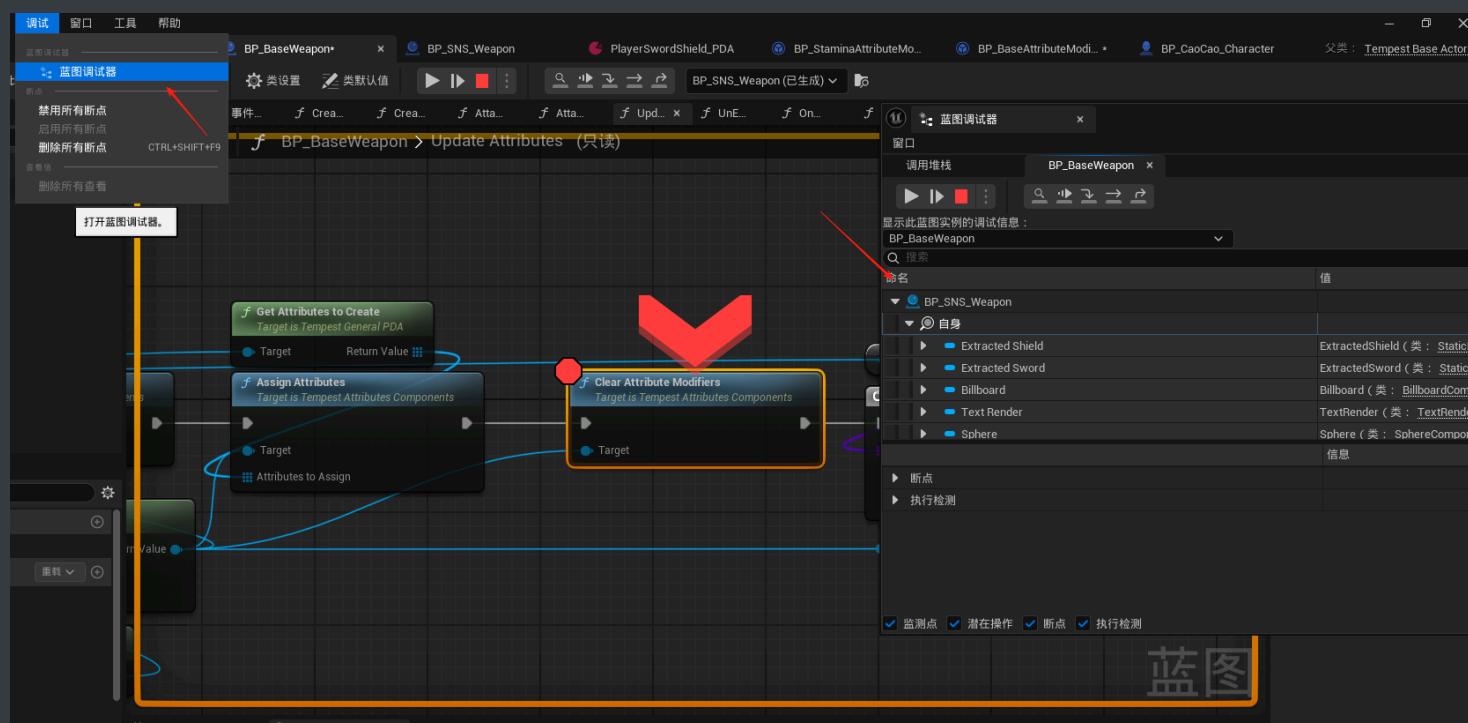
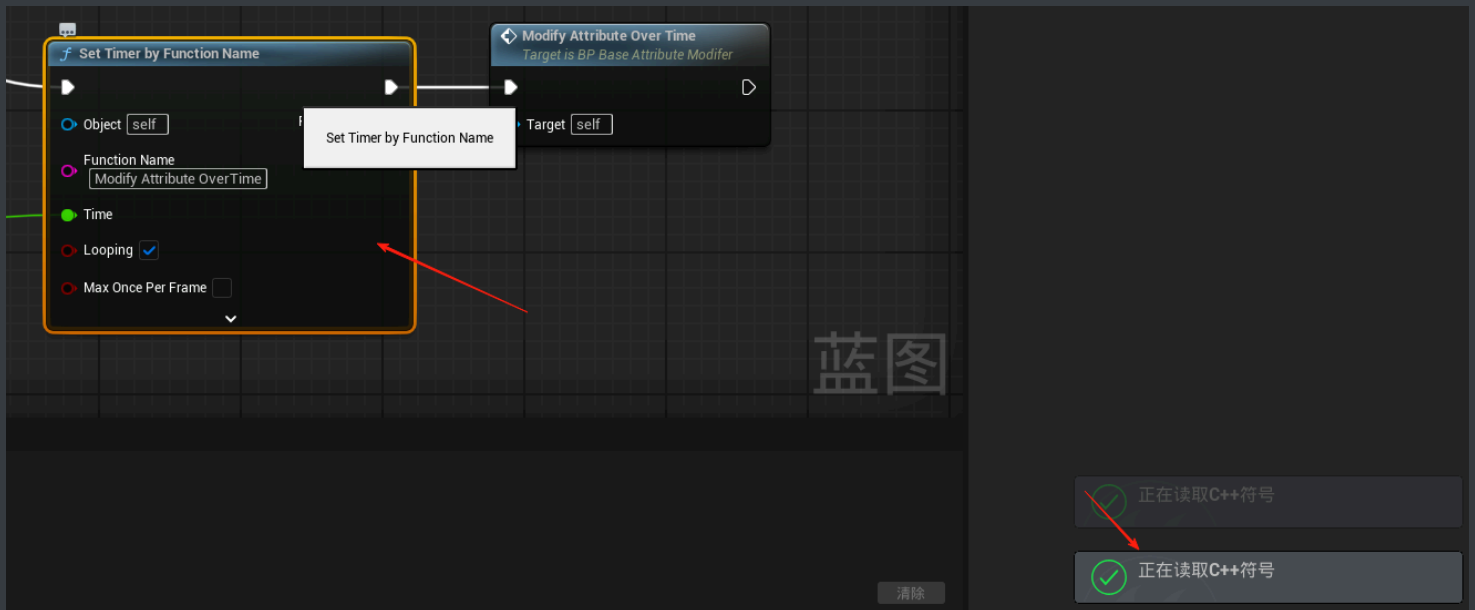


在需要断点的地方F9可以断点，点调试里面的蓝图调试器可以查看调用的信息



如果一个一个方法前面有个箭头，代表这个方法是在事件图表里面定义的，在事件图表里面去实现





其他的方法是可以跳转的

蓝图与C++之间交互是通过宏来实现的，只有被宏定义了的方法变量类才会进入UE的声明周期被管理，不然就得你手动管理，不同的宏关键字有不同的效果，比如在蓝图中显示，在蓝图中可编译/复写等等

基础概念对应

Unity 概念	UE 概念	说明
GameObject	Actor	UE 中的基础对象
Transform	SceneComponent	处理位置、旋转、缩放
Prefab	Blueprint Class	可复用的资源模板
Scene	Level	游戏场景
Script	Blueprint/C++ Class	游戏逻辑实现方式

常用 API 对应

对象管理

- **Unity:** Instantiate()
UE: UWorld::SpawnActor()

- **Unity:** Destroy()
UE: AActor::Destroy()

组件系统

- **Unity:** GetComponent<T>()
UE: GetComponentByClass<T>()
- **Unity:** AddComponent<T>()
UE: AddActorComponent

类型转化

- **Unity:** Rigidbody rb = GetComponent<Rigidbody>();
UE: AMyActor* MyActor = Cast<AMyActor>(SomeActor)

常用方法

1. 对象有效性检查

Unity	Unreal Engine (蓝图)	Unreal Engine (C++)	说明
<code>obj != null</code>	Is Valid 节点	<code>IsValid()</code> 或 <code>IsValid(对象)</code>	检查对象是否有效
<code>Destroy(gameObject)</code>	Destroy Actor 节点	<code>Destroy()</code> 或 <code>Actor->Destroy()</code>	销毁对象

2. Transfrom 操作

Unity	Unreal Engine (蓝图)	Unreal Engine (C++)	说明
<code>transform.position</code>	Get Actor Location	<code>GetActorLocation()</code>	获取位置
<code>transform.rotation</code>	Get Actor Rotation	<code>GetActorRotation()</code>	获取旋转
<code>transform.localScale</code>	Get Actor Scale	<code>GetActorScale()</code>	获取缩放

3. 时间控制

Unity	Unreal Engine (蓝图)	Unreal Engine (C++)	说明
<code>Time.deltaTime</code>	Get World Delta Seconds	<code>GetWorld()->GetDeltaSeconds()</code>	帧间隔时间
<code>Invoke()</code>	Delay 节点	<code>GetWorld()->GetTimerManager().SetTimer()</code>	延迟执行
<code>StartCoroutine()</code>	Timeline 或 Async Task	<code>AsyncTask</code> 或 <code>Latent Actions</code>	协程/异步操作

4. 调试与日志

Unity	Unreal Engine (蓝图)	Unreal Engine (C++)	说明
<code>Debug.Log()</code>	Print String 节点	<code>UE_LOG()</code>	打印日志
<code>Debug.DrawLine()</code>	Draw Debug Line	<code>DrawDebugLine()</code>	绘制调试线