚拟相机不需要在场景中启用。剪辑在时间线中外于活动状态时，该相机将变为启用状态。

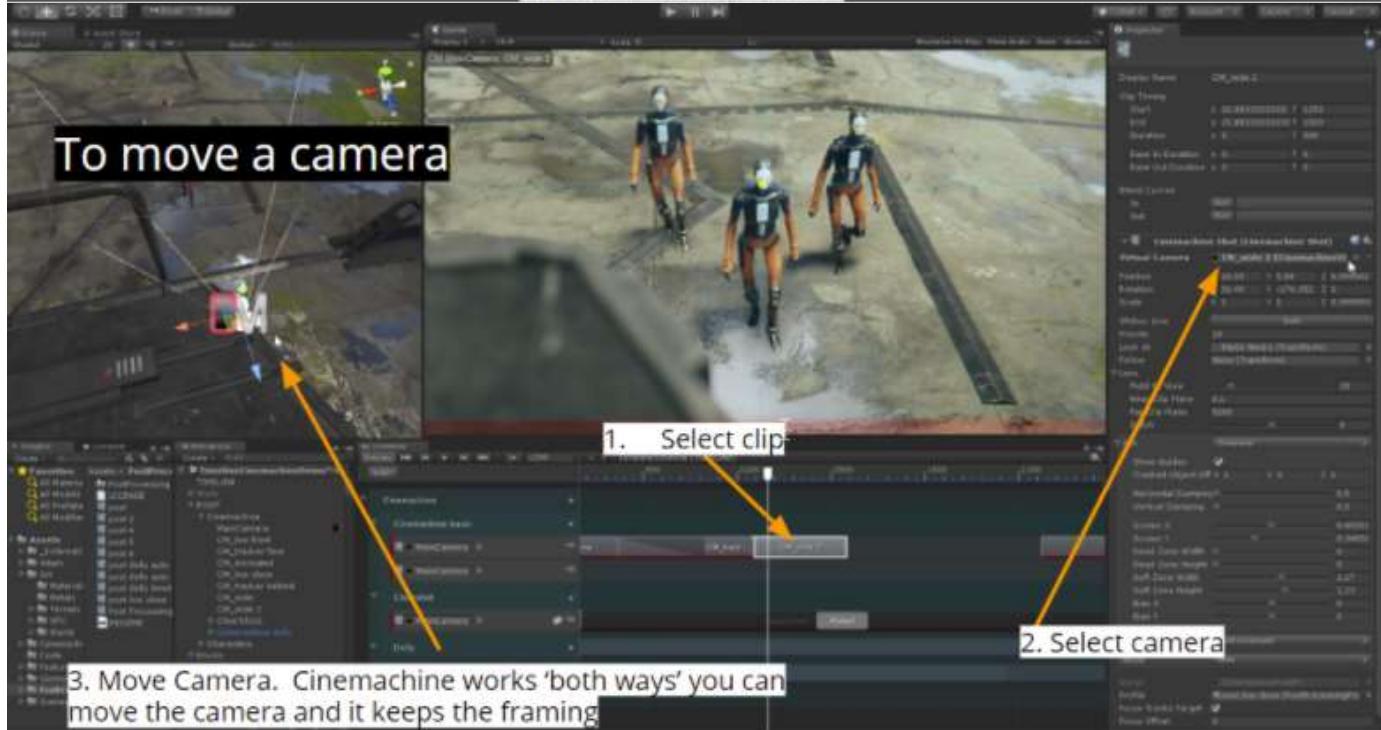
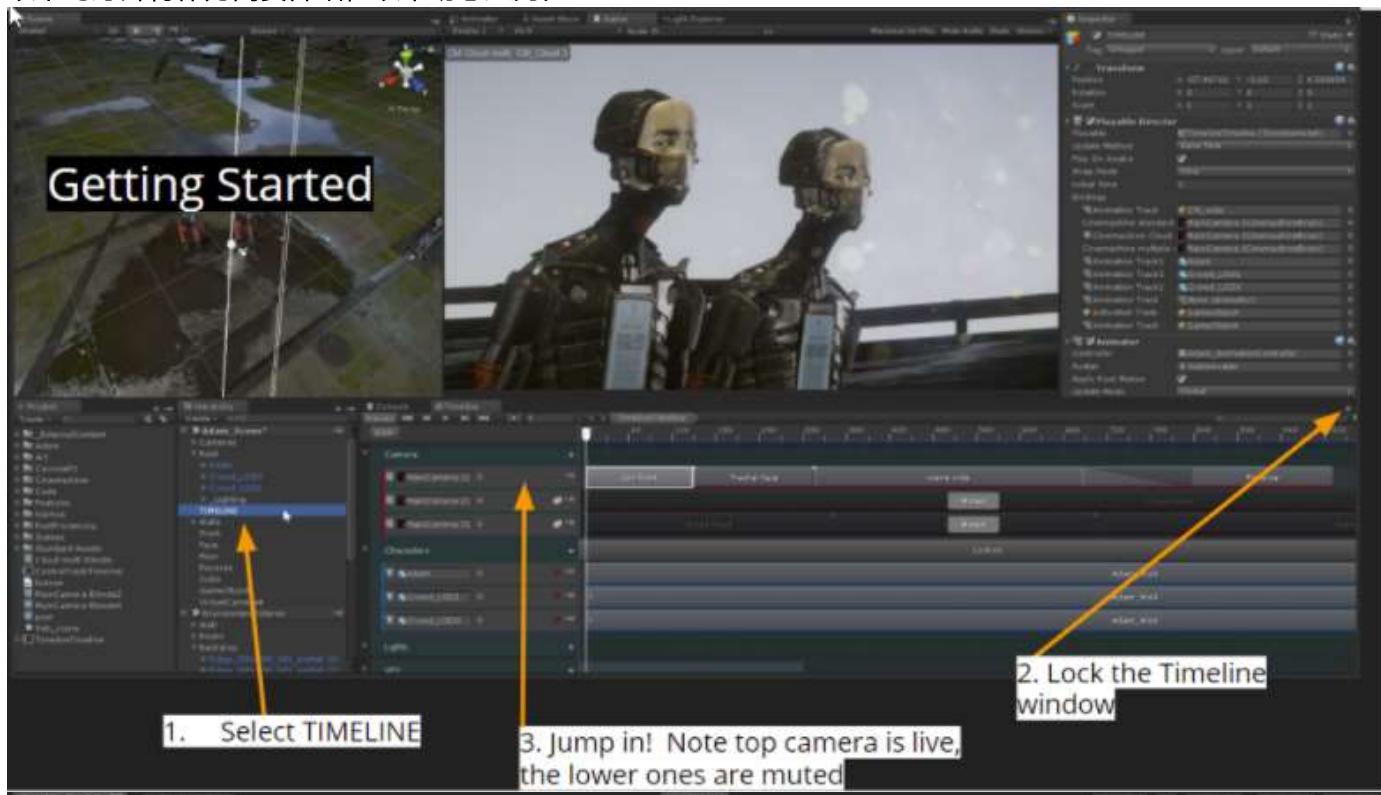


已翻译为以下语言： 中文(简体)

显示原文

选项 ▼

以下幻灯片将作为简要介绍，以帮助您入门。

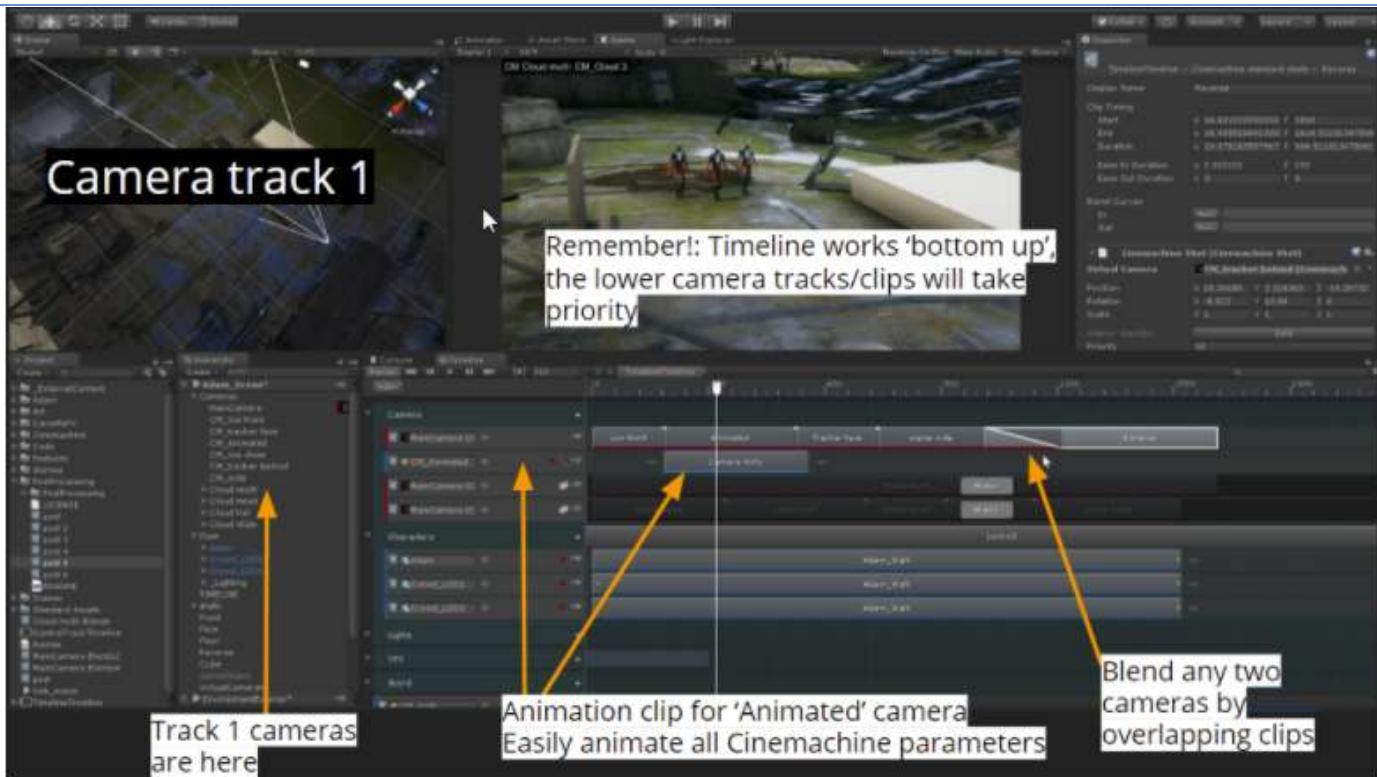




已翻译为以下语言： 中文(简体)

显示原文

选项 ▼



## 特殊环境

### 自上而下的游戏

Cinemachine虚拟相机以人类相机操作员为模型。因此，他们对上/下轴有一定的敏感度，并且会尽量避免将滚动引入摄像机画面 - 除非您通过诸如荷兰语等设置故意引入它。由于这种敏感性，虚拟相机不喜欢长时间直观或向下看。他们会顺便通过，但如果LookAt目标往往是直线上升或下降，他们并不总是会给出预期的结果。

如果您正在制作自上而下的游戏，则最佳做法是重新定义向上方向，用于相机目的。您可以通过将CinemachineBrain中的World Up Override设置为一个游戏对象来实现此目的，该游戏对象的局部向上点指向您希望摄像机正常运行的方向。这将适用于连接到该大脑的所有虚拟相机。

## 分屏

您可以使用Cinemachine 2.0设置多相机分屏：

1. 制作2台Unity摄像机，为每台摄像机分配自己的CinemachineBrain，并设置其视口。
2. 现在让2个虚拟相机跟随玩家。将这些虚拟摄像机分配给不同的图层。我们将它们称为A层和B层。
3. 返回到两台Unity摄像机，并设置他们的剔除掩模，以便一台摄像机看到A层而不是B层，而另一台摄像机则相反。
4. 而已！相机A将由层A上的虚拟相机驱动，相机B将由层B上的虚拟相机驱动。他们将独立进行混合等。
5. 根据需要添加尽可能多的图层和相机。

## 脚本

Cinemachine是一个pure-csharp实现，完全可以通过脚本访问。检查器中可见的所有设置在csharp类中都有相应的字段。此外，还有一个完整的脚本API，您可以通过检查实施过程中经过良好评论的源代码，或通过查看示例中的脚本编写场景或查阅Cinemachine附带的API文档并在线获得来发现。

已翻译为以下语言： [中文\(简体\)](#)[显示原文](#)[选项 ▼](#)

日期	原因
2018年2月15日	更新了2.1.11
2017年11月21日	初始版本。