UE4与Unity特效对比

1. 总结概括

1、资源结构的问题，unity的粒子系统只是一个组件而已，需要添加到GameObject里打包成预设来作为资源使用，而ue4中的粒子系统则是一个资源。要理解ue4的粒子系统必须先要了解Emitter（粒子系统对象）、ParticleSystem（资源）、ParticleEmitter（粒子发射器）、ParticleModule（粒子模块）、ParticleSystemComponent（粒子系统组件）这几个概念，也就是说，一个粒子系统由若干个粒子发射器组成，而一个发射器可以添加不同的粒子模块来操作粒子的不同行为效果。Unity的粒子系统虽然也是发射器与粒子模块组成，但它的一个粒子系统只会有一个粒子发射器，跟UE4的不同。UE4的粒子系统详细介绍请参考UE4粒子系统相关介绍文档。

2、粒子类型方面，ue4可以在一个粒子系统里同时发射5中粒子类型，从而能制作出非常复杂的表现效果。而unity的粒子系统，在这方面只会支持有限的3种，在表现效果方面弱于ue4。

3、ue4支持GPU Sprite的粒子渲染，也就是说粒子的计算可以在Gpu上进行计算，区别于传统的cpu计算，利用 GPU 模拟可对成千上万个粒子进行高效模拟和渲染。而unity的粒子系统没这方面的支持，起码在2017的版本没有。

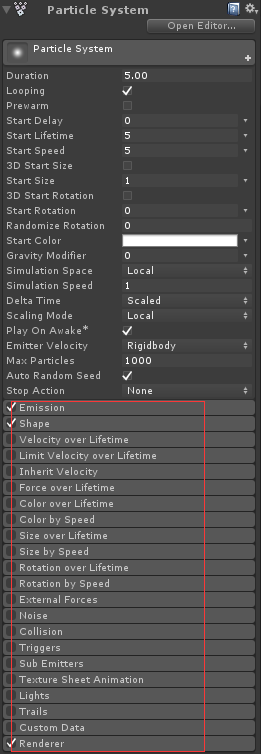
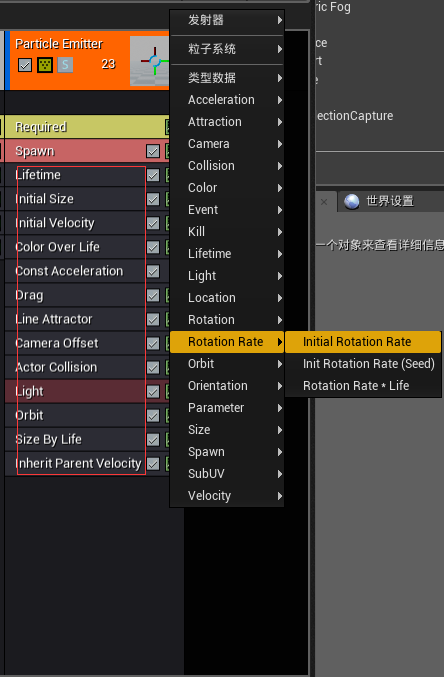
4、粒子系统Lod细节层次控制，ue4在编辑器这方面做了详细的支持，会根据设置不同的层级距离来是否显示粒子系统的其他细节或是否关闭某个发射器等等，对性能的提高有一定的帮助。对于unity的粒子系统在这方面就没有支持，虽然也可以通过其他嵌套等方面来实现lod的控制，但操作上复杂不方便，不想ue4那样，都在cascade编辑器上进行很便利的操作。

5、模块参数的多样化，ue4目前提供使用的粒子模块就有19大类，而每一类在每一种粒子类型下又包含若干的粒子模块子类，提供了更加细致、多样的参数设置，从而更加精确的控制粒子的运动行为。而unity提供的粒子模块就相对于简陋一点，可控的细节就没那么多。

总之，在操作使用和功能方面，ue4的cascade都略胜一筹，但unity的最新版本在粒子系统方面也不断添加功能模块，有向ue4方面靠拢，比如Lights和Trails这些粒子类型都在粒子系统上添加进去了。

接下来就是ue4与unity粒子系统各个功能的相同点与区别的详细介绍：

首先先上两张两个引擎的粒子系统的截图：

上面左边红色框的是unity的粒子系统的模块，而右边红色框的是ue4粒子系统的模块列表

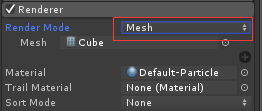
下面介绍一下ue4每个粒子模块相对应的unity（2017.3版本）的粒子系统的模块（如果有的话）

首先来的是UE4的粒子数据类型：

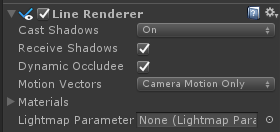
1. **sprite类型**，该类型为CPU计算模拟粒子运动，并使用Billboard面片渲染的粒子。这是ue4新建一个粒子发射器的默认类型，这也是unity粒子系统的默认方式。
2. **GPU Sprite** 类型，与Sprite类型效果类似，只是将模拟粒子运动的计算放在GPU端执行。相比CPU端模拟粒子运动，它的优势是减少了CPU端的开销，借助GPU的高性能并行计算，能够支持更多的粒子数目。GPU Sprite目前在PC端都支持，在中高端Mobile机型中也同样支持，即只要设备支持MRT功能即可。这个类型在Unity的粒子系统上是没有的。



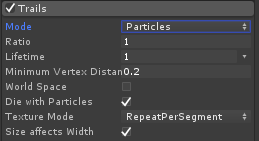
1. **Mesh类型**,就是使用静态网格数据来渲染粒子，对于碎片或是弹片的特效来说，是非常好用的。这个类型就类似于Unity的Renderer模块下的Render Mode 下的Mesh模式，如下图：



1. **Beam类型**, 是一种可实现在两个物体之间连线的粒子，并且可随着两物体之间距离变化进行拉伸或缩短，其效果类似于连接友军的灵魂链，或是带跟踪的激光束。Unity引擎同样也支持这种效果，但不在它的粒子系统里，而是作为一种单独的渲染效果，那就是LineRenderer。如下图所示



1. **Ribbon类型**，就是发射器将产生尾迹 - 连接粒子形成条带并按其生成顺序连接，粒子初始速度模式越不稳定，条带便越无序。Unity引擎也支持类似效果，就是Tails模块。如下图

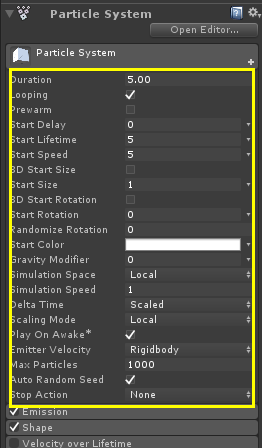


1. **AnimTrail类型**，能模拟拖尾效果。它可以挂接到角色动画的骨骼上，跟随挂接骨骼的运动轨迹形成拖尾效果。如果使用Unity引擎来做的话，就是使用TrailRenderer或Trails模块来做。

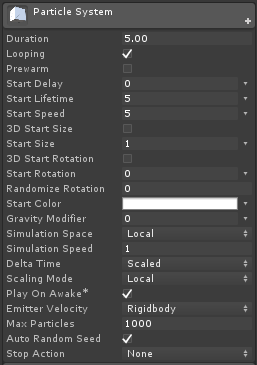
总之，ue4的每一种类型，除了GPU Sprite类型外，unity引擎都可以通过粒子系统或其他渲染类型来实现该效果。

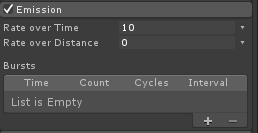
接下来就是具体的粒子模块对比：

Required模块，每个粒子发射器均包含此模块，该模块包含粒子发射器必需的所有属性。这个模块类似于unity粒子系统的初始化数据，如下图的黄色框的属性，注意该模块是不能删除的。

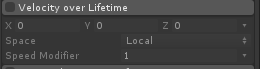


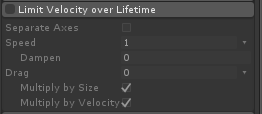
1. 不同点
2. ue4相对于unity的新特性
3. Unity特效转ue4特效



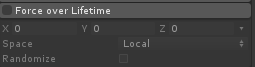








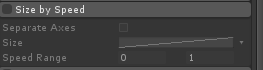




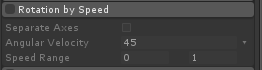




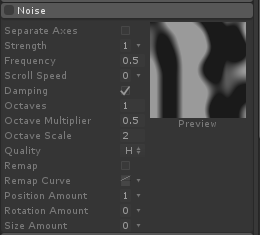


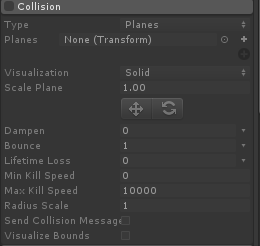


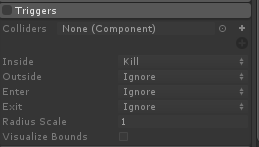




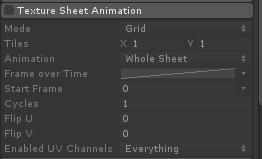


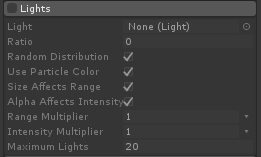


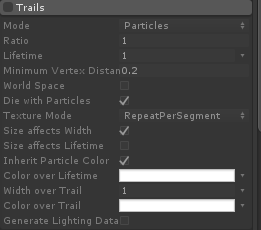


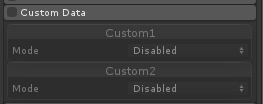


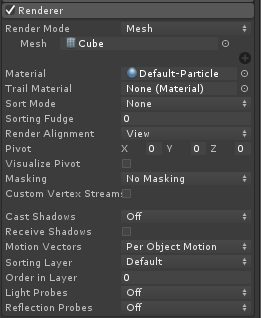












Duration:粒子发射器，发射粒子的时间。单位为S（秒）。

　　　　Looping:是否开启。如果开启了循环，Duration值只要大于最小值0.10即可。

　　　　Prewarm:是否开启预热。只有在开启的循环的时候，预热才有效果，粒子量相似发射了一个粒子周期。

　　　　Start Delay:预热延迟。游戏开始多长时间后，预热开启。单位为S。

　　　　Start Lifetime:粒子从发射器出来，到消失的时间。单位为S。

　　　　Start Speed:粒子发射的速度。

　　　　Start Size:粒子大小。

　　　　Start Rotation:粒子的旋转角度。

　　　　Start Color:粒子的颜色。

　　　　Gravity Multiplier:设置重力对粒子的影响，数值越大影响越大。

　　　　Inherit Velocity:速度继承。当粒子系统是运动的时候，粒子的速度会继承运动的速度。但粒子坐标系必须在世界坐标系。

　　　　Simulation Space:坐标系是本身还是世界坐标系。

　　　　Play On Awake:在游戏开始播放，但不影响Start Delay效果。

　　　　Max Particles:粒子释放的最大数量，当达到最大数量时，停止释放粒子，当有粒子消失时继续释放。

发射模块(Emission):

控制粒子的发射时的速率，在粒子系统存续期间可以在某个时间生成大堆粒子（模拟爆炸）。

速率（Rate）：每秒的粒子发射的数量 （小箭头可选择“每秒（Time）”或“每米（Distance）”）。

突发（Bursts）：在粒子系统生存期间增加爆发

爆炸的时间 | 数量（Time | Particles）：指定时间（在生存期内，以秒(Time)为单位时），将发射指定数量的粒子。用" "或"-"调节爆发数量。

发射器形状模块(Shape):

回顶部

发射器形状内部随机位置生成，并能提供初始的力，该力的方向将延表面法线或随机方向发射。

球体【Sphere】

半径（Radius）：球体的半径。

从外壳发射（Emit from Shell）：从球体外壳发射。

随机方向（Random Direction）：随机方向或是沿表面法线。

半球【Hemisphere】

半径（Radius）：半椭圆的半径。

从外壳发射（Emit from Shell）：从半椭圆外壳发射。

随机方向（Random Direction）：随机方向或是沿表面法线。

锥体【Cone】喇叭~

角度（Angle）：圆锥的角度（喇叭口）。如果是0，粒子将延一个方向发射（直筒）。

半径（Radius ）：发射口半径。

立方体【Box】

Box X：X轴的缩放值

Box Y：Y轴的缩放值

Box Z：Z轴的缩放值

随机方向（Random Direction）：粒子将延一个随机方向发射（取消：延Z轴发射）。

网格【Mesh】

类型（Type）：粒子将从顶点（Vertex）、边（Edge）、或三角面（Triangle）发射。

网格（Mesh）：选择一个面作为发射面。

随机方向（Random Direction）：粒子发射将随机方向或是沿表面法线。

存活时间的速度模块(Velocity Over Lifetime):

回顶部

粒子的直接动画路径。（通常用于复杂物理粒子，不过是简单的视觉行为和物理世界的小互动，如与飘荡的烟雾和气温降低。）

XYZ：使用常量曲线或在曲线中随机去控制粒子的运动。

Space：局部/世界 速度值在局部还是世界坐标系。

存活期间的限制速度模块(Limit Velocity Over Lifetime):

回顶部

-基本上被用于模拟的拖动。如果有了确定的阀值，将抑制或固定速率。可以通过坐标轴或向量调整。

分离轴（Separate Axis）：用于每个坐标轴控制。

----速度（Speed）：用常量或曲线指定来限制所有方向轴的速度。（未选中）

----XYZ：用不同的轴分别控制。见最大最小曲线。（选中）

阻尼（Dampen）：（0-1）的值确定多少过度的速度将被减弱。（值为0.5，将以50%的速率降低速度）

存活期间的受力模块(Force Over Lifetime):

回顶部

XYZ：使用常量或随机曲线来控制作用于粒子上面的力。

Space:Local自己的坐标系，World世界的坐标系。

随机（Randomize）：每帧作用在粒子上面的力都是随机的。(两组xyz时可选择，随机范围是这两组xyz之间的差值。)

存活时间的颜色模块(Emission):

回顶部

控制每个粒子存活期间的颜色（与初始颜色叠加）。粒子存活时间越短变化越快。

两种颜色随机比例互相叠加。

颜色速度模块(Color By Speed):

回顶部

使粒子颜色根据其速度动画化。为颜色在1个特定范围内重新指定速度。

颜色（Color）：用于指定的颜色。使用渐变色来指定各种颜色。

颜色缩放（Color Scale）：使用颜色缩放可以方便的调节纯色和渐变色。

速度范围（Speed Range）：min和max值用来定义颜色速度范围。

存活时间的大小模块(Size Over Lifetime):

回顶部

大小（Size）：控制每个粒子在其存活期间内的大小。曲线，随机双曲线或随机范围数值。

存活时间的大小速度模块(Size By Speed):

回顶部

大小（Size）：大小用于指定速度。用曲线表示各种大小。

速度范围（Speed Range）：min和max值用来定义大小速度范围。

存活期间的旋转速度模块(Rotation Over Lifetime):

回顶部

以度为单位指定值。

旋转速度（Rotational Speed）：控制每个粒子在其存活期间内的旋转速度。使用常量，曲线，2曲线随机。

旋转速度模块(Rotation By Speed):

回顶部

旋转速度（Rotational Speed）：用来重新测量粒子的速度。使用曲线表示各种速度。

速度范围（Speed Range）：为min和max值用来定义旋转速度范围。

碰撞模块(Collision Module):

回顶部

为粒子系统建立碰撞。现在只有平面碰撞被支持，这个将很有效率的做简单探测。

平面（Planes）：Planes被定义为指定引用。可以动画化。如果多个面被使用，Y轴作为平面的法线。

阻尼（Dampen）：0-1 在碰撞后变慢。

反弹（Bounce）：0-1 当粒子碰撞后的反弹力度。

生命减弱（Lifetime Loss）：(0-1) 每次碰撞胜铭减弱的比例。0，碰撞后粒子正常死亡。1，碰撞后粒子立即死亡。

可视化（Visualization）：可视化平面：网格还是实体

-----网格（Grid）：在场景渲染为辅助线框。

----实体（Solid）：在场景渲染为平面。

缩放平面（Scale Plane）：重新缩放平面

子粒子发射模块(Sub Emitter):

回顶部

可以生成其他粒子系统，用下列的粒子事件：出生、死亡、碰撞。

出生（Birth）：在每个粒子出生的时候生成其他粒子系统

死亡（Death）：在每个粒子死亡的时候生成其他粒子系统

碰撞（Collision）：在每个粒子碰撞的时候生成其他粒子系统。重要的 碰撞需要建立碰撞模块。见碰撞模块

纹理层动画模块(Texture Sheet Animation):

回顶部

在粒子存活期间动画化UV坐标。动画每帧可以显示在表格或1个表格的每行，这样将动画分开。每帧可以用曲线动画或者在2个曲线取随机。速度被定义为"循环"、

注意：用于动画的纹理是在渲染器模块中材质找到使用的。

平铺（Tiles）：定义纹理的平铺

动画（Animation）：指定动画类型：整个表格或是单行。

----整个表（Whole Sheet）：为UV动画使用整个表格。

-------时间帧（Frame over Time）：在整个表格上控制UV动画。使用常量，曲线，2曲线随机。

----单行（Single Row）：为UV动画使用表格单独一行。

-------随机行（Random Row）：如果选择第一行随机，不选择得指定行号（第一行是0）

-------时间帧（Frame over Time）：在1个特定行控制每个粒子的UV动画。使用常量，曲线，2曲线随机。

周期（Cycles）：指定动画速度。

渲染器模块(Renderer):

回顶部

渲染模块显示粒子系统渲染组件的属性。注意：即使一个游戏物体有渲染粒子系统组件，当此模块被删除/添加后，它的属性也只能显示在这里。这个实际上是粒子系统渲染组件的添加和删除。

渲染模式（Render Mode）：选择下列粒子渲染模式之一

----广告牌（Billboard）：让粒子永远面对摄像机。

----拉伸广告牌（Stretched Billboard）：粒子将通过下面属性伸缩。

-------摄像机缩放（Camera Scale）：决定摄像机的速度对粒子伸缩的影响程度。

-------速度缩放（Speed Scale）：通过比较速度来决定粒子的长度。

-------长度缩放（Length Scale）：通过比较宽度来决定粒子的长度。

----水平广告牌（Horizontal Billboard）：让粒子延Y轴对齐，面朝Y轴方向。

----垂直广告牌（Vertical Billboard）：当面对摄像机时，粒子延XZ轴对齐。

----网格（Mesh）：粒子被渲染时使用mesh而不是quad。

-------网格（Mesh）：渲染粒子所用的网格引用。

材质（Material）：材质。

排序模式（Sort Mode）：绘画顺序可通过具体，生成早优先和生成晚优先。

排序校正（Sorting Fudge）：使用这个将影响绘画顺序。粒子系统带有更低sorting fudge值，更有可能被最后绘制，从而显示在透明物体和其他粒子系统的前面 。

投射阴影（Cast Shadows）：粒子能否投影？这是由材质决定的。

接受阴影（Receive Shadows）：粒子能否接受阴影？这是由材质决定的。

最大粒子大小（Max Particle Size）：设置最大粒子大小，相对于视窗大小。有效值为0-1。