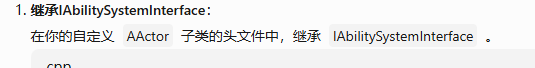
要在Unreal Engine中为你的AActor子类集成**游戏玩法技能系统（Gameplay Ability System, GAS）​**，你需要按照以下步骤进行设置。这些步骤包括实现

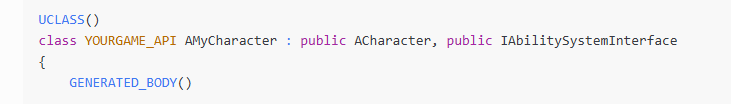
IAbilitySystemInterface接口、

覆写GetAbilitySystemComponent函数，

以及正确配置技能系统组件（Ability System Component, ASC）。

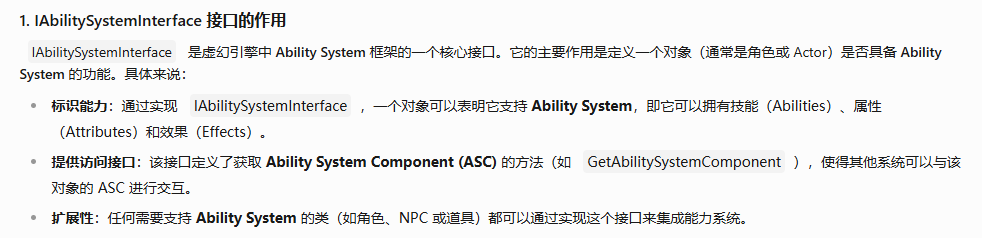
此外，还会涵盖一些高级用法，例如让某些Actor共享另一个Actor的ASC，以支持持久化的技能数据和冷却计时器。

















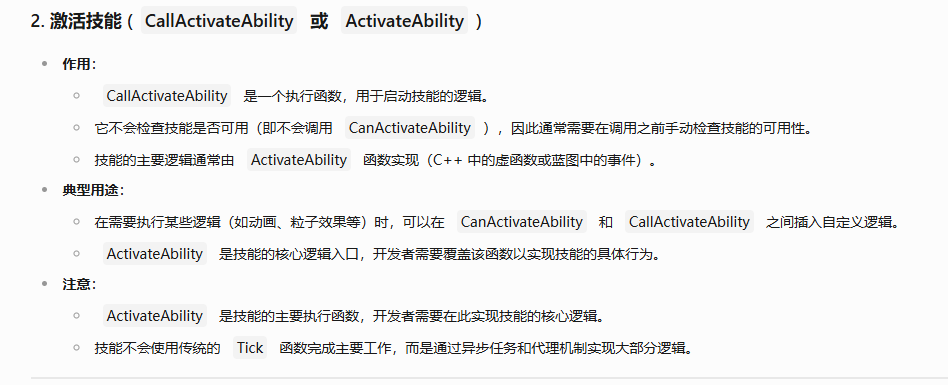










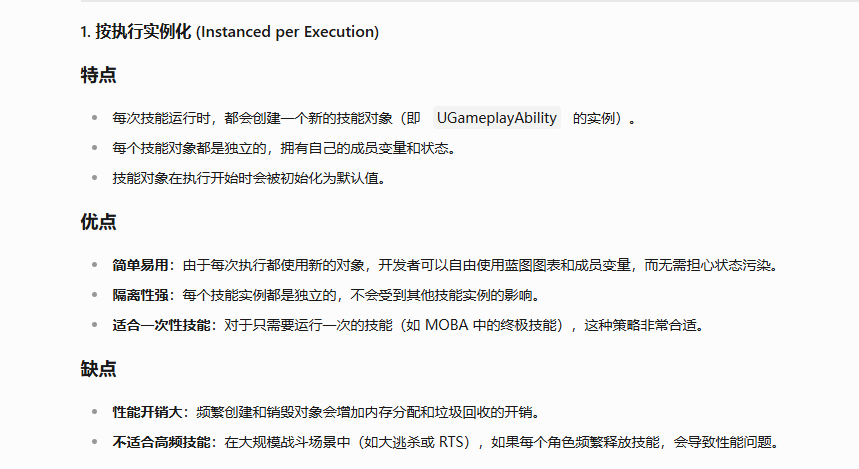


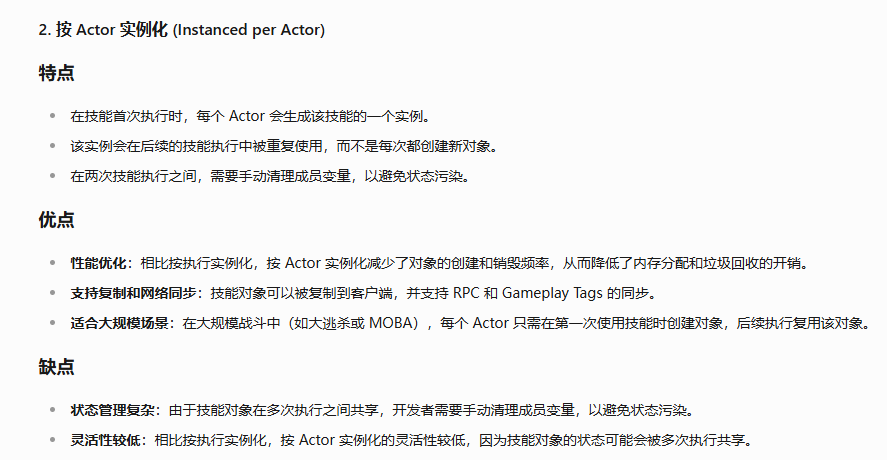


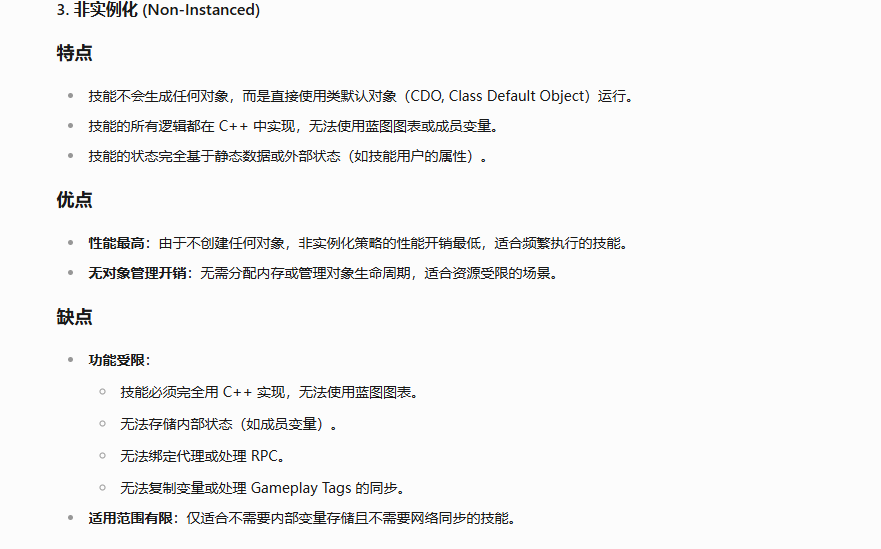














**游戏技能系统（Gameplay Ability System）** 使用 **游戏属性（Gameplay Attributes）**（FGameplayAttribute）来保存、计算和修改与游戏相关的浮点值。这些值可以描述其拥有者的任何特征，比如角色的剩余生命值，车辆的最高速度，或者物品在损坏前可以使用的次数。游戏技能系统中的角色将他们的游戏属性存储在一个 **属性集（Attribute Set）** 中，该属性集有助于管理游戏属性与系统其他部分之间的交互，并将自己注册到角色的技能系统组件中。这些交互包括对数值范围的限定、临时数值变更、对永久改变基础值的事件做出反应等。



**什么是Gameplay技能？**

**Gameplay技能（Gameplay Ability）** 是Actor可以拥有并反复触发的一种游戏内动作。常见例子有咒语、特殊攻击或物品触发的效果。这一概念在电子游戏中十分普遍，常常被视为理所当然，不过运行技能所涉及的过程常常很复杂，并需要具体的时序。例如，虽然编写攻击激活代码本身相当简单，但在长期项目的过程中，随着你为玩家添加或去除增益或减益效果、增加资源成本、连击系统及其他细节，构建技能的复杂度可能会呈爆炸式增长。因此，在设计虚幻引擎的Gameplay技能系统时，有三大注意事项。



















**协调技能的执行**

技能必须能够在执行期间使用特定时序与多个不同的系统交互。这些交互可以包括：

* 激活动画蒙太奇。
* 临时控制角色的移动。
* 触发视觉效果。
* 执行重叠或碰撞事件。
* 临时或永久更改角色的统计数据。
* 增加或减少游戏内资源。
* 允许或阻止激活其他技能。
* 处理技能冷却以限制技能使用。
* 被游戏内事件中断。
* 取消进行中的其他技能。
* 对角色做出重大状态更改，例如激活新的移动模式。
* 响应其他交互中途的输入。
* 更新UI元素，显示技能的游戏内状态。

根据技能的工作方式，它可以在处于活动状态时在许多不同时间点执行其中任意交互，包括在动画中途，并且一些效果可能需要在技能本身完成后持久存在。

**ameplay技能系统的组件**

Gameplay技能系统指在处理所有这些用例，方法是将技能建模为负责自身执行的完全独立的实体。该系统由多个组件构成：

* 带有 **技能系统组件（Ability System Component）** 的所属Actor，维持该Actor拥有的所有技能的列表，并处理激活。
* **Gameplay技能蓝图（Gameplay Ability Blueprints）** ，表示各个技能，并协调其游戏内执行。
  + 由 **Gameplay技能任务（Gameplay Ability Tasks）** 以及其他函数构成。
* **属性集（Attribute Set）** ，附加到技能系统组件。
  + 包含 **Gameplay属性（Gameplay Attributes）** ，用于驱动计算或表示资源。
* **Gameplay效果（Gameplay Effects）** ，处理Actor因使用技能而发生的更改。
  + **Gameplay效果计算（Gameplay Effect Calculations）** ，提供模块化、可复用的方法来计算效果。
  + **Gameplay提示（Gameplay Cues）** ，与Gameplay效果关联，并提供数据驱动的方法来处理视觉效果。

下面的分段更详细地总结了这些类。