

---

## 1. 实验名称及目的

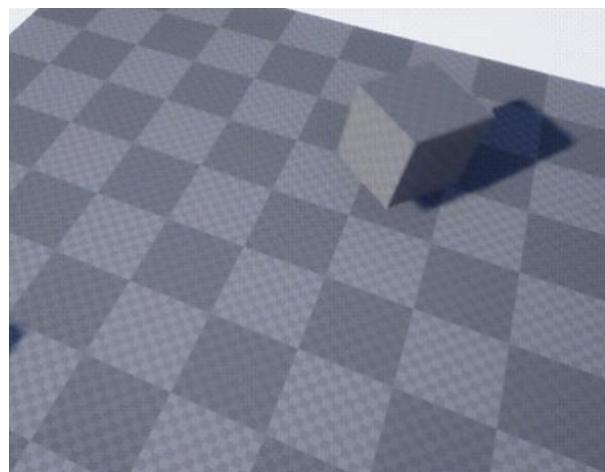
**UE 安装与蓝图编程实验：**下载并安装 UE，使用蓝图控制物体进行简单运动

## 2. 实验原理

虚幻引擎是一款由 Epic Games 开发的强大的游戏开发引擎。它具有图形渲染、物理模拟、动画、蓝图脚本和人工智能等功能，可用于创建高品质的三维游戏、交互性应用程序和虚拟现实/增强现实体验。与 Unity 引擎相比，虚幻引擎提供了更高度可定制和可扩展的工具集，并公开了所有源代码，使开发者可以自行解决问题。虚幻引擎还使用了蓝图系统，可将业务逻辑与底层功能隔离开来，使模型设计人员与 C++ 程序开发人员互不干扰，从而保持开发架构的稳定性。

## 3. 实验效果

安装好 UE、使用蓝图控制物体进行自转的同时再绕原点进行一个圆周运动。



## 4. 文件目录

| 文件夹/文件名称 | 说明 |
|----------|----|
|          |    |

## 5. 运行环境

| 序号 | 软件要求                | 硬件要求                  |    |
|----|---------------------|-----------------------|----|
|    |                     | 名称                    | 数量 |
| 1  | Windows 10 及以上版本    | 笔记本/台式电脑 <sup>①</sup> | 1  |
| 2  | Epic Games          |                       |    |
| 3  | UnrealEngine 4.27.2 |                       |    |

①：推荐配置请见：<https://doc.rflysim.com>

## 6. 实验步骤

### Step 1: 安装 EpicLauncher

前往 UE 官网 ([虚幻引擎 | 最强大的实时 3D 创作平台 - Unreal Engine](#)) 安装 EpicGames 启动程序。



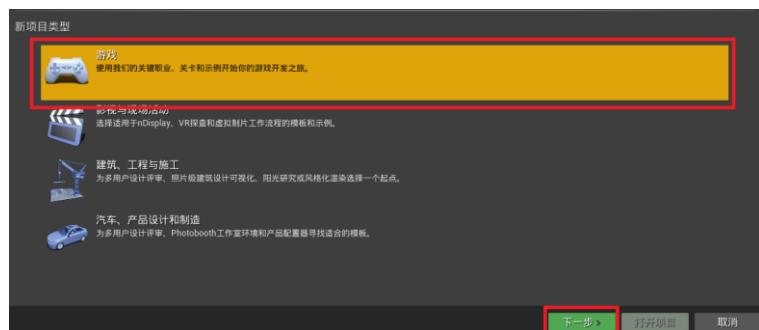
点击虚幻引擎->库，点击加号选择需要下载的引擎版本，如果是 4.xx，则是 UE4，如果是 5.xx，则是 UE5，这里就选择 4.27.2，等待它下载完成。

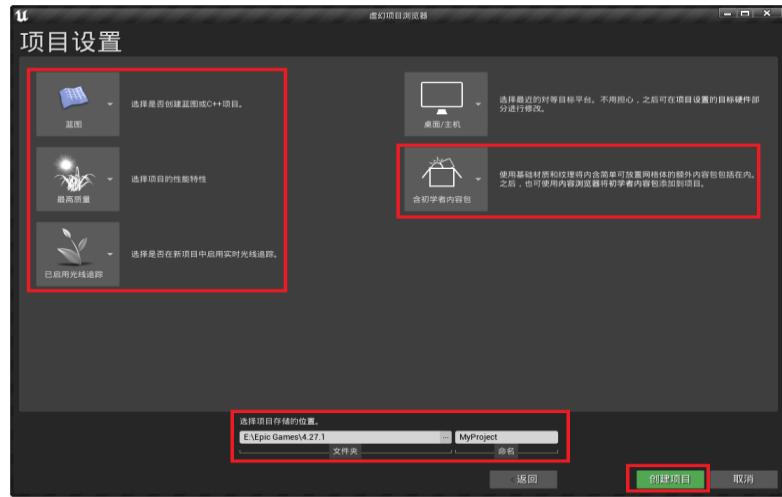
### Step 2: 创建关卡

选择创建一个项目

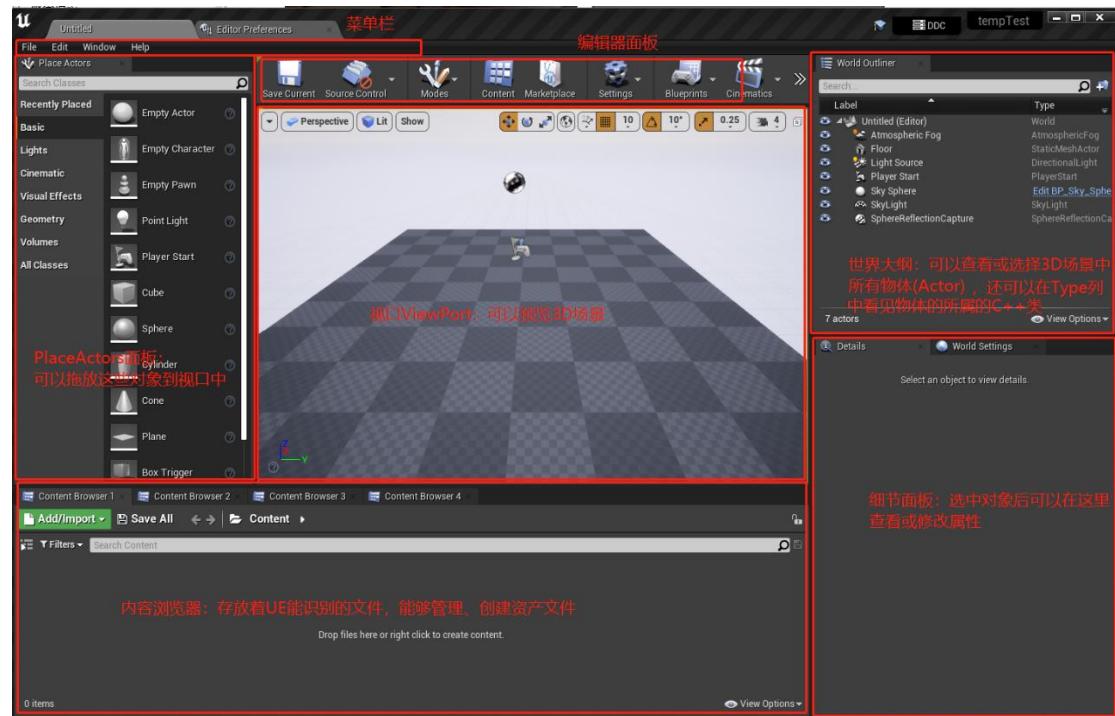


这些模板会额外提供一些基础的蓝图，选择空白模板即可



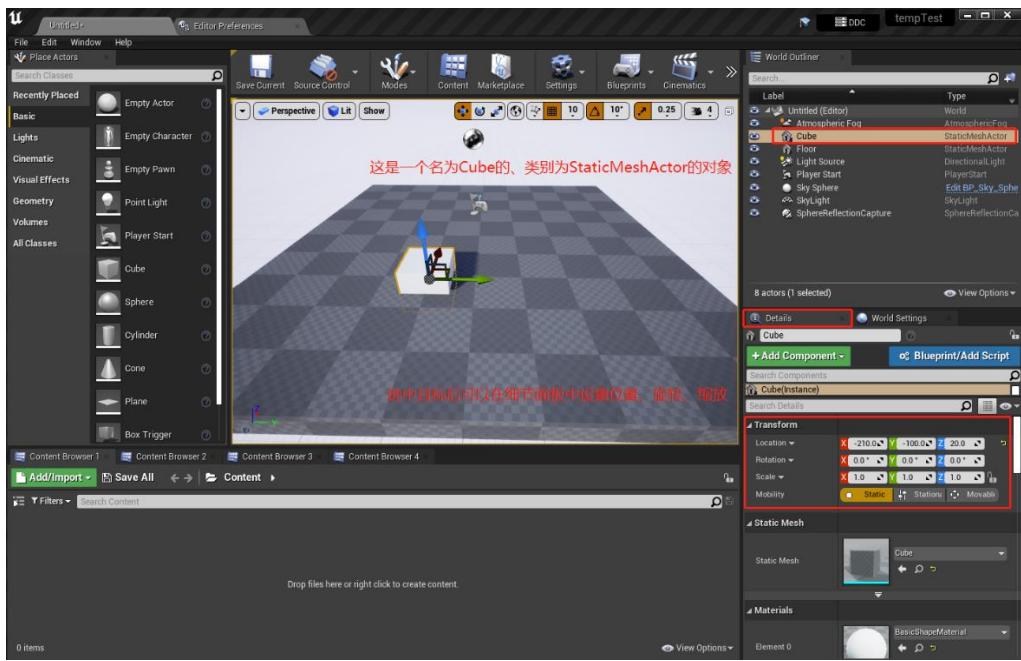


### Step 3: UE4 编辑器界面介绍



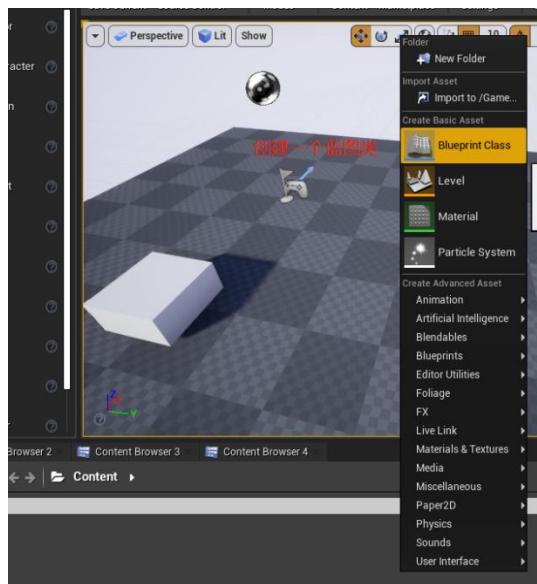
UE 的界面是可以自定义布局的，大多数窗口都可以通过拖拽它的选项卡（Tab Options）来重新设置布局。

可以在视口（ViewPort）中按住鼠标右键，使用鼠标与 WASDQE 进行移动与旋转查看 3D 场景，我们还可以从 PlaceActors 面板中拖拽一个盒体 Cube 到视口（ViewPort）中查看效果。

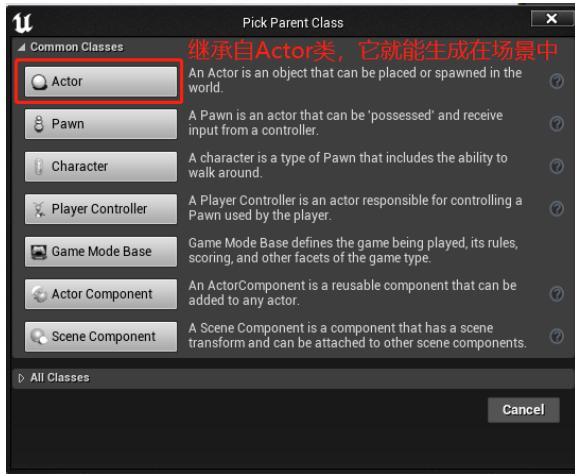


## Step 4: 创建蓝图类

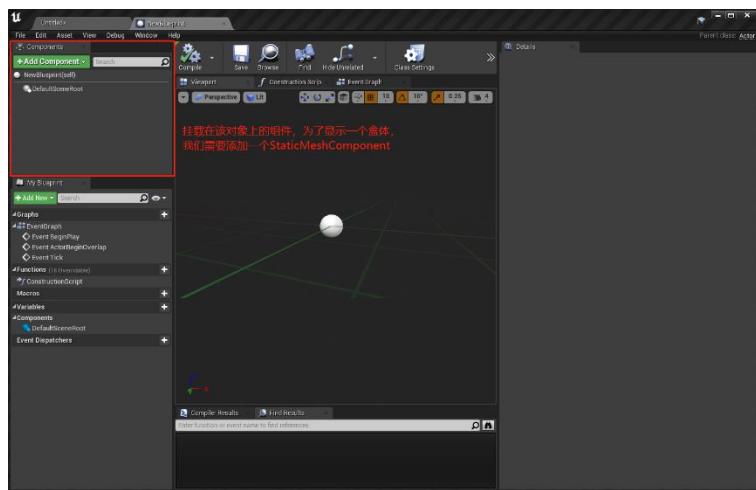
接下来简单的使用一下蓝图系统，控制一个盒体进行旋转，主要是了解一下这些物体。  
右键点击内容浏览器，创建一个蓝图类（Blueprint Class）



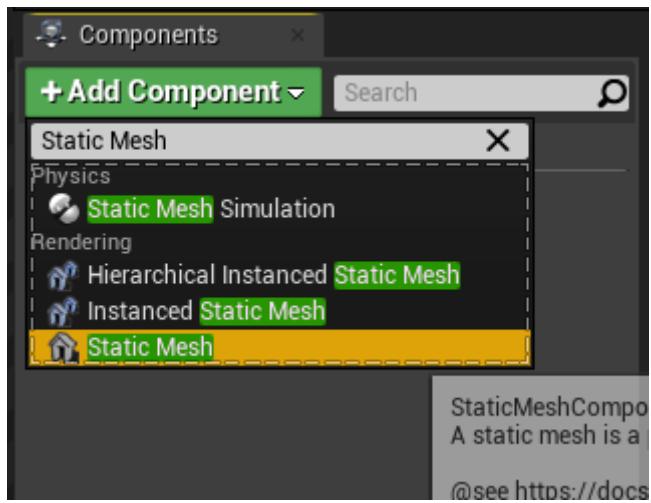
选择 Actor 类作为父类，它是场景中所有物体的基础父类



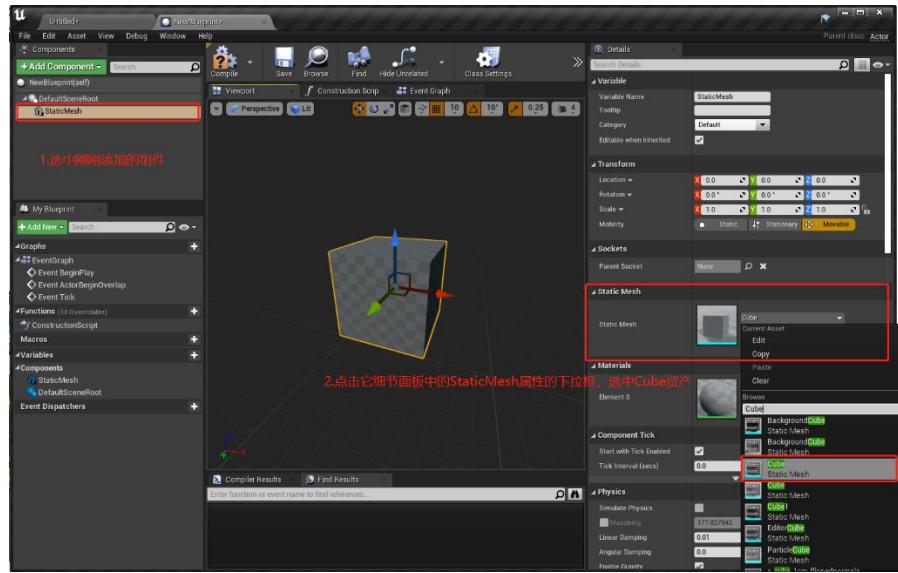
双击进入刚刚创建的蓝图资产



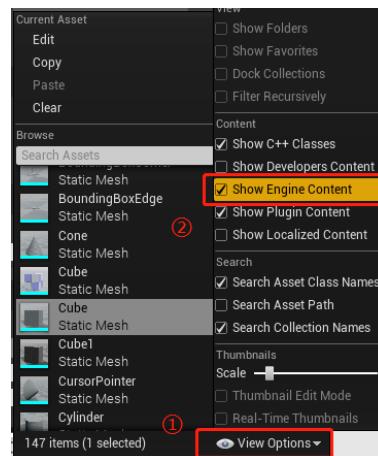
点击 AddComponent，添加一个 Static Mesh 组件



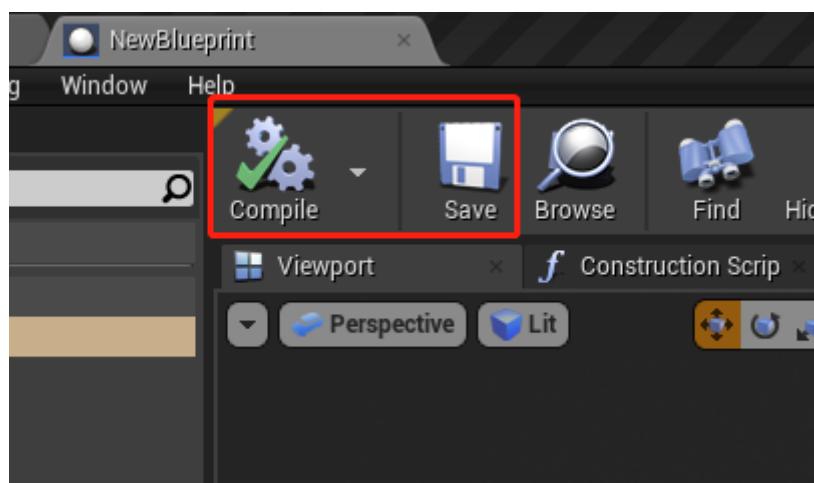
Actor 本身是无形的，许多功能都需要通过为它添加组件的形式来为它添加，这个 StaticMesh 就是“网格体组件”可以让 Actor 显示一个网格体资产。(通常是导入 3D 模型文件得到的资产，格式为.fbx 或者.obj 等通用格式)



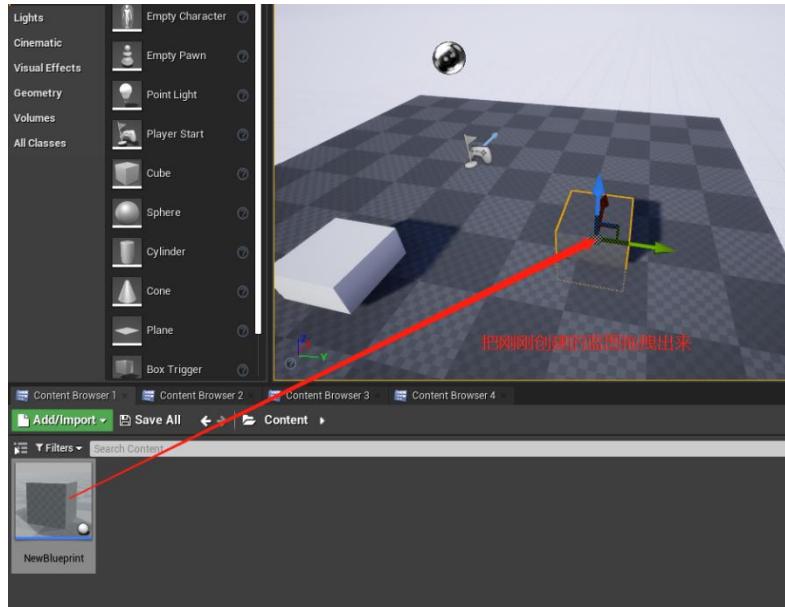
这个 Cube 网格体是 UE 引擎自带的，如果这里没有显示任何内容，则需要按照下图开启“显示引擎内容”：



点击编译与保存，则该 Actor 蓝图就保存好了

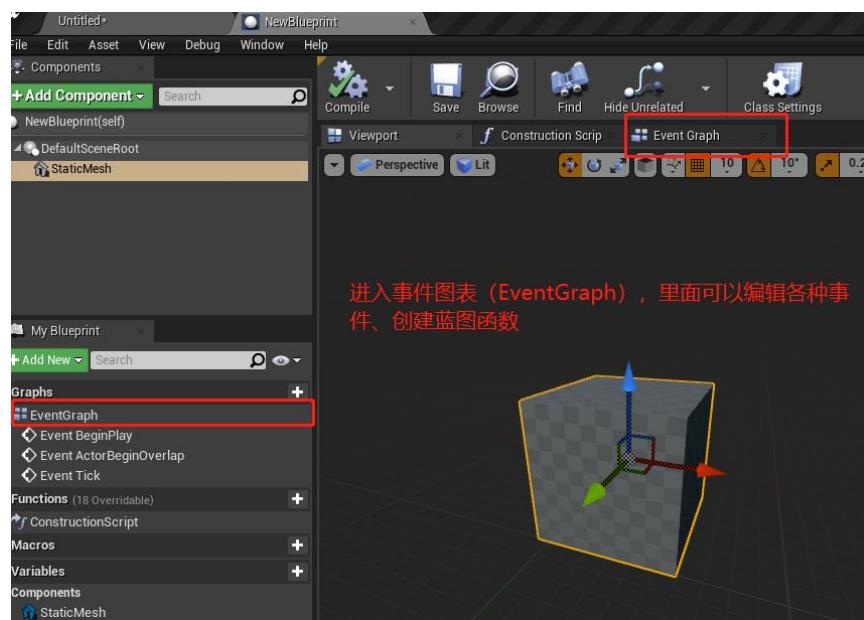


在场景中创建蓝图类的实例，将蓝图资产拖入场景中即可

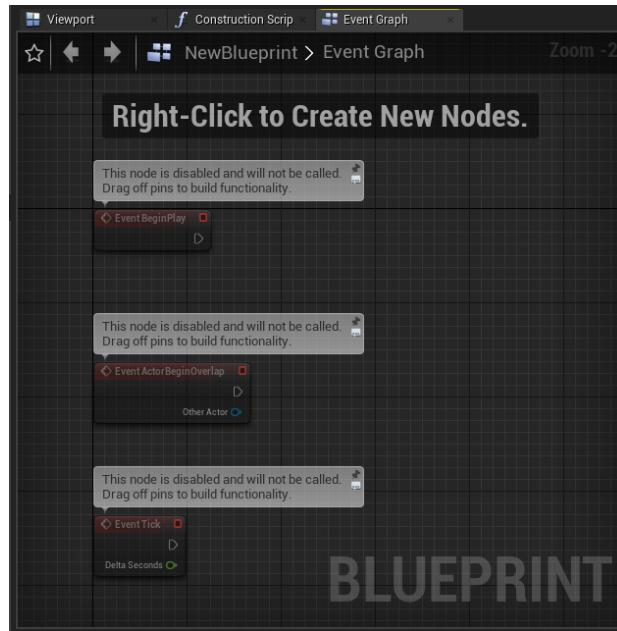


## Step 5: 编写蓝图事件

控制蓝图盒体的运动，双击蓝图，回到该蓝图类的编辑界面，进入事件图表

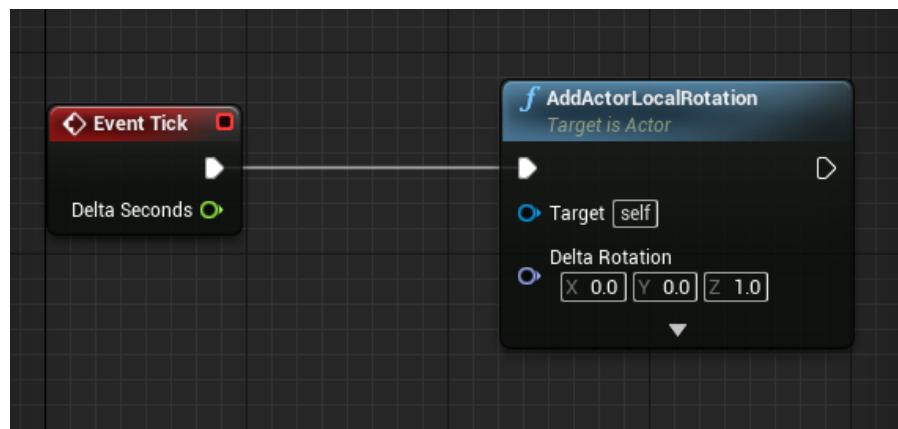


其中有一些默认的节点，这些红色的节点可以看作是回调函数，会在特定时刻被调用触发。例如其中的 BeginPlay 事件，会在游戏开始时被调用，而 ActorBeginOverlap 会在该对象与其他物体发生重叠时被调用，Tick 事件则在游戏中的每一帧都会被调用。

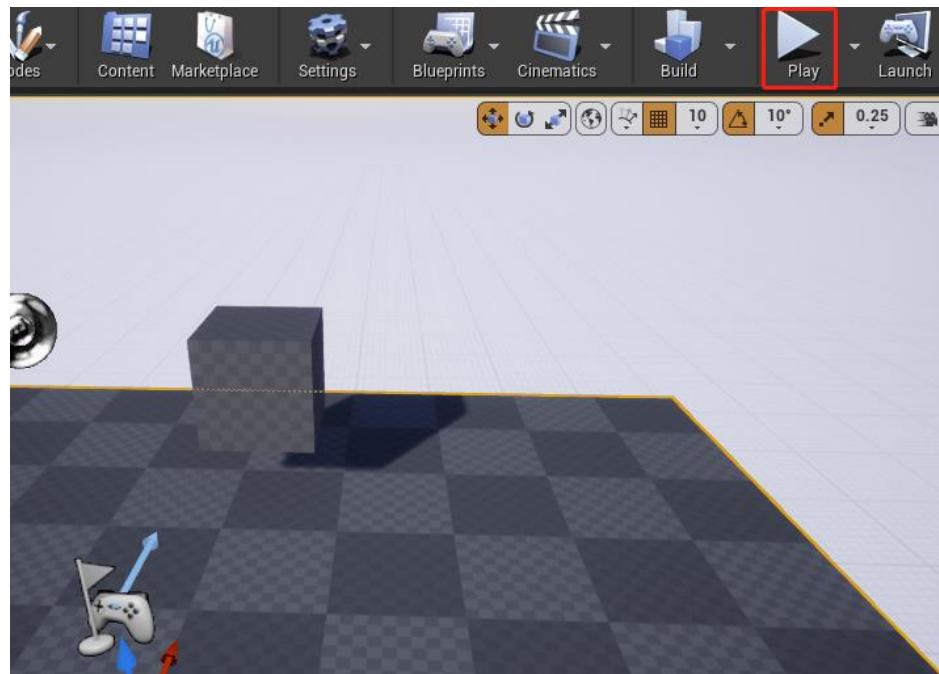


我们在 Tick 事件后添加一个 AddActorLocalRotation 节点，并设置值为 0, 0, 1。其中白色的线为执行引脚，事件触发后会按照白线的路径按顺序执行各个节点，这些节点通常就像一个个函数。

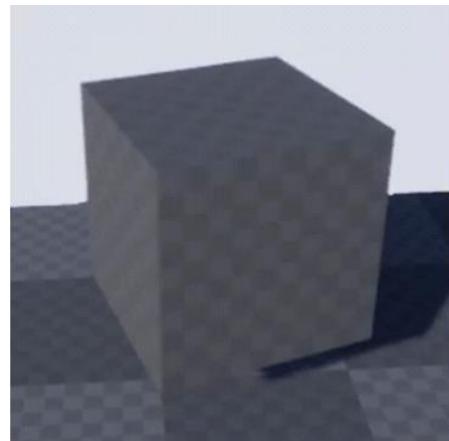
这个 AddActorLocalRotation 节点表示会增加一个 Actor 的本地旋转值，Target==self 表示它执行的目标是自身，也就是说每执行一次该节点都会让自身绕 z 轴旋转 1°，而 Tick 事件每帧都会执行，如果帧率为 120hz，那么每秒绕 z 轴旋转 120°



修改完后需要重新编译+保存，然后可以在视口上方点击 Play 按钮

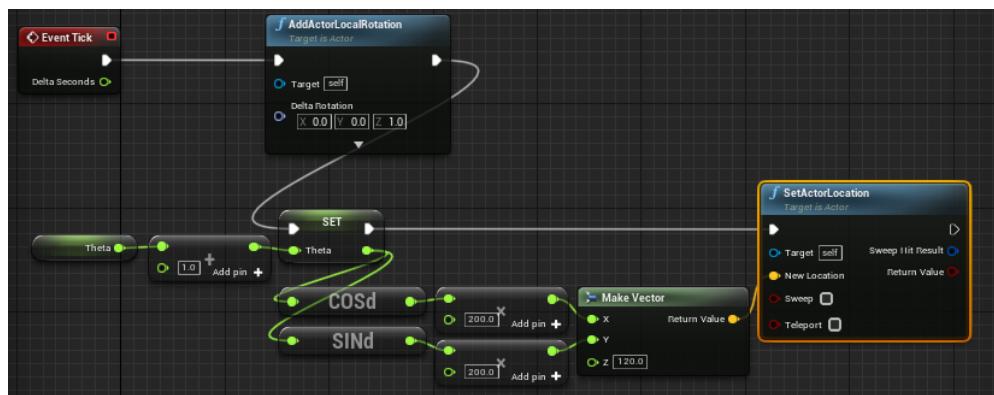


可以看见它一直在沿着自身 z 轴旋转（该物体的 z 轴是自身中心向上的轴）

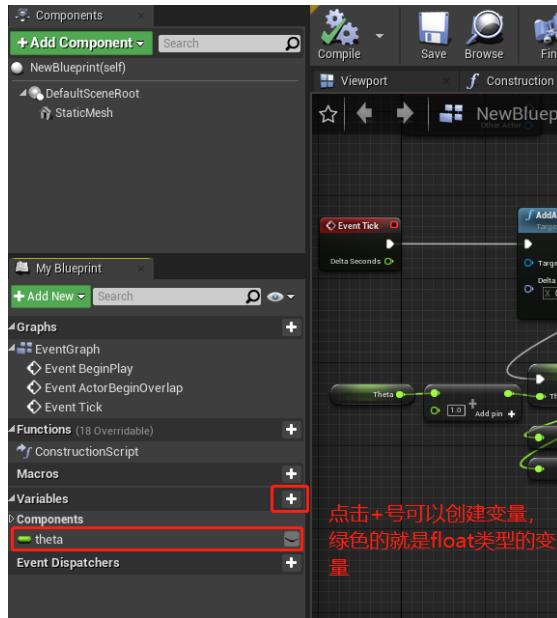


## Step 5：修改事件图表

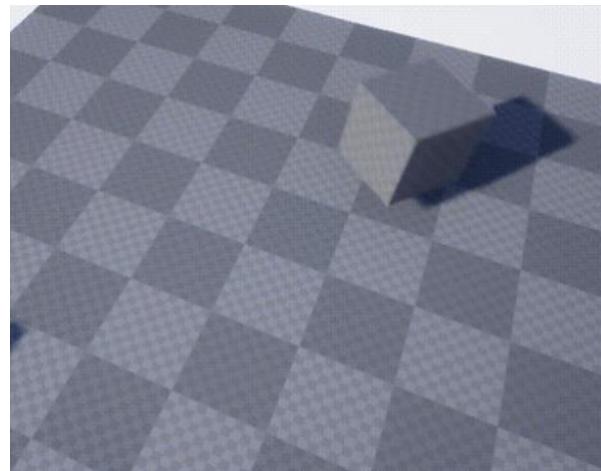
我们还可以继续修改一下它的事件图表，每帧沿着自身 z 轴旋转 1° 后，再计算一个新的位置，并利用 SetActorLocation 重新设置它的位置进行一个圆周运动。



这里创建了一个绿色的 float 变量名为 Theta，并且每帧+1，然后分别求它的 cos 与 sin，乘以半径 200cm，再作为新位置的 x 与 y 坐标，新位置的高度固定为 120cm。



那么此时它每一帧都会绕自身的 z 轴旋转  $1^\circ$ ，并且还会绕场景中的 (0, 0) 点进行一个圆周运动，如果帧率为 120hz 的话，它的自转速度为  $120^\circ /s$ ，圆周旋转速度也为  $120^\circ /s$ 。



## 7. 参考文献

希望通过这个简单的实验能让读者简单了解一下 UnrealEngine，以及它是大致如何运作的，但显然这样是不足以掌握它的，读者一定还对很多方面相当疑惑，如果想要继续了解 UE，这里推荐一些入门教程：

- [1]. [Hour of Code with Unreal Engine \(1 of 5\) - YouTube](https://www.youtube.com/watch?v=bcu6GmTY8mI): <https://www.youtube.com/watch?v=bcu6GmTY8mI>
- [2]. [Your First Hour in Unreal Engine 5.0 Overview - Your First Hour in Unreal Engine 5.0 \(epic games.com\)](https://dev.epicgames.com/community/learning/courses/ZpX/your-first-hour-in-unreal-engine-5-0): <https://dev.epicgames.com/community/learning/courses/ZpX/your-first-hour-in-unreal-engine-5-0>

- 
- n-unreal-engine-5/RPwK/your-first-hour-in-unreal-engine-5-0-overview
- [3]. [【虚幻 4】UE4 初学者系列教程-基础入门全集 | 谌嘉诚 哔哩哔哩 bilibili](https://www.bilibili.com/video/BV1C54y1R7co): <https://www.bilibili.com/video/BV1C54y1R7co>
  - [4]. [虚幻引擎 4 入门 | 虚幻引擎文档 \(unrealengine.com\)](https://docs.unrealengine.com/4.27/zh-CN/Basics/GettingStarted/): <https://docs.unrealengine.com/4.27/zh-CN/Basics/GettingStarted/>

## 8. 常见问题

Q1: 无

A1: 无