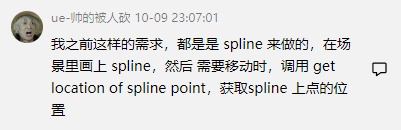
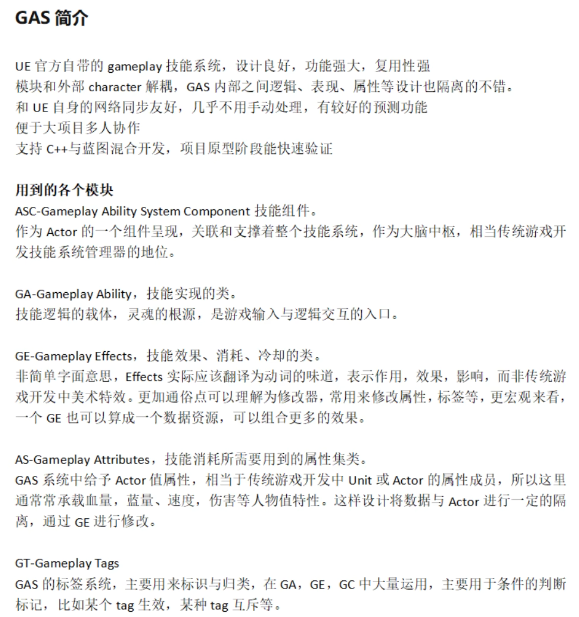
路径与路径点



Gas



参考文献：

https://www.bilibili.com/video/BV1zD4y1X77M/?spm\_id\_from=333.337.search-card.all.click&vd\_source=79fbe818ff96aae3677e123f0374bd91

如何设计一个易于扩展的技能系统

逻辑部分

技能的获得和释放

触发判断的条件

Buff系统

视听部分

动作动画

特效

声效

数据部分

数值计算

数据配置

联机同步

ASC,UAbilitySystemComponent，谁释放技能，who

GA,UGameplayAbility,技能逻辑，技能的逻辑，how

GE，UGameplayEffect，技能效果，change

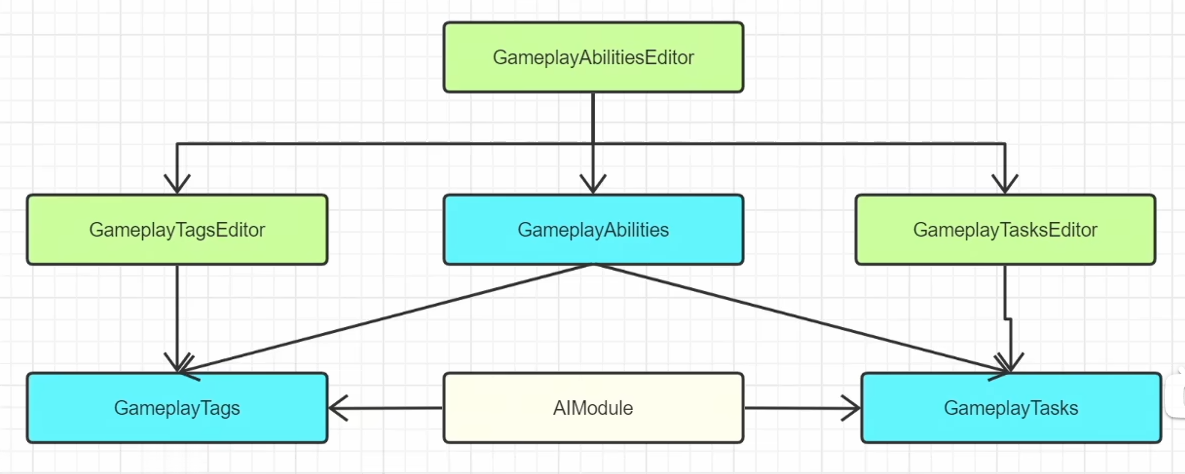
GC，UGameplayCueNotify，特效部分，视觉特效

Attribute，FGameplayAttribute，属性。多个Attribute组合成AttributeSet，挂在Actor上

Tag，FGameplayTag，技能涉及的条件

Task，UGameplayTask，异步操作，技能的长时行动

Event，FGameplayEventData，ASC之间发送一些游戏事件来通知对方。技能消息事件



Editer负责绘制UI

GameplayTags

层次化的字符串标签A.B.C

轻量FName，可附加到各类上做搜索条件

整体所有Tag构成一颗Tag树

GE

所有技能效果由GE管理

决定一个技能的逻辑效果（伤害扣血）、Tag标签（眩晕沉默）、能力提供、触发特效

纯配置蓝图，不需要重载函数

GE是修改Attribute的唯一通道

GC

决定一个技能的视觉效果

可全局配置TagHandler的映射

可通过Effect触发，也可手动触发

Static：一次性

Actor：持久

可重载：OnActive/WhileActive/Executed/Removed

GA

能力是很广义的抽象

也包括收击、格挡、击溃等这种细粒度行为

基本上角色身上大部分行为都可以用能力来表示

不要把基础移动、射线检测、UI交互当做能力

能力更应该表示专门触发一件什么事情，而不是平时一直在做的事情

在不同的Actor上学习、取消、释放不同能力构成了主体逻辑

主要逻辑的书写处

可重载：ActivateAbility/CommitAbility/CancelAbility/EndAbility

GA一般是在外部被OwnerActor触发的

通过调用TryActivateAbility

Task

执行异步任务的框架

可被单独使用

可用于异步长时动画动作的触发和等待

已经预实现好一系列Tasks

可重载：Activate、TickTask

Event

手动触发游戏事件

事件靠Tag识别，可携带Payload数据

可触发技能

在另一端可等待具体Tag事件触发

ASC

作为核心，管理协调其他组件

拥有ASC的Actor才拥有释放技能的能力

ASC放在Pawn还是PlayerState上，这是一个问题

后面会用到链接所示的项目来进行cpp的学习和gas的学习

<https://www.bilibili.com/video/BV12m4y1W76X?spm_id_from=333.788.videopod.episodes&vd_source=79fbe818ff96aae3677e123f0374bd91>

<https://www.bilibili.com/video/BV1ZX4y177vz/?spm_id_from=333.337.search-card.all.click&vd_source=79fbe818ff96aae3677e123f0374bd91>