**粒子系统**

在 3D 游戏中，大多数角色、道具和景物元素都表示为\_\_网格\_\_，而 2D 游戏则使用\_\_精灵\_\_来表现它们。网格和精灵是描绘具有明确形状的“实体”对象的理想方式。然而，游戏中还有其他存在物，它们本质上是流动和无形的，因此很难用网格或精灵来描绘。对于流动性的液体、烟雾、云、火焰和魔法等效果，可使用不同的图形方法（称为\_\_粒子系统\_\_）来捕捉固有的流动性和能量。本部分将介绍 Unity 的粒子系统及其可能的用途。

# 什么是粒子系统？

\_\_粒子\_\_是由粒子系统大量呈现或移动的简单小型图像或网格。每个粒子代表一小部分流体或无定形实体，而众多粒子将共同营造出完整的实物感。以烟幕云团为例，每个粒子都具备细小烟雾的质感，自身形态类似于微型云团。在场景内，大量的这些微型云团通过排列组合，营造出体积更大的整体云团效果。

## 系统动态性

每个粒子都有预设的\_生命周期\_，通常为几秒钟，期间会经历各种变化。粒子由粒子系统产生或者\_发射\_时，即开始生命周期。该系统在球体、半球体、圆锥体、盒体或任意网格之类的空间区域内随机发射粒子。生命周期结束之后，粒子便不再显示，并被移出系统。系统的\_发射速率\_表示每秒大约发射的粒子数，不过具体发射次数随机微调。处于“稳定”状态（即，粒子的发射速率和死亡速率相同）的粒子数以及系统达到该状态所需时长取决于粒子的发射速率和平均生命周期。

## 粒子动态性

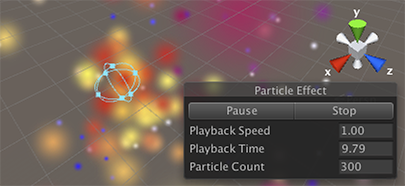
发射和生命周期设置会影响系统的整体行为，不过粒子个体也会随时间变化。每个粒子都有其\_\_速度\_\_矢量，该矢量决定了粒子在每一帧更新期间移动的方向和距离。系统自身采用的\_\_作用力\_\_和\_\_重力\_\_或者地形 (Terrain) 上将粒子吹散的\_\_风区\_\_都可以改变该速度。每个粒子的颜色、大小和旋转方式也可能在其生命周期内变化，或与其当前的移动速度成比例。粒子的颜色包括 Alpha（透明度）分量，因此粒子可逐渐淡入和淡出，而非突然出现和消失。

多种粒子动态性混合运用时，可以将多种流体效果模拟得栩栩如生。例如，利用稀薄的发射形状，使水粒子单纯受重力下落并逐渐加速，即可模拟瀑布。火堆冒出的烟雾往往会上升、扩散并最终消逝，所以系统应为烟雾粒子设置升力，并随时间的推移增大其体积和透明度。

# 在 Unity 中使用粒子系统

Unity 使用组件实现粒子系统，因此将粒子系统放置在场景中涉及到添加预制的游戏对象（菜单：\_\_GameObject\_\_ > **Effects** > **Particle System\_\_）或将组件添加到现有游戏对象（菜单：\_\_Component** > **Effects** > **Particle System\_\_）。由于组件非常复杂，因此 Inspector 分为多个可折叠子部分或\_\_模块\_\_，每个子部分或模块都包含一组相关属性。此外，可使用单独的 Editor 窗口（通过 Inspector 中的** Open Window\_\_ 按钮访问）同时编辑一个或多个系统。请参阅有关[粒子系统组件](file:///E:\\UnityDocumentation\\Manual\\class-ParticleSystem.html)和各个[粒子系统模块](file:///E:\UnityDocumentation\Manual\ParticleSystemModules.html)的文档，了解更多信息。

选择带有粒子系统的游戏对象时，Scene 视图包含一个小的 **Particle Effect** 面板，其中有一些简单控件，用于显示对系统设置的更改。



**Playback Speed** 用于加快或减慢粒子模拟速度，可以直观查看在高级状态下的效果。**Playback Time** 表示自系统启动以来经过的时间；这可能比实时更快或更慢，具体取决于播放速度。**Particle Count** 表示系统中当前有多少粒子。通过单击 **Playback Time** 标签并向左和向右拖动鼠标，即可前后移动播放时间。面板顶部的按钮可用于暂停和恢复模拟，或停止模拟并重置为初始状态。

## 随时间推移而变化的属性

粒子甚至整个粒子系统的许多数字属性都可能随时间而变化。Unity 提供了几种不同的方法来指定这种变化的发生方式：

* \_\_Constant：\_\_属性的值在其整个生命周期内是固定的。
* \_\_Curve：\_\_该值由曲线/图形指定。
* \_\_Random Between Two Constants：\_\_两个常量值定义了值的上限和下限；实际值随着时间的推移在这些边界之间随机变化。
* \_\_Random Between Two Curves：\_\_两条曲线定义了值在生命周期内给定点的上限和下限；当前值在这些边界之间随机变化。

同样，主模块中的 **Start Color** 属性具有以下选项：

* \_\_Color：\_\_粒子初始颜色在整个系统的生命周期内是固定的。
* \_\_Gradient：\_\_使用渐变指定的初始颜色发射粒子，渐变表示粒子系统的生命周期。
* \_\_Random Between Two Colors：\_\_选择两种给定颜色之间的随机线性插值作为初始粒子颜色。
* \_\_Random Between Two Gradients：\_\_在对应于系统当前时期的点处从给定渐变中挑选两种颜色；选择这两种颜色之间的随机线性插值作为初始粒子颜色。

对于其他颜色属性，例如 \_\_Color over Lifetime\_\_，有两个单独的选项：

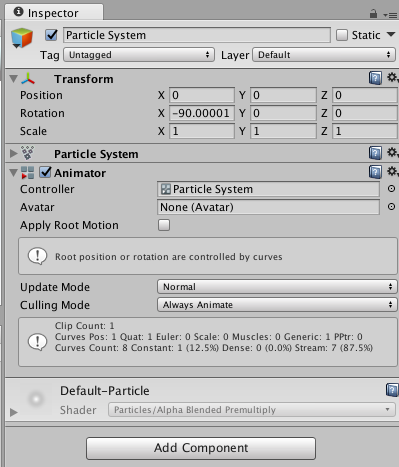
* \_\_Gradient：\_\_颜色值取自表示粒子系统生命周期的渐变。
* \_\_Random Between Two Gradients：\_\_在对应于粒子系统当前时期的点处从给定渐变中挑选两种颜色；选择这两种颜色之间的随机线性插值作为颜色值。

各种模块中的颜色属性按照每个通道相乘，从而计算出最终的粒子颜色结果。

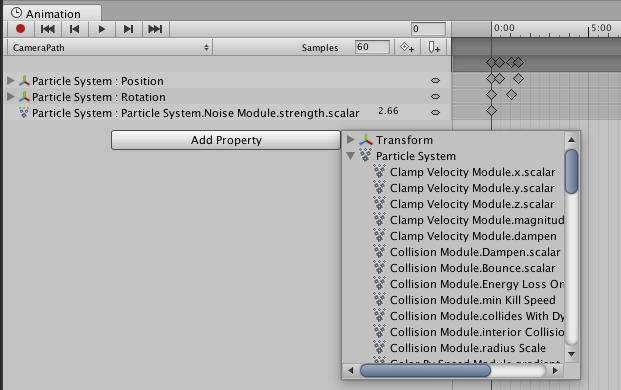
## 动画绑定

动画系统可以访问所有粒子属性，这意味着可以将它们设置到关键帧中并从动画中控制它们。

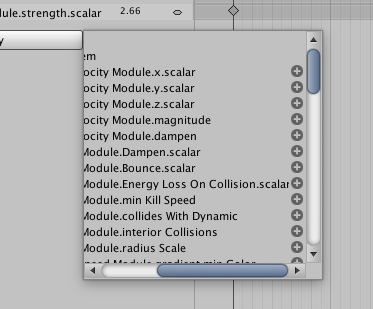
要访问粒子系统的属性，必须有一个 Animator 组件连接到粒子系统的游戏对象。此外还需要动画控制器 (Animation Controller) 和动画。

要动画化粒子系统，请添加 Animator 组件，并为动画控制器 (Animator Controller) 分配动画。

要动画化粒子系统属性，请打开 **Animation 窗口\_\_，并选择包含 Animator 和粒子系统的游戏对象。单击** Add Property\_\_ 以添加属性。

在 Animation 窗口中为动画添加属性。

向右滚动以显示\_\_添加控件\_\_。



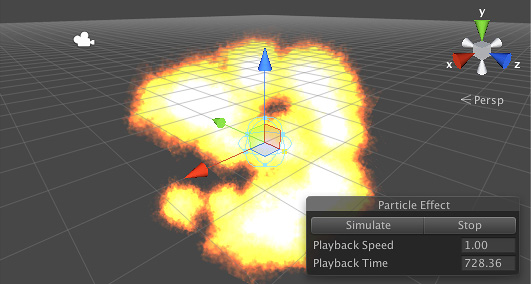
请注意，对于曲线，只能对整体\_\_曲线乘数\_\_进行关键帧设置（可在 **Inspector** 中的曲线编辑器旁边找到该曲线乘数）。

# 粒子系统操作方法

本部分介绍如何实现常见的粒子系统类型。正如文档中所有的代码一样，在使用时无需知会 Unity。

# 简单爆炸

可使用粒子系统来创建令人信服的爆炸效果，但动态性可能比最初看起来要复杂一些。爆炸的核心就是向外爆发的大量粒子，可以应用一些简单的修改使效果显得更真实。

正在发生的粒子系统爆炸

## 粒子的时间轴

一次简单爆炸会产生在各个方向迅速向外扩展的一团火焰。最初爆发有很多能量，因此温度非常高（即亮度高）并且移动得非常快。此能量迅速消散，导致火焰膨胀速度减慢并且温度降低（即亮度降低）。最后，随着所有燃料燃尽，火焰将消失并很快完全消失。

爆炸粒子的生命周期通常较短，可以在该生命周期内改变几种不同的属性来模拟效果。粒子将以非常快的速度开始移动，但随着从爆炸中心移开，其速度应该会大大降低。此外，颜色应该从最开始的明亮逐渐变暗，并最终淡化为透明状态。最后，在粒子的生命周期内逐渐减小其大小将产生火焰随着燃料耗尽而消散的效果。

## 实现

从默认粒子系统对象（菜单：\_\_GameObject > Effects > Particle System\_\_）开始，选择 Shape 模块，并将发射器形状设置为小型\_球体 (Sphere)，比如半径约为 0.5 个单位。标准资源中的粒子包括一种名为 ***ParticleFireball*** 的材质，这种材质非常适合用于爆炸（菜单：\_\_Assets > Import Package > ParticleSystems\_\_）。可以使用 Renderer\_ 模块为系统设置此材质。 在打开 Renderer 后，还应禁用 Cast Shadows 和 \_Receive Shadows\_，因为爆炸火焰应该发出光而不是接受光。

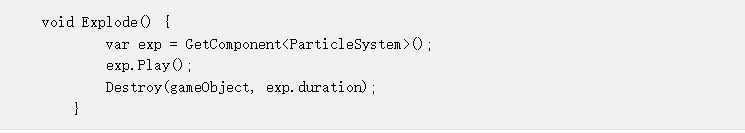
在此阶段，系统看起来像是从中心点抛出的许多小火球。当然，爆炸应该同时产生大量粒子爆发的效果。在 Emission 模块中，可将 Rate 值设置为零，并在时间为零的位置添加一个粒子 Burst 状态。爆发中的粒子数量取决于需要的爆炸规模和强度，但一般在开始时使用大约五十个粒子是一种不错的选择。设置好爆发后，系统现在开始看起来很像爆炸了，但速度相当缓慢，火焰似乎长时间停留不动。在粒子系统模块（名称与游戏对象相同，例如“Explosion”）中，请将系统的 Duration 和粒子的 Start Lifetime 均设置为两秒。

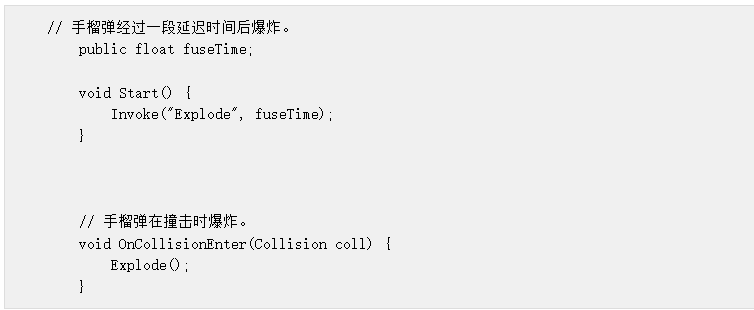
此外还可以使用 Size Over Lifetime 模块来创建随着燃料耗尽而变化的火焰效果。应使用“斜降”(ramp down) 预设来设置大小曲线（即，大小从 100% 开始并减小到零）。要使火焰变暗和淡化，请启用 Color Over Lifetime 模块，并将渐变设置为在左侧以白色开始，然后在右侧以黑色结束。Fire Add 材质使用附加着色器进行渲染，因此颜色属性的暗度也控制着粒子的透明度；随着颜色逐渐变黑，火焰将变得完全透明。而且，附加材质允许粒子的亮度在粒子彼此重叠产生时“加”在一起。当粒子全部靠近到一起时，这种功能有助于进一步增强在爆炸开始时出现的明亮闪光效果。

到目前为止，爆炸初步成形，但看起来好像是在太空中发生爆炸。粒子首先被发射出来，然后在消失之前以恒定速度长距离行进。如果游戏设定是在太空中，那么这种效果是很贴切的。但是，在大气中发生的爆炸会被周围的空气减缓和减弱。启用 Limit Velocity Over Lifetime 模块，然后将 Speed 设置为大约 3.0 并将 Dampen 分数设置为大约 0.4，这样应该会看到爆炸发生后逐渐失去一些能量。

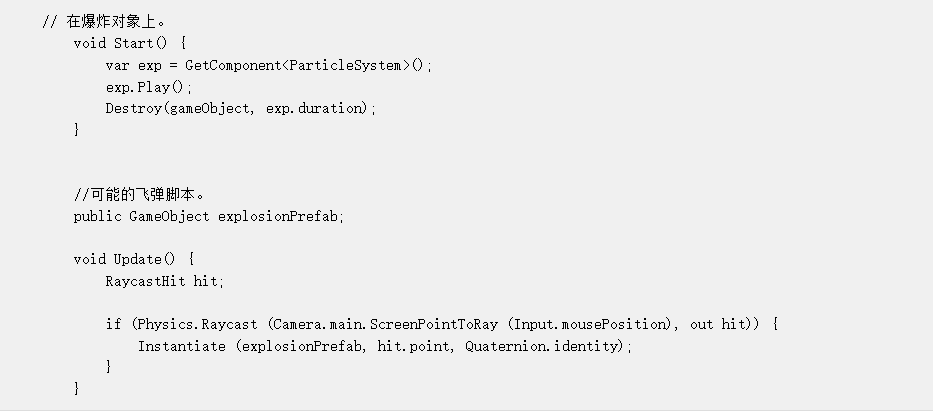
最后要注意的是，当粒子从爆炸中心移开时，它们各自的形状变得更容易识别。也就是说，此时会看到所有粒子具有相同的大小和相同旋转，感觉很明显每个粒子重复使用了相同的图形。为了避免这种情况，一种简单方法是在粒子生成时为粒子的大小和旋转添加一些随机变化。在检视面板顶部的粒子系统模块中，单击 Start Size 和 Start Rotation 属性右侧的小箭头，并将这两者均设置为 Random Between Two Constants。对于旋转，请将两个值设置为 0 和 360（即完全随机旋转）。对于大小，请将值设置为 0.5 和 1.5 以便呈现一些变化，而不会有太多巨大或微小粒子的风险。现在将会看到，粒子图形的重复问题现在已经不显示。

## 用途

在测试过程中，开启 Looping 属性很有用，这样可以反复看到爆炸，但在已完成的游戏中，应关闭该属性，使爆炸只发生一次。为有可能爆炸的对象（例如油箱）设计爆炸时，需要将粒子系统组件添加到该对象并禁用 Play On Awake 属性。然后，可根据需要从脚本中启用爆炸。

在其他一些情况下，爆炸发生在撞击点。如果爆炸来自对象（例如手榴弹），则可在延迟一段时间后或在与目标接触时调用上述 Explode 函数。

如果爆炸来自实际上并未在游戏中表示出来的对象（例如，飞行速度太快而无法看到的飞弹），可在适当的位置实例化爆炸。例如，可以确定[射线投射](file:///E:\\UnityDocumentation\\ScriptReference\\Physics.Raycast.html)的接触点。



## 思路拓展

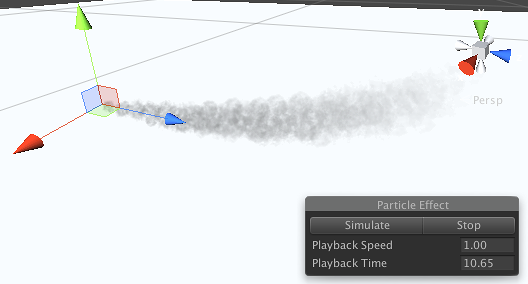
此处介绍的爆炸是非常基本的，但可对其进行各方面的修改，从而在游戏中获得所需的确切效果。

使用的粒子图形将在很大程度上影响玩家对爆炸的“解读”。出现大量可独立识别的小型火焰表明抛出了燃烧碎片。不完全分离的较大粒子看起来更像是被破坏的油箱激发的火球。通常，需要一起更改多个属性才能达到效果。例如，火球会在消失之前持续更长时间并且膨胀更少，而剧烈的爆发可能会使燃烧碎片散开相当远的距离。

此处一些属性设置为随机值，但是其他许多属性有 Random Between Two Constants/Curves 选项，因此可通过各种方式添加变化。改变大小和旋转有助于避免最明显的粒子重复效果，但也可以考虑在 Start Delay、Start Lifetime 和 Start Speed 属性中添加一些随机性。少量的变化有助于使爆炸看起来更像是一种“自然”和不可预测的效果，而非受控的机械过程。较大的变化表明是“脏”爆炸。例如，改变 Start Delay 将产生不再剧烈而爆发速度更慢的爆炸，比如可能是因为车辆中的油箱被单独点燃。

# 车辆的排气烟雾

汽车和许多其他车辆在将燃料转化为动力时会产生排气烟雾。可以使用粒子系统为车辆添加排气效果以作为巧妙的点睛之笔。

由粒子系统生成的排气效果

## 粒子的时间轴

排气烟雾从管道中快速排出，但随后与大气接触时会迅速减慢速度。在减速过程中，排气烟雾会扩散，变得稀薄，很快消散在空气中。由于排气温度很高，当它通过周围较冷的空气时，还会略微上升。

排气烟雾的粒子在开始时不得超过管道宽度，但随后会在其短暂的生命周期内显著增大。它通常在开始时为部分透明状态，随后与空气混合后逐渐淡化至完全透明状态。在动态性方面，粒子将以相当快的速度发射，但随后迅速减速，并还会略微上升。

## 实现

在 Shape 模块中，选择 Cone 形状并将其 Angle 属性设置为零；在这种情况下，“锥体 (Cone)”实际上是圆柱形管。该管道的 Radius 自然取决于车辆的大小，但通常可通过将 Scene 视图中的半径辅助图标与车辆模型相匹配来设置值（例如，车辆模型通常在尾部有排气管或孔，可与其大小进行匹配）。半径实际上在很大程度上决定了属性设置的选择，例如粒子大小和发射速率。在本示例中，假设车辆遵循 Unity 的标准尺寸惯例，即一个世界单位为一米；因此，半径设定为约 0.05 或 5cm。

标准资源中的 Smoke4 材质提供了烟雾粒子的合适图形。如果尚未安装这些资源，请从菜单中选择 **Assets > Import Package > Particles**。然后，选择粒子系统的 Renderer 模块，并将 Material 属性设置为 Smoke4。

五秒的默认生命周期通常对于汽车尾气而言太长，因此应打开粒子系统模块（其名称与游戏对象相同，例如“Exhaust”），并将 Start Lifetime 设置为大约 2.5 秒。同样在此模块中，将 Simulation Space 设置为 World\_，并将 Gravity Modifier\_ 设置为较小的负值，例如约 –0.1。通过使用世界模拟空间，即使车辆移动，烟雾也可停留在产生烟雾的位置。负重力效应导致烟雾粒子上升，使它们看起来好像是由热气体组成的。一个很好的额外做法是使用 Start Rotation 旁边的小菜单箭头选择 Random Between Two Constants 选项。将两个值分别设置为 0 和 360，烟雾粒子将在发射时随机旋转。大量排列方式相同的粒子看起来非常显眼，并会降低随机无形烟雾痕迹的效果。

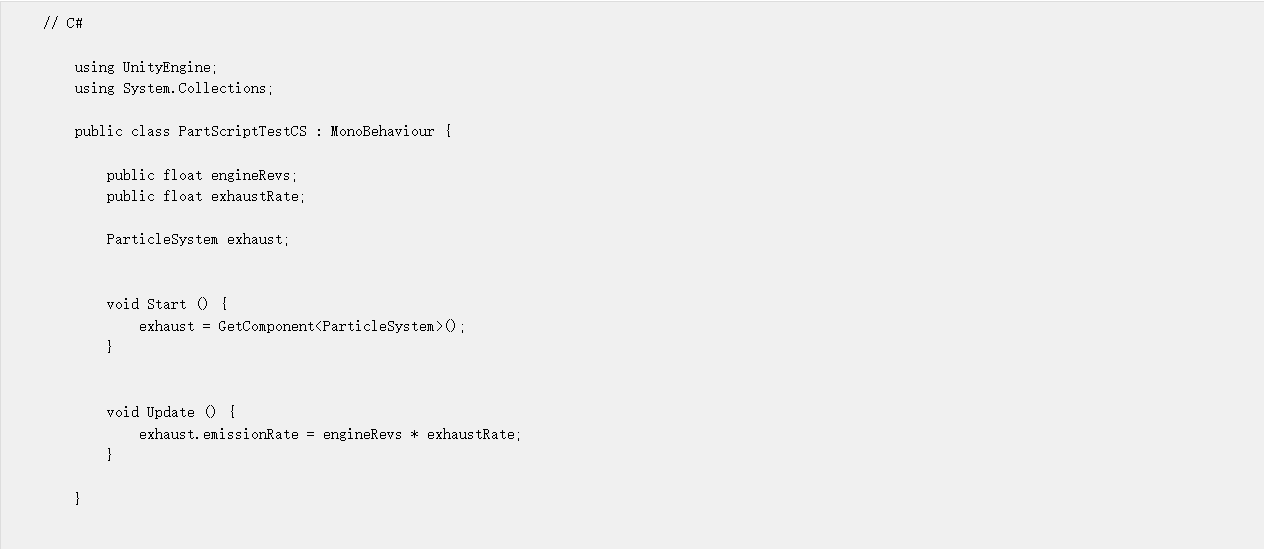
在此阶段，烟雾粒子开始看起来很逼真了，而且默认的发射速率会产生很好的发动机“嚓嘎声”效果。然而，烟雾并没有向外翻腾并消散。打开 Color Over Lifetime 模块，然后单击渐变右端的顶部渐变色标（此控件将控制颜色的“Alpha”值透明度）。将 Alpha 值设置为零，应该会看到场景中的烟雾粒子渐渐消失。根据发动机的洁净程度，可能还希望在开始时减小渐变的 Alpha 值；浓烟往往表示脏污、低效的燃烧物。

除了淡化外，烟雾在逃逸时还应该增大，可以使用 Size Over Lifetime 模块轻松创建这种效果。打开此模块，选择曲线并滑动左端的曲线控制柄，使粒子以最大大小的一个比例开始发射。选择的确切大小取决于排气管的大小，但略大于管道的值会产生良好的气体飘散效果。（发射与管道相同大小的粒子表明气体通过管道保持形状，虽然气体没有确定的形状。）在 Scene 视图中使用粒子系统的模拟功能可获得烟良好的雾外观视觉效果。此外，如果烟雾消散程度达不到期望的效果，可在此时增大粒子系统模块中的 Start Size。

最后，烟雾在消散过程中也会减慢速度。实现这一目标的一种简单方法是使用 Force Over Lifetime 模块。打开此模块，将 Space 选项设置为 \_Local\_，并将力的 Z 分量设置为负值，以指示粒子被力推回（系统沿对象的局部空间中的正 Z 方向发射粒子）。如果其他参数已按上述方式设置，则此处约 –0.75 的值对于系统便会非常有效。

## 用途

可以将排气粒子系统放置在主车辆的子对象上。对于简单游戏，可仅启用 Play On Awake 和 Looping 属性，然后让系统运行。但是，在大多数情况下，可能希望至少在车辆移动时改变粒子发射速率。这样做首先是为了真实性（即，发动机工作强度提高时产生更多烟雾），但也有助于防止烟雾粒子在车辆移动时散开。如果快速移动的车辆的发射速率太低，看起来会产生明显的“一股股”烟雾，而这是非常不真实的。

可以通过脚本轻易改变发射速率。如果脚本中有一个表示发动机转速或车辆速度的变量，则可简单地将该值乘以一个常量，然后将结果分配给 ParticleSystem 的 emissionRate 属性。

## 思路拓展

基本方案可产生相当逼真的排气烟雾效果，但可以注意到，当改变参数时，发动机的“特性”会发生变化。调整不当、效率低下的发动机往往不能完全燃烧燃料，因此会导致浓黑的烟雾在空气中持续弥漫很长时间。这种效果对于老旧的农用拖拉机来说是完美的，但对于高性能的跑车则不行。对于“洁净”的发动机，应使用较小的值来表示粒子的生命周期、不透明度和大小增加量。对于“脏”发动机，应增大这些值，可能还要使用 Emission 模块的 Bursts 属性来创建发动机噗噗声效果。