**设置角色**



**概述**

无论您的游戏项目或游戏类型是什么，到了某个时间点，您总会需要一些可以在您的场景四处移动的动画角色。这可以是玩家控制的角色，也可以是以某种方式同世界交互的AI驱动的实体。不管怎样，您需要知道如何设置这样的角色，以便它们可以正常地在您的世界中产生动画效果。本文的目的是为您提供如何实现这个任务的高度概述，同时会引导您查看一些专门的文档及示例，以便获得详细信息。出于我们的目的，我们将假设您想创建一个由玩家以某种方式控制的角色。

在这个文档中，我们将谈及可以通过蓝图实现的各种脚本操作。任何可以在蓝图中实现的内容都可以在C++中完成，所以您不要认为仅限于蓝图可视化脚本。本文的最后一部分介绍了示例内容，展示了如何在C++和蓝图中设置这些示例内容。

**工作流程概述**

在虚幻引擎4中设置角色的主要流程如下所示：

1. 使用第三方数字内容创建包（比如3ds Max 或Maya）创建您的美术资源(骨架网格物体)及动画。
2. 将您的骨架网格物体及动画导入到虚幻引擎4中，或者为新骨架网格物体创建一个新骨架资源，或者为同样的或类似的骨架网格物体重用一个现有的骨架资源。
3. 创建PlayerController脚本或蓝图来处理玩家的输入。
4. 创建一个Character或Pawn脚本或蓝图来解析输入及控制角色的实际运动 (不是骨架动画) 。
5. 构建角色的动画蓝图。
6. 创建一个使用您的自定义PlayerController 及任何其他自定义脚本资源的GameMode脚本或蓝图。
7. 运行您的游戏！

这里提到的每一步一般都需要大量的细分步骤操作才能完全成功。该列表只是给出了该流程的总体思想。在接下来的部分中，我们将深入了解每一步中的具体细节及您可以如何应用它们。

**创建美术资源**



在多方面，创建美术资源可能是角色开发过程中最具挑战性的一部分。一般，在您使用虚幻引擎之前，您要花费大量的时间进行设计、建模、贴图 、骨架绑定及动画设计。尽管我们不能教您进行角色设计和动画制作的细节，但我们确实有一些工具可以辅助您完成这个过程。

**Maya绑定工具**

在Epic, 我们的很多美术师都使用Maya来制作角色的动画。因此，我们的团队开发了一个高级绑定工具，用于简化绑定过程，从而把数小时的工作量降低为几分钟。

动画绑定工具集 (A.R.T)是一个全面的工具集，可以处理角色的骨架创建、骨架放置、rig(定位标尺)创建，同时还包含很多动画工具，从而可以加速动画师的工作流程，目前针对Maya 2013版本进行的测试及确认。功能包括：

* 直观的用户界面引导用户操作每一步
* 完成骨架创建。
* 通过直观地控制rig（定位标尺）的位置来自定义骨架的放置位置。
* **用不到一分钟的时间** 就可以发布整个rig（定位标尺），并将其发送给动画团队
* 提供了给最终角色进行简单贴图的功能。
* 具有用户界面的完整的动画工具，包括导入运动捕获数据或动画数据、导出数据到FBX、姿势工具、镜像、空间切换等！

**导入骨架网格物体**

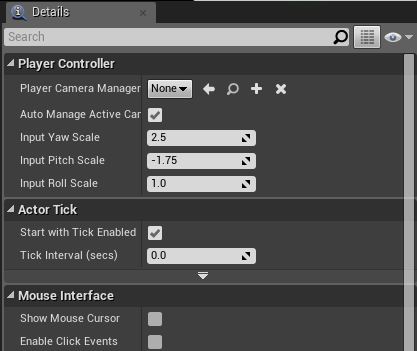


FBX 是由 Autodesk 拥有并开发的文件格式。它使数字内容创建应用程序（比如 Autodesk MotionBuilder、Autodesk Maya 和 Autodesk 3ds Max）间具有互操作性。 Autodesk MotionBuilder 软件本身就支持 FBX，而 Autodesk Maya 和 Autodesk 3ds Max 需要使用 FBX 插件。

虚幻引擎有一个 FBX 导入通道，它允许简单地转换在任何的支持该格式的数字内容创建应用程序中创建的内容。

在您创建带动画的角色的过程中，正确地把您的骨架网格物体导入到UE4中是至关重要的一步。虚幻引擎包含一个强大的导入系统，它具有大量的选项来加速导入过程。

**创建Player Controller（玩家控制器）**



PlayerController是一种特殊类型的脚本或蓝图，它们的主要作用是把玩家的输入解析为可以驱动角色的事件。比如，它可以控制如何把一个控制器上的摇杆向上推动来产生一个事件，最终该事件用于推动角色在屏幕上向前运动。

PlayerController是虚幻引擎中现有的类。在编辑器中，您可以使用PlayerController作为父类创建一个新蓝图，然后使用该蓝图来设置收到玩家输入时要发生的事件。

要想查看一个自定义的蓝图PlayerController的示例，您可以在编辑器中启动一个新项目( **File（文件） > New Project（新建项目）**) 并选择 **Blueprint Top Down(蓝图自上而下)** 蓝图模板。所有基于蓝图的模板都包含某种PlayerController（或者是默认的Player Controller或者是一个玩家控制器蓝图），如果您想查看一个使用Player Controller的自定义应用程序，那么使用 **Blueprint Top Down（蓝图自上而下）** 模板是最简单的方法。

在新项目中，您可以在 **类别查看器** 中搜索 PlayerController，关闭 **类别查看器** 中的过滤器。**双击** 这个资源将会打开它，您可以自己查看它的设置。

您可以通过创建一个新项目(**File（文件） > New Project（新建项目）**)并选择 **Blueprint Top Down（蓝图自上而下）** 模板，来查看以C++脚本编写的PlayerController。

**创建一个Pawn或Character蓝图**

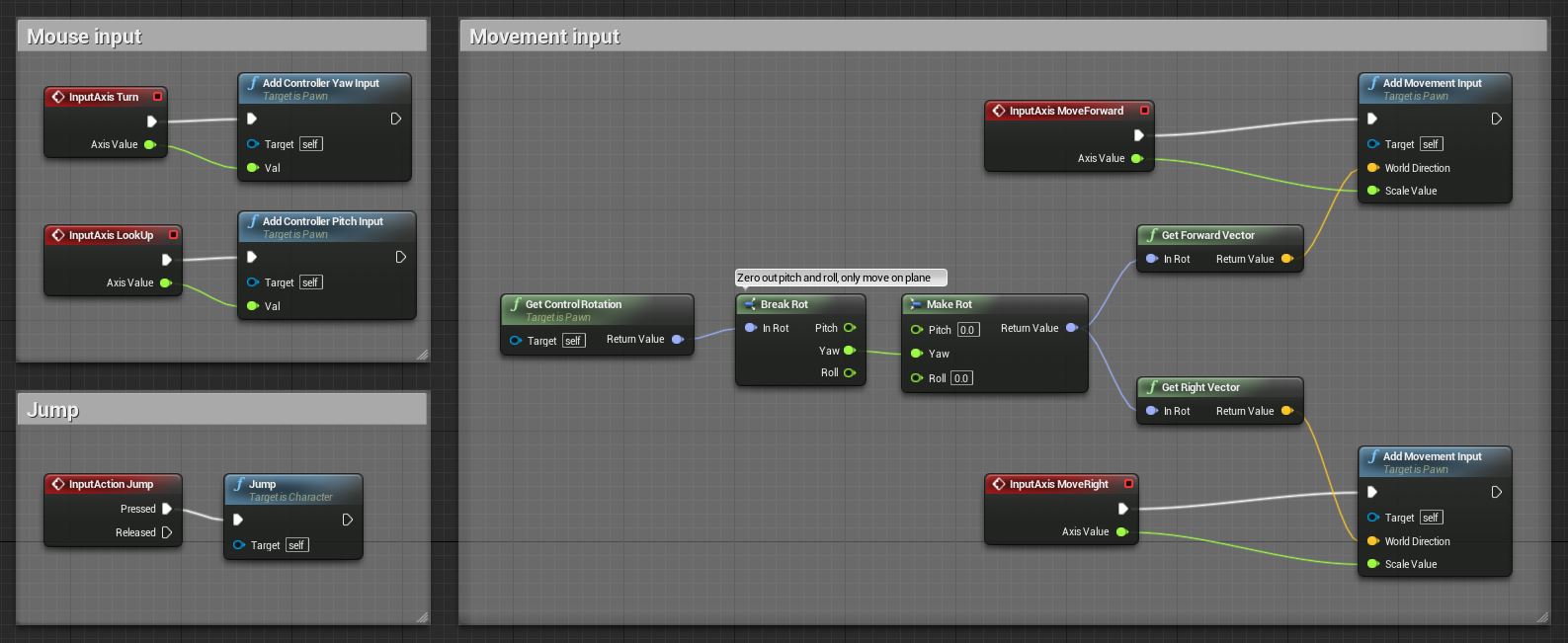


一旦您创建了您的PlayerController，您的系统现在就准备好处理玩家的输入了。但是，现在您必须把这些输入转换为可以反过来在屏幕上驱动玩家到处移动的东西。这意味着，这些输入需要 转换 （或解析）为动作。这里，就是Pawn或Character类发挥作用的地方了。

**选择Pawn或Character**

您会注意到这里我们提到了两种潜在的类： Pawn和Character。这两个类都用于游戏中的实体，这些实体或者由玩家控制或者由游戏中的AI控制。二者的关键区别是Character类是Pawn类的扩展，加入了玩家物理、特定网格物体支持、及创建游戏中可玩的角色时所需的通用的处理类型。出于我们的目的，我们将使用Character类。对于场景中仅需AI驱动的简单元素，您一般用Pawn就可以成功。

**Character类设置**



您的Character 类将以PlayerController中触发的事件开始，并使用脚本(包括蓝图可视化脚本)来控制使用这些输入到底 做什么 以及如何使用它们来控制角色。比如，当PlayerController简单地创建一个事件来将控制器上的摇杆向上移动时，Character 则负责接收那个事件并使用它来驱动玩家前进。

Character类还包含到一个骨架网格物体的引用，该骨架网格物体是玩家在游戏中看到的角色的基础。在第一人称游戏中，通常仅有一对悬浮的手臂，但是，如果您需要角色躯体适当地在环境中产生阴影，那么可能会有全身。对于第三人称游戏，该网格物体就是代表角色的网格物体。

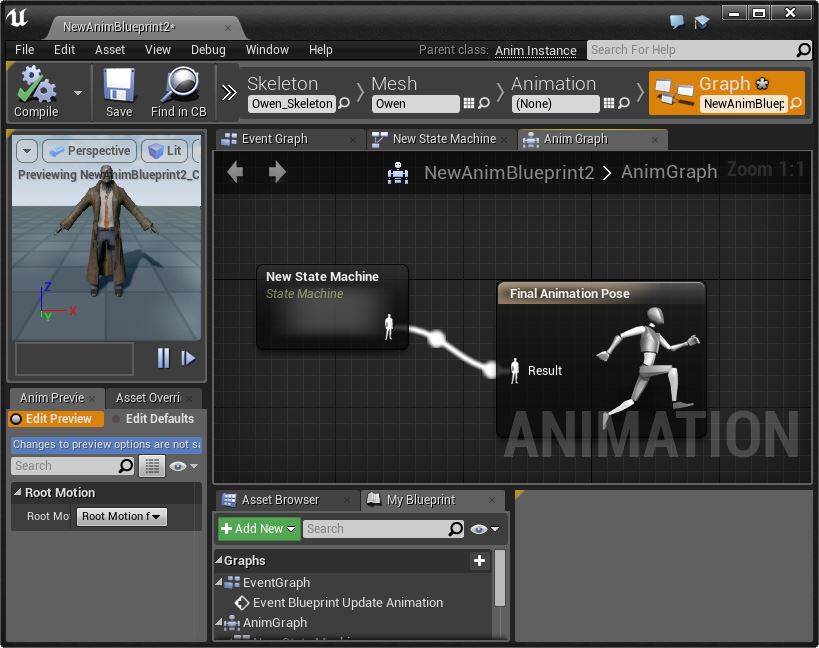
角色上的运动通常是通过给物理形状(一般是胶囊型)应用一些运动来处理的。这种运动也和一种MovementMode（运动模式）相吻合。这是个用于跟踪玩家做作的动作（也就是走路、跑动、降落、游泳等）的枚举值。该信息稍后将用于驱动骨架网格物体上播放的动画。

要想查看一个自定义的蓝图Character 类的示例，那么您可以在编辑器中启动一个新项目( **File（文件） > New Project（新建项目）**) 并选择First Person（第一人称）或Third Person（第三人称）的蓝图模板。所有基于蓝图的模板都包含某种Character ，但是我们推荐First Person或Third Person模板，因为它们总体非常简单而且是最常用的类型。

一旦位于新的项目中，您可以在 **类别查看器** 中搜索Character，在Game文件夹中使用Buleprints进行过滤。**双击** 这个资源将会打开它，您可以自己查看它的设置。

您可以通过创建一个新项目(**File > New Project**)并选择First Person 或Third Person代码模板，来查看以C++脚本编写的PlayerController。

**动画蓝图**



您将动画附加到角色上的大部分繁重处理都将在动画蓝图中完成。

当您在角色蓝图中定义了骨架网格物体Actor在世界中的移动方式后，您可以在动画蓝图中基于这些运动（比如速度）开始分配特定动画。

在 动画蓝图 中，有两个主要部分相互协作来创建每帧的最终姿势。 **EventGraph(事件图表)** 可以更新值，然后可以在 **AnimGraph（动画图表）** 中使用这些值来 驱动 状态机 、 混合空间、可以混合多个 动画序列 的其他混合节点、或者 驱动那些可以触发到其他系统的通知，从而发生动态动画效果的姿势。

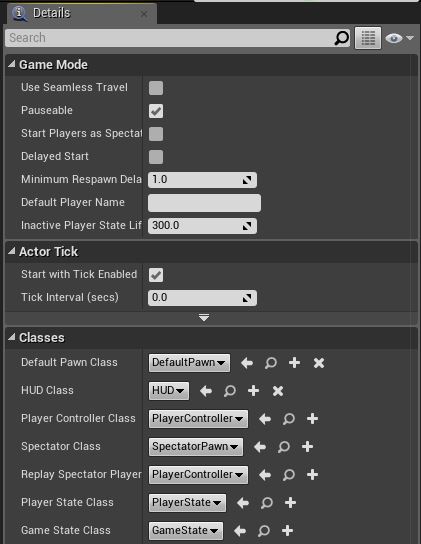
到目前为止，动画蓝图是角色设置过程中最复杂的方面。这里是您的所有数据汇集起来真正地使骨架网格物体执行适当的动画的地方。为了完全地理解动画蓝图及其它们的威力，您需要知道很多不同的动画资源，这包括：

* [状态机](http://api.unrealengine.com/CHN/Engine/Animation/StateMachines/index.html)
* [混合空间](http://api.unrealengine.com/CHN/Engine/Animation/Blendspaces/index.html)
* [动画序列](http://api.unrealengine.com/CHN/Engine/Animation/Sequences/index.html)

这些仅是冰山一角。如果你看下[动画蓝图](http://api.unrealengine.com/CHN/Engine/Animation/AnimBlueprints/index.html) 文档，及某些示例内容中包含的一些动画蓝图，比如First（第一人称）和Third Person Templates（第三人称模板）以及 Content Examples项目中您可以找到的资源，那么您将会做得更好。

一旦您创建了定义您的角色动作的动画蓝图，您则需要确保将该蓝图分配给 **Anim Blueprint Generated Class（动画蓝图生成的类）** 属性，可以在角色蓝图的 Mesh Component（网格物体组件） 下找到该属性。必须要这么做，因为您可以为每个骨架网格物体提供多个动画蓝图，且角色蓝图需要知道它要把必要的动画及变量数据发送到哪个动画蓝图中。

**游戏模式设置**

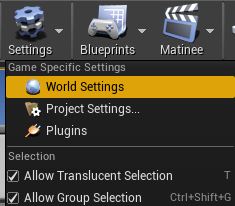


GameMode（游戏模式）是一种特殊类型的类，用于定义您的游戏。一般来讲，它就是一组属性，定义了您游戏的实质类别。您将设置的主要属性包括：

* **Default Pawn Class（默认Pawn类）** - 该项存放了您为您游戏中的角色设置的Character类。
* **HUD Class** - 该项包含了创建的任何特殊的平头显示(HUD)类,本文中没有涉及该类。
* **PlayerController Class（玩家控制器类）** - 该项将存放您为游戏中的角色设置的自定义的PlayerController。
* **Spectator Class（观察者类）** - 该项存放了用于控制观察者的任何特殊类或仅观看动作的被动玩家。本文中没有涉及到该项。
* **Game State Class（游戏状态类）** - 该类是用于控制您的游戏如何运行的任何特殊规则或设置，但是本文没有涉及该项。
* **Player State Class（玩家状态类）** - 该类定义了复制相关玩家信息到所有客户端的所有特殊规则。

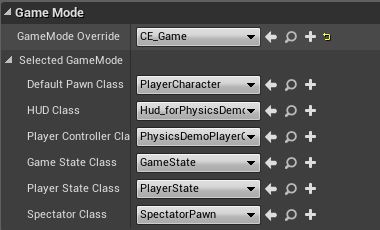
要想测试您的角色，您至少需要设置 Default Pawn Class(默认Pawn类)和PlayerController Class属性。

**世界设置**



一旦您设置好了您的GameMode，在运行您的自定义角色之前，最后一步是确保当前的关卡正在使用您的GameMode(游戏模式)。这可以通过使用World Settings(世界设置)选卡来完成，通过位于主工具条顶部的World Settings按钮可以访问该选卡。

在世界设置中，您需要确保将 GameMode Override 设置为您的 GameMode 类的名称。一旦您完成这些操作，您便可以保存及测试您的新角色了！



**总结**

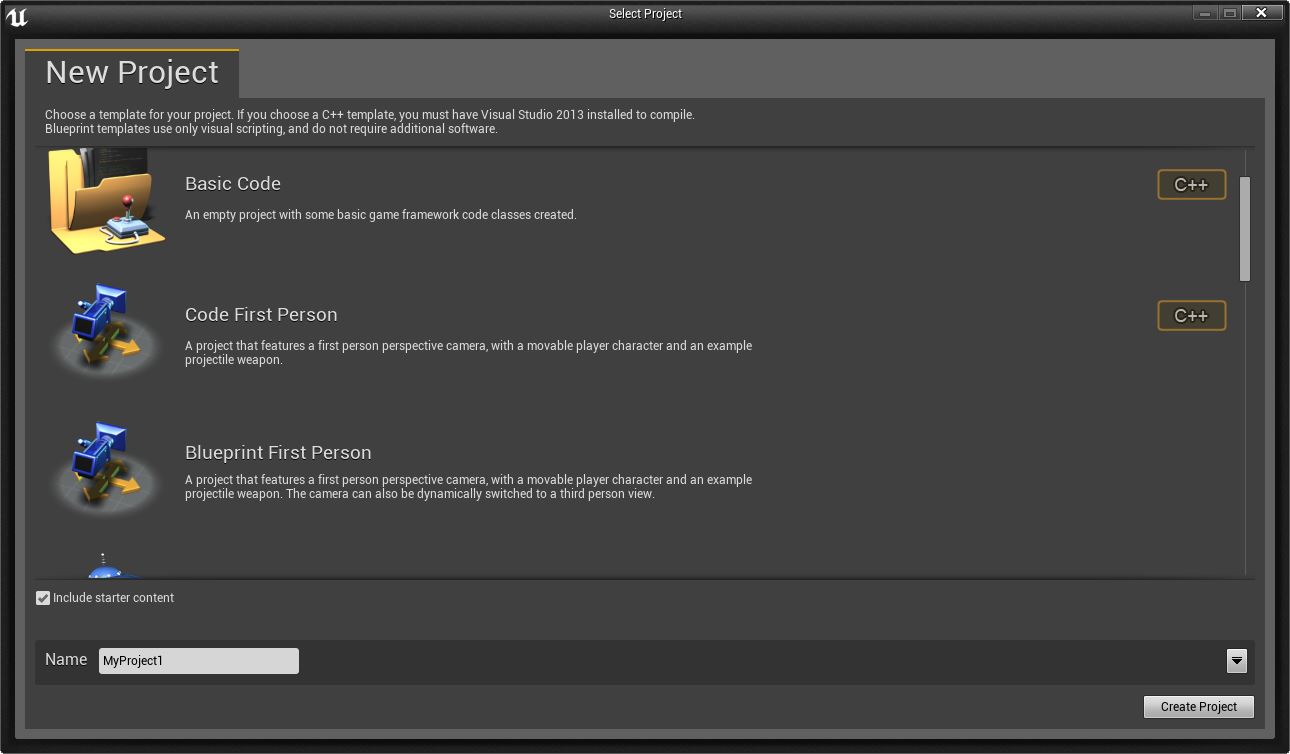
让我们以倒序的顺序来总结一下这个设置过程：

* 您的关卡的World Settings(世界设置)设置为您正使用的GameMode。
* GameMode指出了您运行游戏所需的Pawn (Character)类和PlayerController类。
* Character 类
  + 包含您通过FBX导入的骨架网格物体
  + 从PlayerController类取入数据并将其转换为运动(而不是动画)
  + 在Mesh Component(网格物体组件)中存储使用哪个动画蓝图来驱动骨架动画。
* 动画蓝图
  + 从Character类中取入数据到其Event Graph(事件图表)中
  + 使用该数据来驱动状态机、混合空间及其他资源。
  + 这些资源使用动画序列 (来自FBX文件中的骨架动画数据)来使角色产生动画。
* 将动画蓝图的最终结果应用到您的骨架网格物体上，以便您可以看到角色在游戏中的动画效果。

**包含的示例**

您可以查看引擎中的几个示例来看一下如何进行设置并自己进行尝试。我们提供了Templates(模板)，它们是您制作您自己游戏时可以使用的基本的项目类型，还提供了Content Examples（内容示例），它们由Epic的美术师及技术人员创建的内容示例。

**模板**



当您在虚幻引擎中创建新项目 (**File（文件） > New Project（新建项目）**) 时，您可以选择一个模板。实际上，所有这些模板都将使用它们自己的 GameModes（游戏模式）、Character Blueprints（角色蓝图）、Animation Blueprints（动画蓝图）及其本文提到的其他资源。出于简化及清晰的目的，我们强烈推荐您看下 First Person或Third Person模板。

注意，每个模板都有代码形式或蓝图形式。这样做是为了让您可以选择您最适合的方式进行开发。如果您是程序员，那么您可能想使用Code Template（代码模板）。如果您是偏向于美工的开发人员，那么您可能更适合使用Blueprint Template(蓝图模板)。请注意，这两种形式的模板不是相互排斥的：您可以添加代码类到Blueprint Template项目中，就像您可以添加一个新的蓝图类到 Code Template项目中一样！

**内容示例**



Content Examples(内容示例)是Epic的美术师及技术人员设计的专业版内容。这些内容可以在一个名称为 **ContentExamples** 的项目中找到，用户可以通过 **Marketplace** 下载该项目。其中特别重要的是 Animation.umap 关卡中的资源，该关卡展示了角色上的骨架网格物体动画的各种应用。