

让我们继续学习吧。

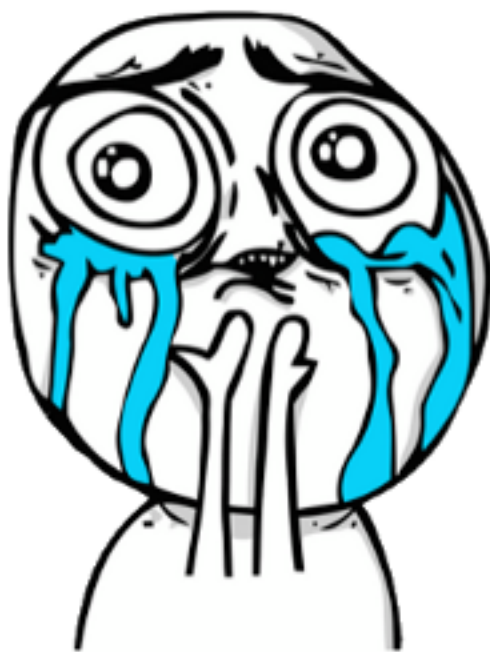
关于蓝图系统

现在我们的👉已经有模有样了，但是如果把它放置在一个上旋转会更有意思。而要创建一个转台，我们需要用到蓝图。

用最简单的话来首，蓝图代表着一个”事物“。我们可以用蓝图来创建游戏对象的定制化行为。游戏对象可以是实物（比如转台），也可以是某种抽象的概念，比如健康系统。

想要设计一辆会飞的汽车吗？用蓝图吧。想要设计一个飞猪？用蓝图吧。想要一个💣的🐶？用蓝图吧。

和材质类似，蓝图使用的也是基于节点的系统。这就意味着使用蓝图系统开发要做的就是创建节点，然后在节点之间创建关联，不需要写一行代码~！



I CAN DO ALL OF THAT
WITH BLUEPRINTS?

注意：如果你是个代码狂魔，那么当然可以选择使用C++。

而且在虚幻4的4.18版本中，使用蓝图创建的游戏是可以转化为C++代码的。

尽管蓝图用起来很贴心很方便，但是在执行效率上当然是不如C++代码的。所以如果我们需要用到某种复杂的算法，那么就必须考虑使用C++。

但即便你是个代码狂魔，那么也仍然可以考虑使用蓝图系统，大不了两个一起用吗。

使用蓝图系统有以下的好处：

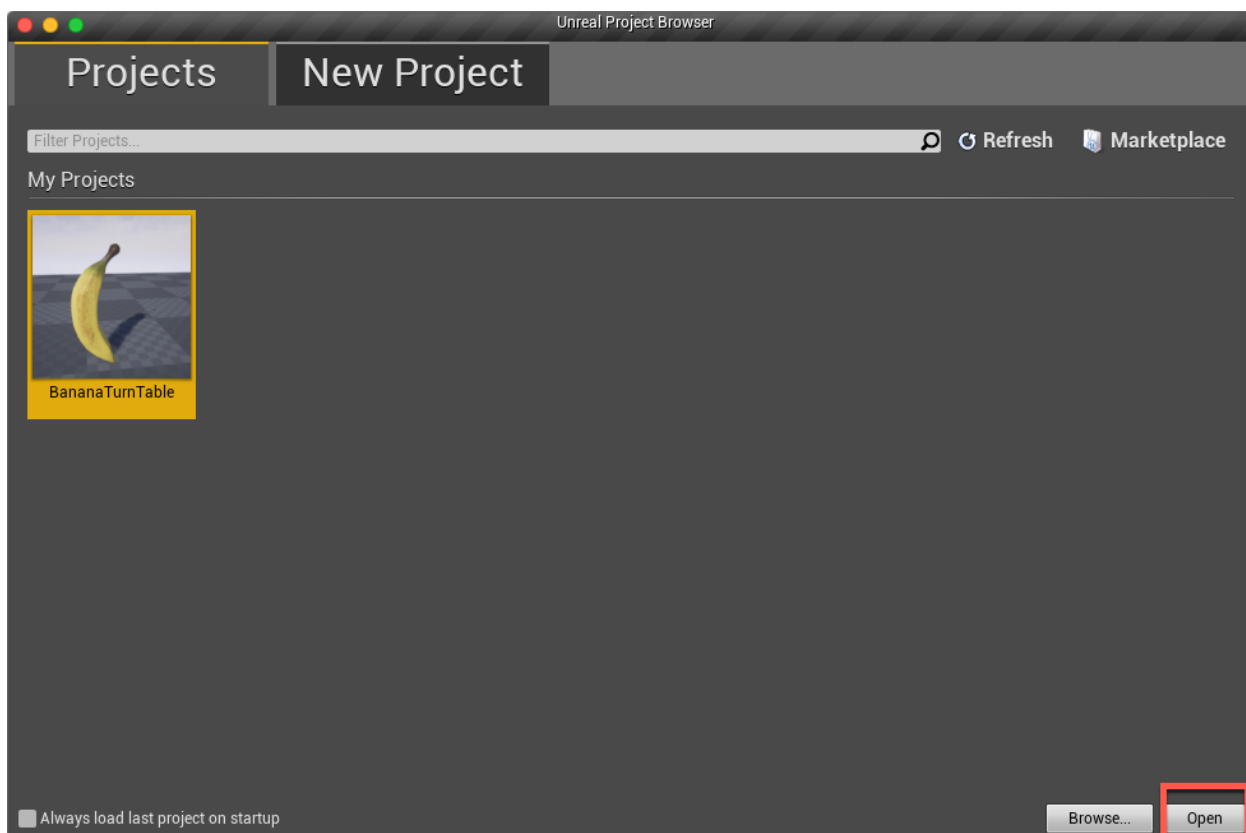
- (1) 通常来说，使用蓝图开发要比C++快。
- (2) 蓝图更便于项目的组织。我们可以把节点很方便的分隔成不同的区域，比如功能和graph。
- (3) 如果你是个3D美术设计人员，或者需要跟3D美术设计人员密切合作，那么使用蓝图系统会更便于修改项目。

通常来说，建议大家使用蓝图系统来创建对象。当我们需要用到比较复杂的算法，或者需要提升运行效能时，可以将其转换为C++代码。

打开项目

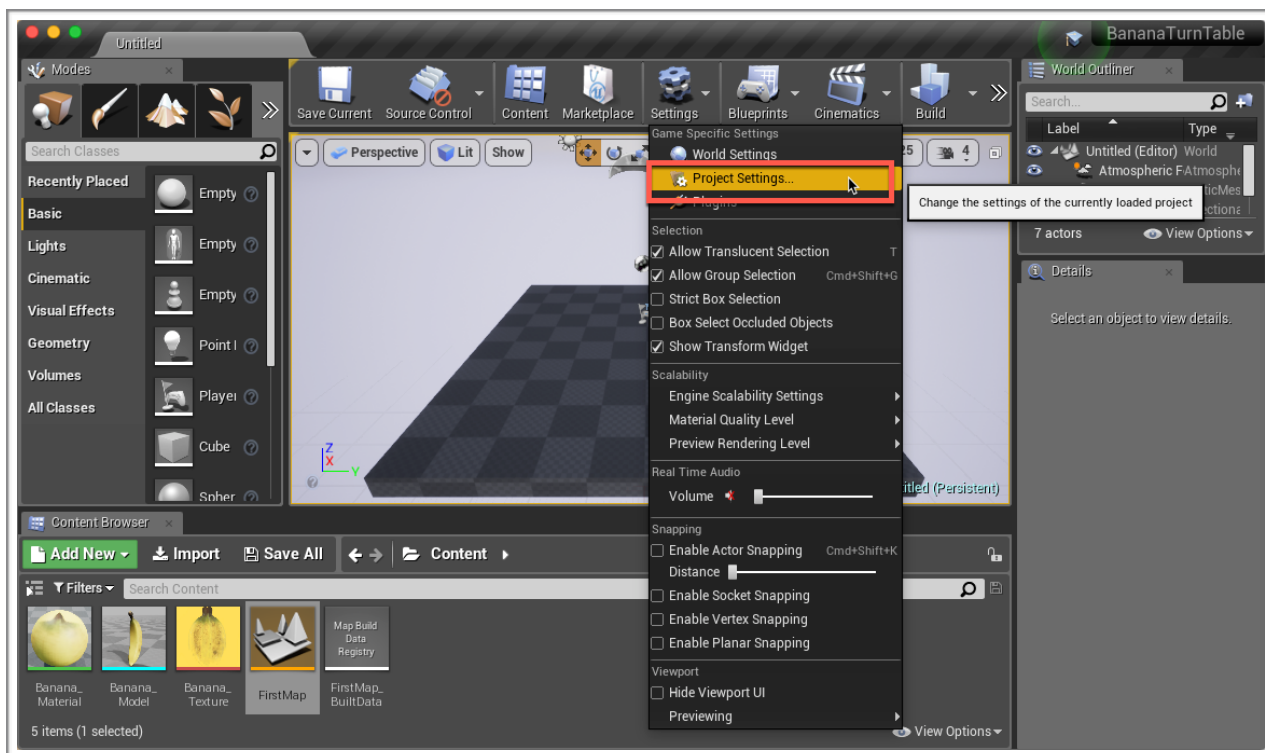
在上一课的内容中我们已经保存了项目。因此，当需要重新打开项目时，首先还是打开Epic Games Launcher，然后点击Launch按钮打开引擎。

等待加载完成后，会看到类似下面的界面：



点击Open即可打开项目。此时你会看到我们之前的游戏场景不见了，只有在Content Browser中双击打开FirstMap才能进入游戏场景。

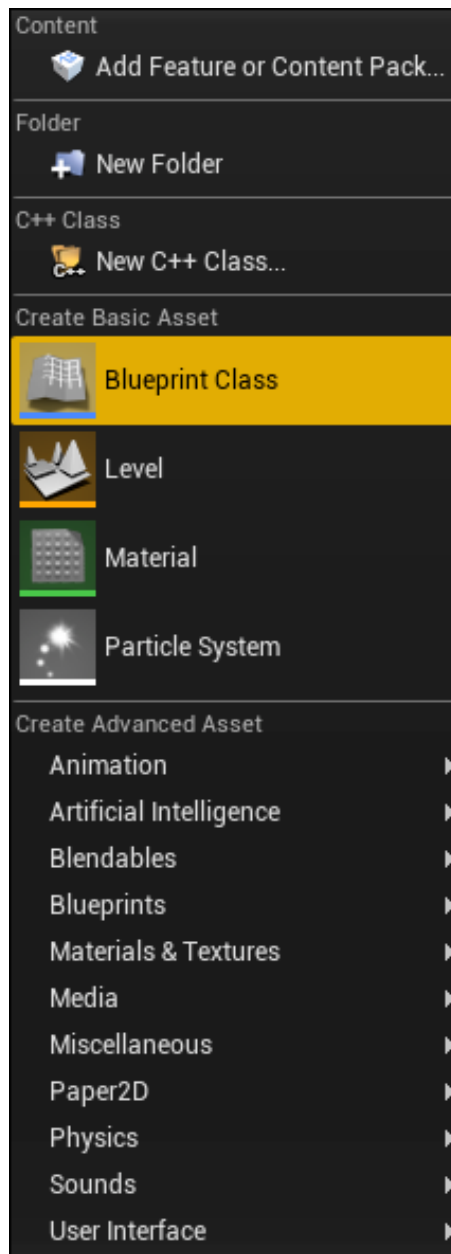
为了方便起见，我们可以对项目做一点设置。点击工具栏上的Setting，选择Project Settings，



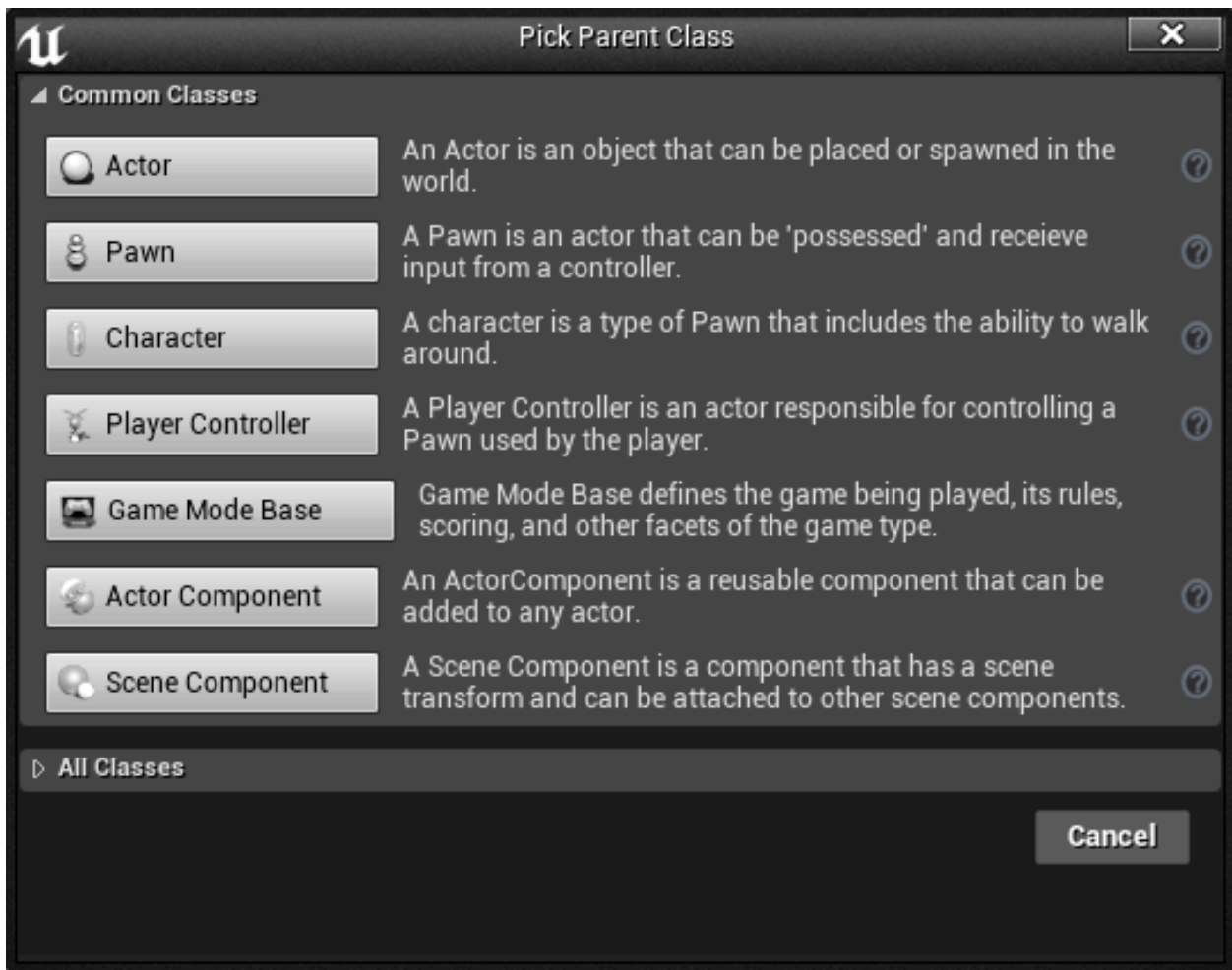
在打开的新界面中，从左侧选择Maps&Modes，然后在Default Map处将Editor Startup Map和游戏 Default Map右边的属性选择为FirstMap，也就是我们在上一课中所保存的游戏场景。关闭虚幻编辑器，从Epic Games Launcher中重新打开引擎，然后打开项目，就会自动打开之前所创建的游戏场景了。

创建蓝图

在虚幻4引擎中切换到Content Browser，然后点击Add New，从列表中选择Blueprint Class。



此时会弹出一个窗口，提示让我们选择蓝图的父类。关于父类和继承的概念，我们会在后面详细介绍，这里只需要知道，当我们选择某一种父类后，蓝图系统将继承该父类的所有变量、函数和组件。在继续之前，建议大家仔细阅读以下每个父类的描述。



注意，对于Actor,Pawn和Character来说，Character继承自Pawn,而Pawn又继承自Actor。此外，Player Controller也继承自Actor。

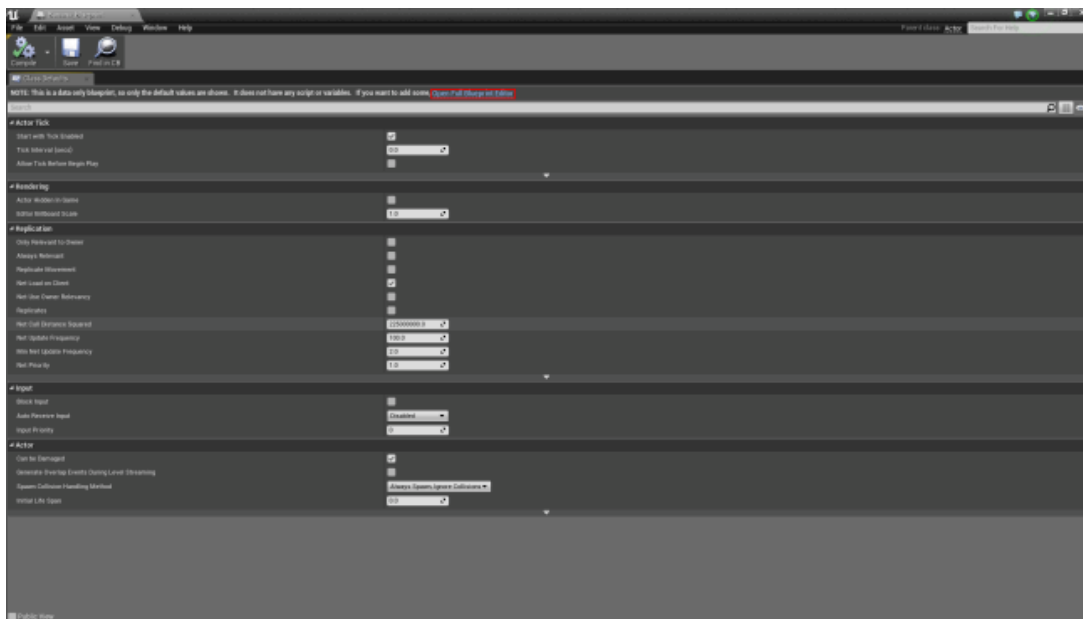
简单来说，Pawn，Character和Player Controller也是Actor。

考虑到我们这里要添加的转台会在游戏场景中的固定位置，Actor类显然最合适。因此我们在这里选择Actor，并将新的蓝图类命名为Banana_Blueprint。

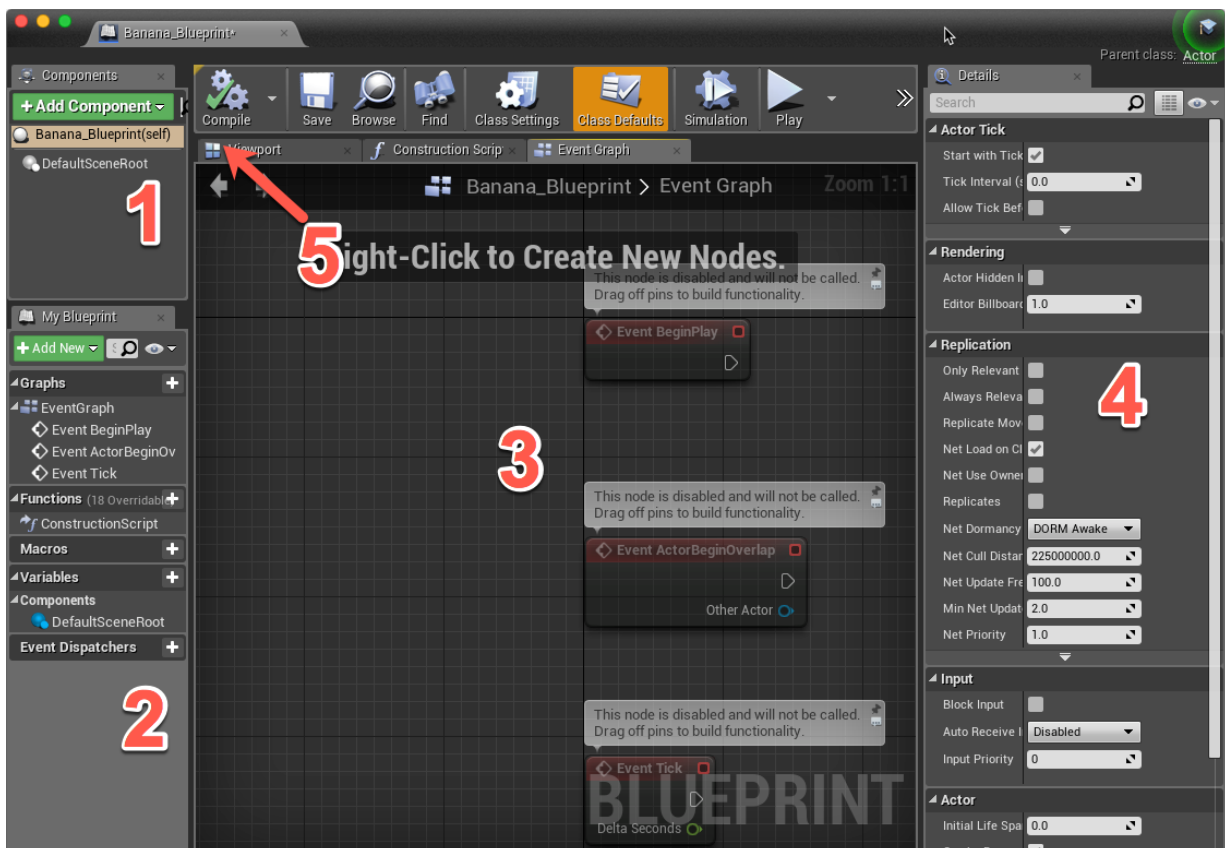


最后，双击Banaba_Blueprint打开这个蓝图文件。

如果你看到的是类似下面的界面，那么请点击Open Full Blueprint Editor，



完全状态下的蓝图编辑器如下图所示



其中各个区域的作用如下：

1.Components区：

包含了当前组件的列表。

虚幻4中的组件和Unity3d中的组件有一定的类似之处，但又有一定的区别。

2.My Blueprint区：

这个部分用于惯例蓝图中的graph，函数和变量。我们可以按住鼠标右键不放，拖动鼠标以平移，也可以通过鼠标滚轮来缩放视图。

3.Graph区：

该部分是蓝图系统的精华所在。这里会显示蓝图中所有的节点和逻辑机制。

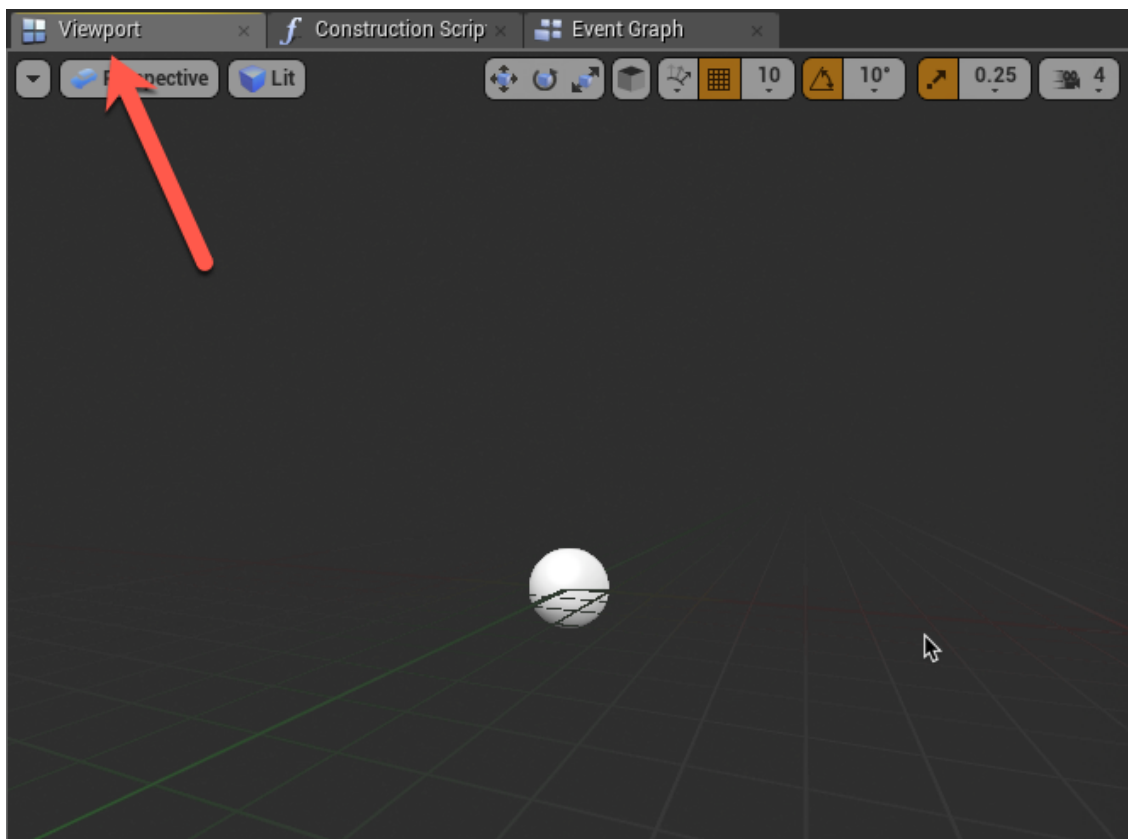
当然，要注意此时我们选择的是Event Graph

4.Details区：

该部分用于显示所选择项目的具体属性

5.Viewport区：

当我们在选项卡上选择的是Viewport，那么就可以看到带有视觉元素的所有组件。我们可以使用 and 主编辑器类似的操作在移动和查看视觉元素。



创建转台

想要创建转台,我们需要两个部分：底座和display。我们可以使用组件来创建这两个部分。

什么是组件？

如果把蓝图系统比喻成一辆车，那么组件就是组成车的各种零部件，比如车门、车轮和引擎都是组件的例子。

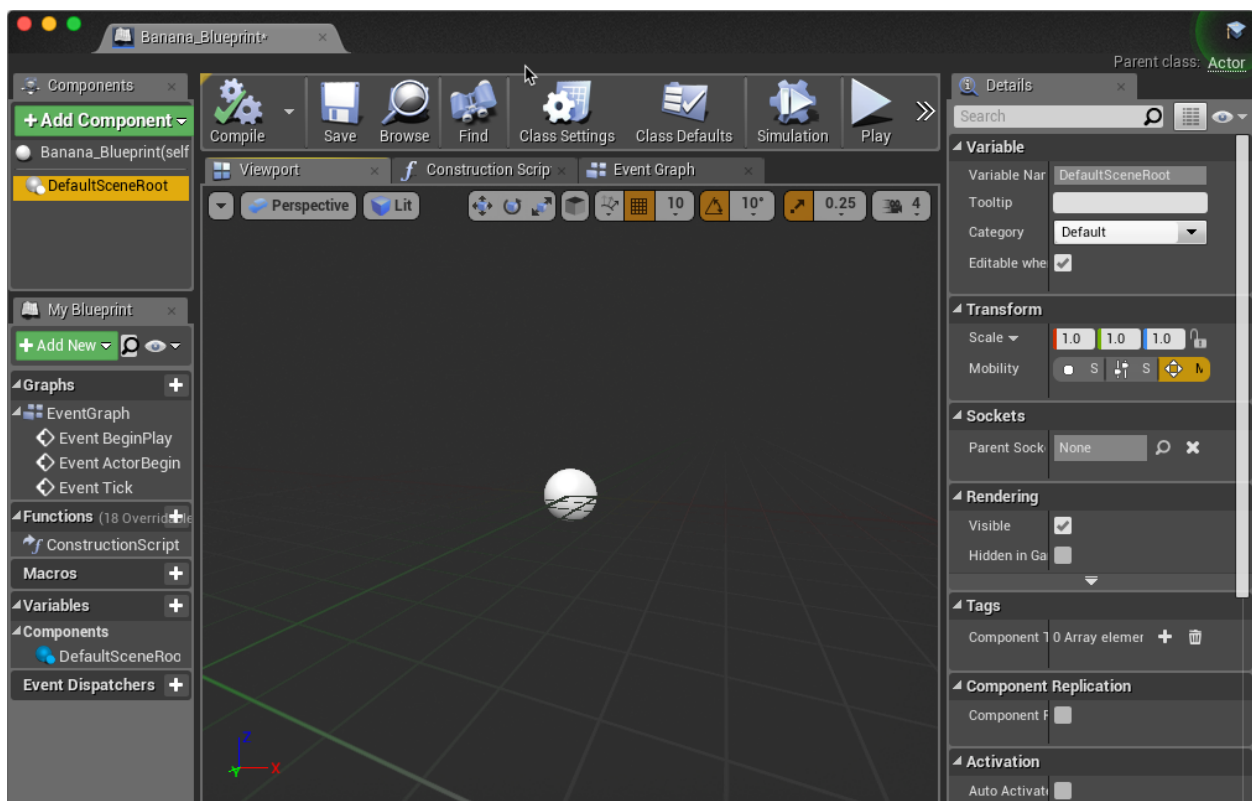
不过组件并不局限于实际的物体。

举例而言，为了让汽车移动，我们需要添加一个移动组件。为了让汽车飞行，我们又需要添加一个飞行组件。

学过Unity3d的朋友可能会对其中的组件系统有一定的了解。虽然和虚幻4中的组件系统有一定的差异，但大致的作用是类似的。

添加组件

如果我们需要查看组件的可视化元素，那么就需要切换到Viewport视图。点击Viewport选项卡以切换到该视图。



注意：DefaultSceneRoot组件在游戏运行的时候并不会显示，它只会显示在编辑器中。

转台包含了两个组件：

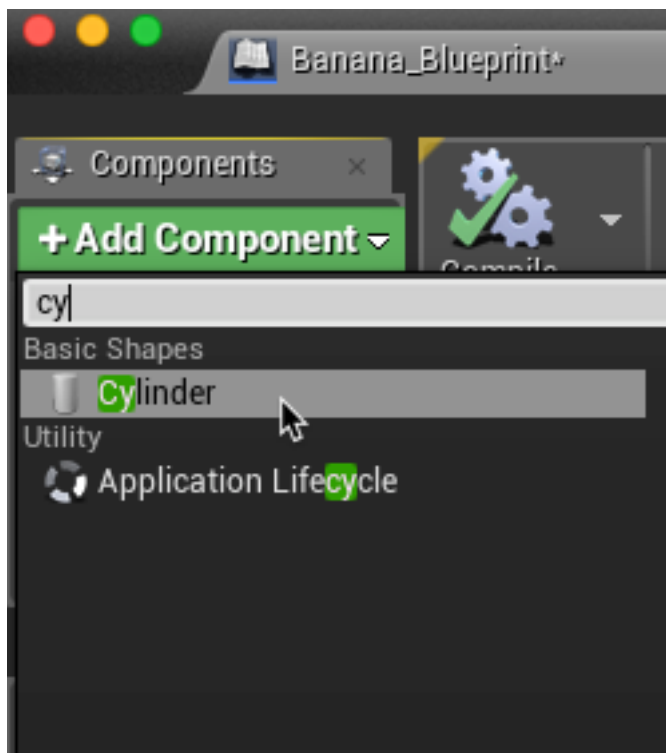
(1) Cylinder:

一个简单的白色圆柱体，它将构成底座。

(2) Static Mesh:

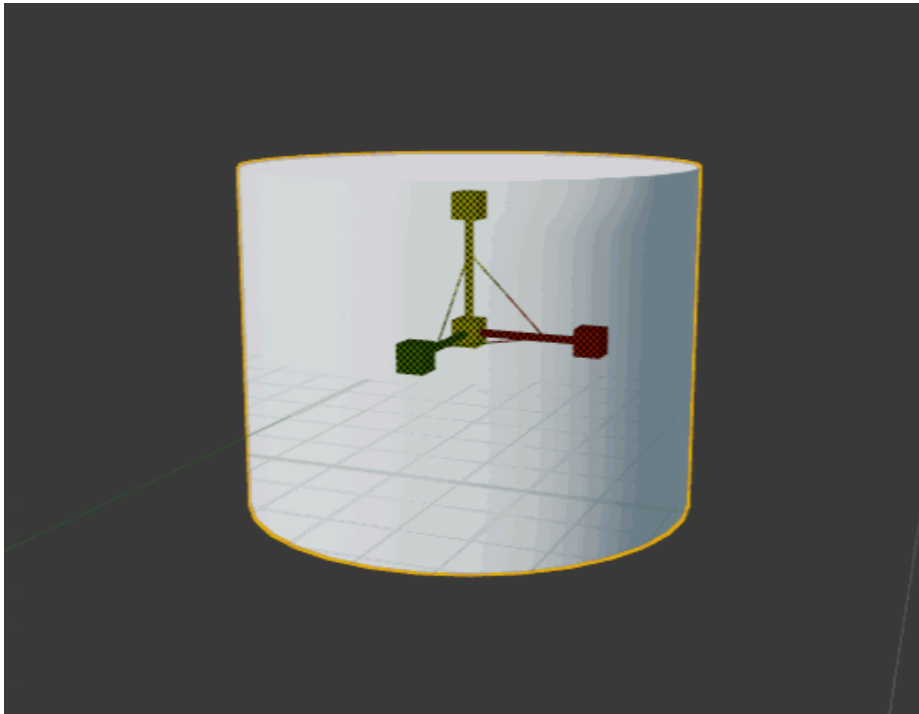
该组件将显示香蕉的模型

为了添加底座，让我们在蓝图编辑器中找到Components面板，点击Add Component，然后选择Cylinder。



此时在Viewport视图中可以看到多了一个圆柱体。

当然现在它的高度有点超出预期，我们需要把它变得扁平一点。按下键盘上的R键，然后把圆柱体的高度降低，具体的尺寸大小不是那么的重要，因为我们后续随时都可以继续调整。

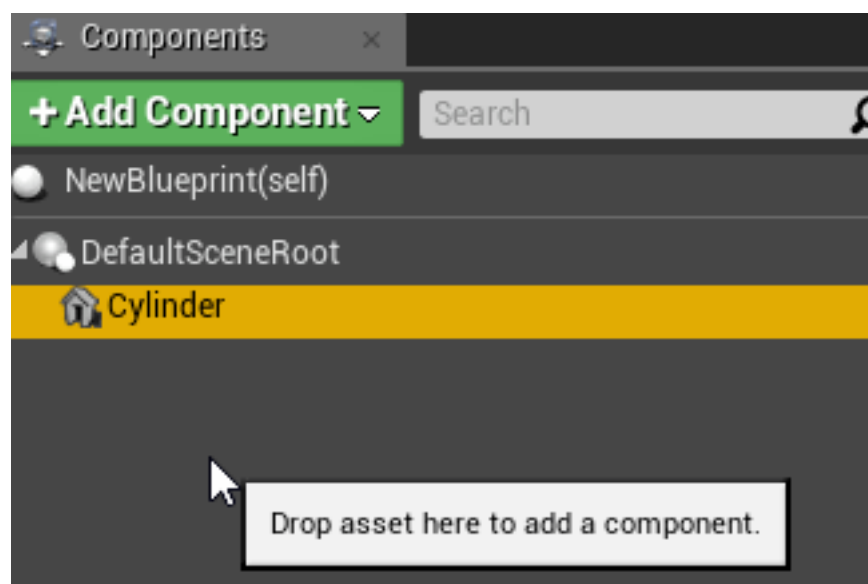


现在该是添加模型的时候了。

返回Components面板，左键单击空白区，以取消选中Cylinder组件。这样确保我们所添加的下一个组件不会关联到Cylinder组件上。

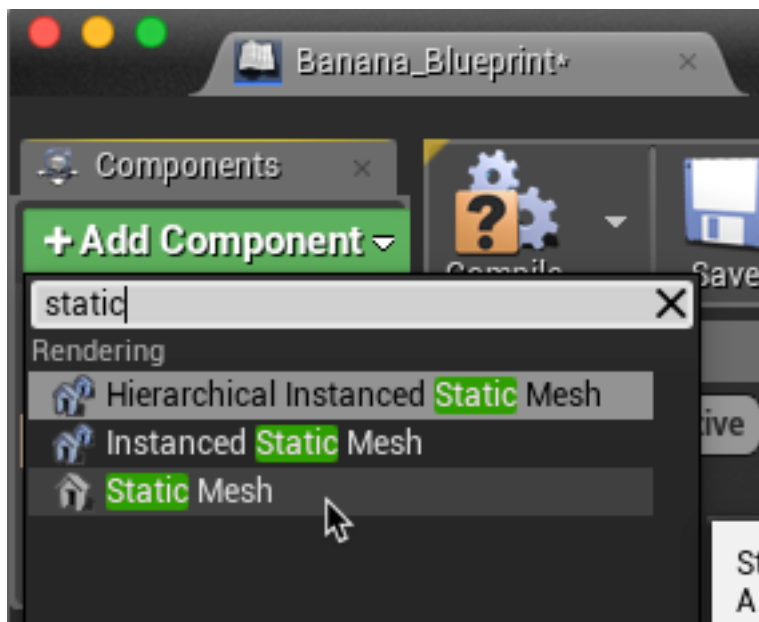
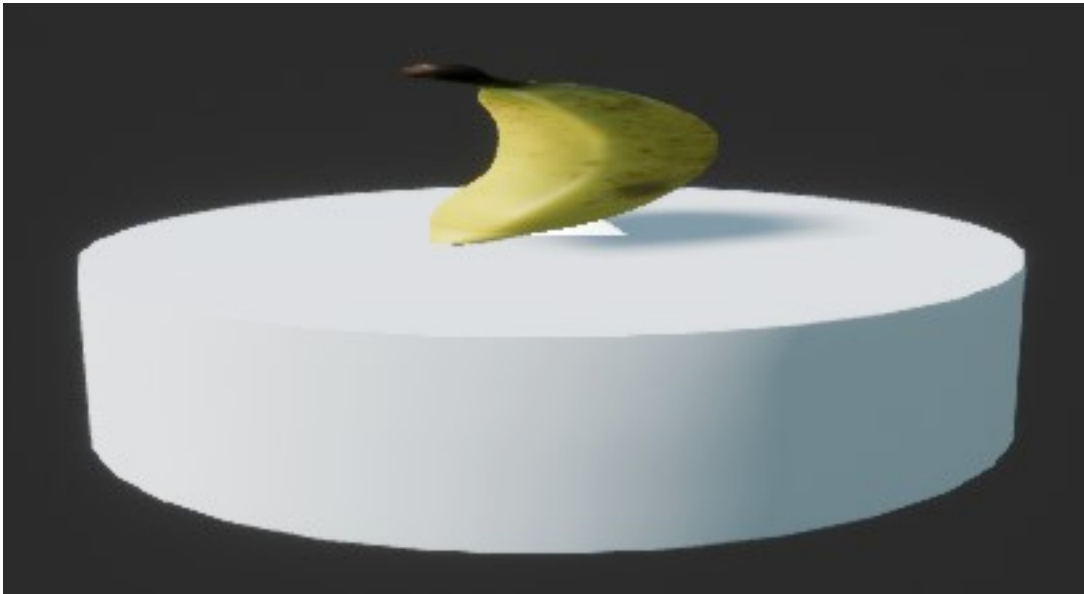
注意：

如果你不这么做，那么所添加的下一个组件就会关联到Cylinder组件上。这就意味着所添加的新组件将继承Cylinder组件的比例。因为我们曾经调整过圆柱体的比例，那么新添加的组件比例也会被调整。

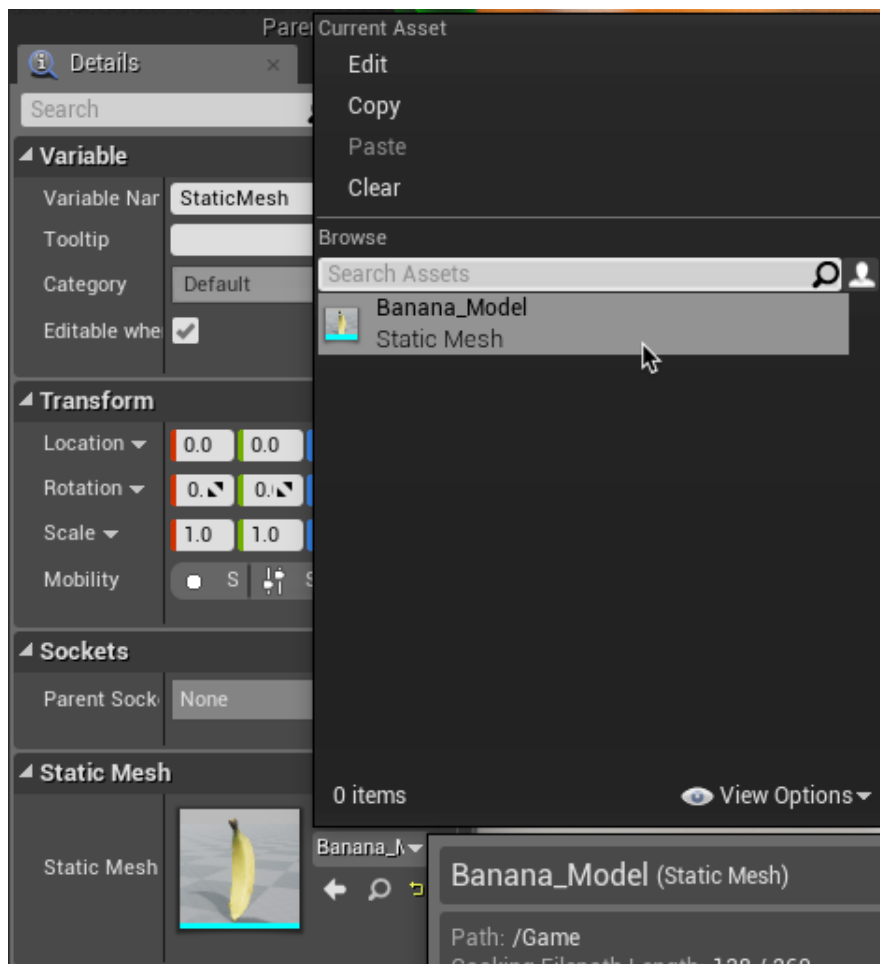


接下来，点击Add Component，从列表中选择Static Mesh。

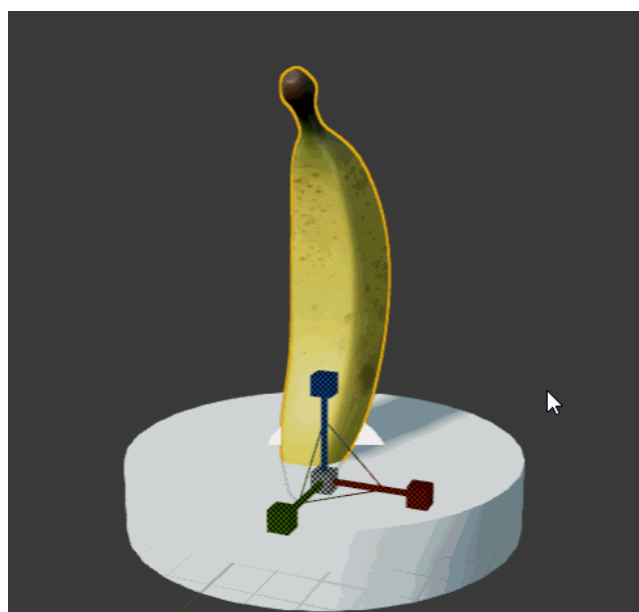
为了显示香蕉，需要选中Static Mesh组件，然后在Details面板中点击Static Mesh右侧的下拉列表，然后选择Banana_Model。



接下来需要把香蕉移动到合适的位置。按下键盘上的W键，然后把香蕉向上移动到合适的位置。



点击蓝图编辑器工具栏上的Compile按钮和Save按钮，以保存所做的修改。
好了，我们现在已经在蓝图中添加了所需的视觉元素。
在下一课的内容中，我们将学习如何让转台旋转。



联系方式:

头条号: 寒酒仙

微信公众号: vrlife

个人微信号: iseedo

教程讨论: <http://icode.fun>