实现角色的：冲刺&击退、眩晕&击飞

无论冲刺还是击退，本质都是给角色一个力让他向着力的方向位移一段距离。这种情况下，位移不能穿墙。如果需要位移技能能够穿墙，则是BP\_总类的胶囊体组件的SetCollisionResponseToChannel节点。

击飞就是提供一个向上方的力，而眩晕和击飞期间，我们需要断开角色与控制器之间的链接。

无论哪种，都是角色会有的属性。

我们在BP\_总类中实现这种效果

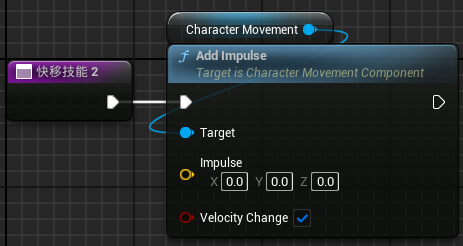
实现冲刺

首先在BP\_总类中，在细节中找到“质量（公斤）”。来启用质量。

我们设定，所有英雄的质量默认都为100kg。韧性可以用在设置角色的质量上。

具体怎么给角色提供这个力，就是角色移动组件的AddImpulse函数。

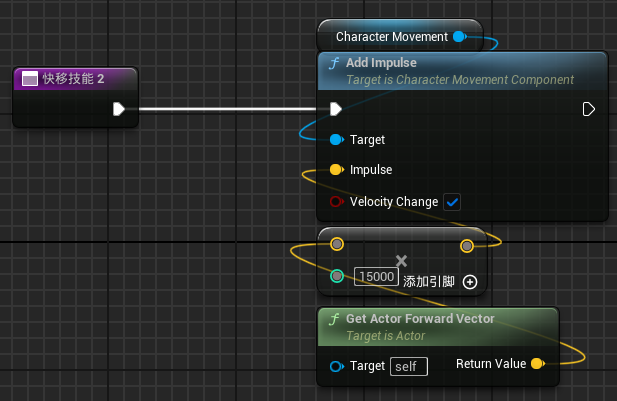
（勾选VelocityChange。这样角色不会突然加速，而是有缓冲的快速加速）



Impulse即为力量的方向，我们要实现向移动方向的快移技能，也就是移动组件的GetVelocity节点或者GetActorForwardVector节点

注意，两个函数的返回值都是Vector类型，但是，也是有区别的。

我们先实现向前方移快移的能力。



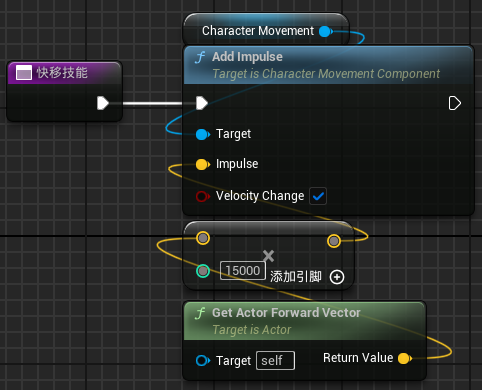
这个时候我们需要提供一个很大的数值，才能让我们的位移技能有明显的效果。这是因为摩擦力的影响。为了能更好更准确的实现快移指定距离，我们在位移期间将角色移动的摩擦力设置为0，并在位移之后将摩擦力设置回原来的大小。

这时，我们就需要创建一个全局变量来保存原来的摩擦力大小。首先找到移动组件的GetGroundFriction节点。



将它在EventBeginPlay时提升为变量。（或者记住默认的摩擦力，默认的GroundFriction就是8。在位移结束之后将GroundFriction重新设置为8即可）。

也就是说，我们需要在函数中实现：SetGroundFriction延迟之后再SetGroundFriction。但是延迟并不能在函数中实现。所以：或者我们使用自定义事件来实现快移技能，或者我们不用SetGroundFriction。这里我们选择后者。因为再考虑到别的功能，自定义事件会很麻烦。

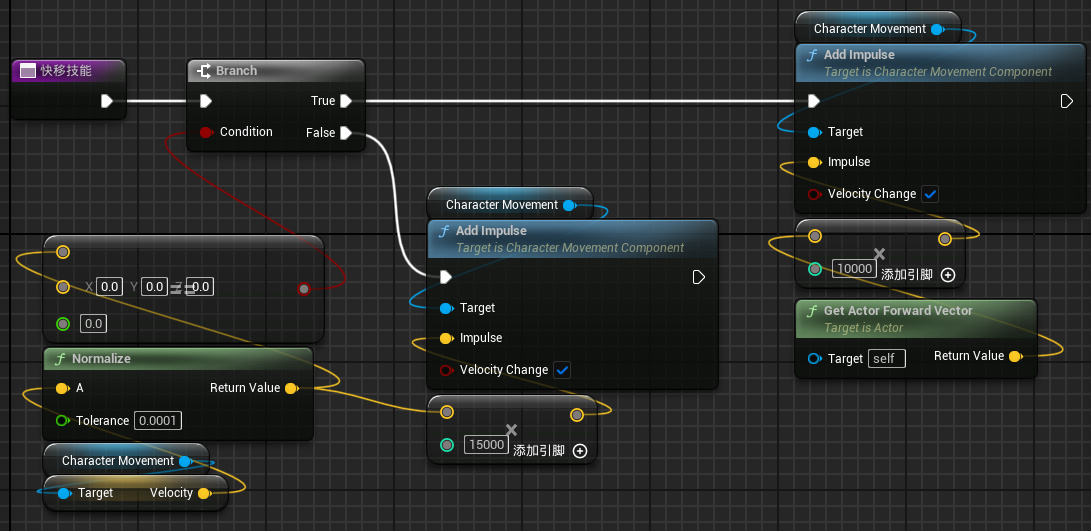


现在我们则需要实现：移动时朝着移动方向快移，不移动时朝着前方快移。

当没有进行移动时，GetVelocity是不管用的。就算在进行移动了，GetVelocity也是有大小的。而GetActorForwardVector是没有大小的。所以我们要给GetVelocity取元。

我们也需要实现：速度为0时向前方移动，速度不为0时向运动方向移动。

这个的实现需要判断



然后将给的力的大小，转换成位移的距离，作为函数的输入值。

然后是实现快移技能。

当我们要使用位移技能时，会有可能穿墙，或者穿透Pawn单位。穿透效果就是胶囊体组件的SetCollisionResponseToChannel函数



我们需要在位移完成后，将碰撞重新设为阻挡。也就是NewResponse参数。在动态场景内强制改为碰撞的话，会站在动态场景的上面。

当Channel为Pawn时可穿透Pawn。

所以我们需要在快移技能完成之后，设置阻挡。这期间是有个Delay的。这个Delay不好计算，我们就设计这个Delay为1。首先默认位移时间就是1，其次这个位移时间应当是需要控制的参数。



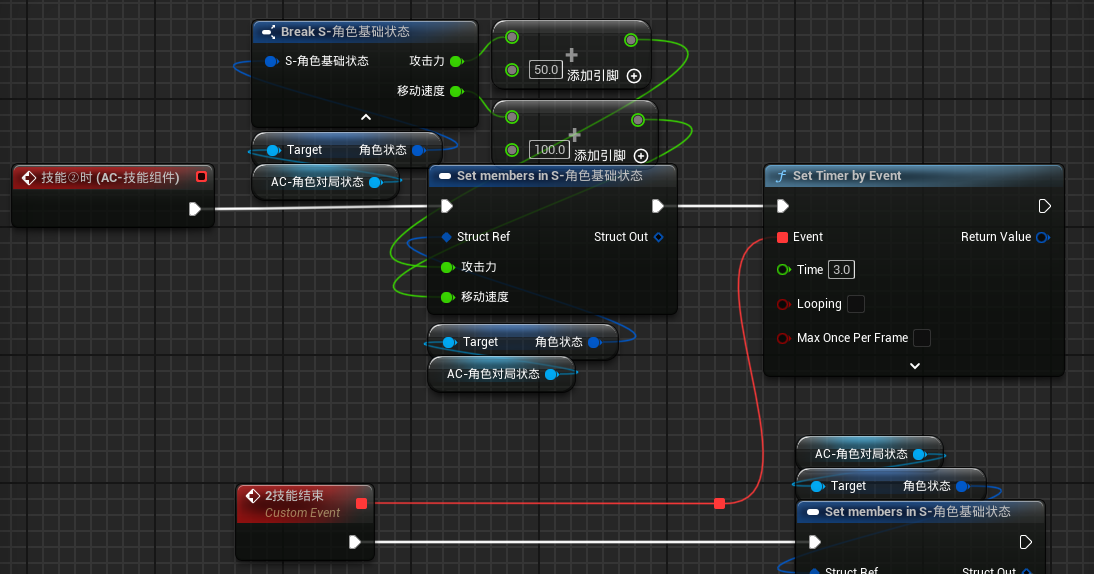
图示是可穿透Pawn的快移。

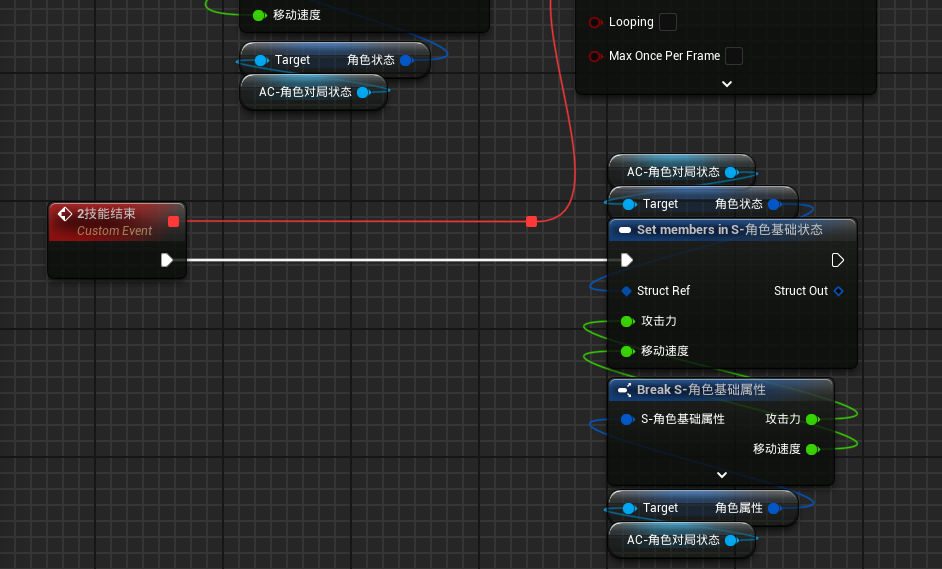
为了和闪现技能同步，我们改一下快移技能的参数。

闪现技能的参数“位移距离”，每100个位移距离代表闪现1格。所以我们将快移技能的参数乘以以24，这样快移技能也是每100个单位移动一格。

更新：角色②技能更新

用SetTimerByEvent实现





重新实现应用伤害功能

现在应用伤害时需要计算护甲、甲穿等。因为该游戏的护甲穿透遵循百分比数值，计算公式为

（无甲穿）

实际伤害=面板伤害\*（1-护甲/(面板伤害+护甲)）

（考虑穿甲）

实际护甲穿透=1-（1-护甲穿透百分比1）\*（1-护甲穿透百分比2）

计算护甲=实际护甲\*（1-实际护甲穿透）=实际护甲\*（1-护甲穿透百分比1）\*（1-护甲穿透百分比2）

实际伤害=面板伤害\*（1-计算护甲/(面板伤害+计算护甲)）

法穿同理。

现在来实现应用伤害函数。

对于转化/绝对/真实伤害，则有不同的计算方法。

转化伤害是基于此次伤害的数值，输入到“应用伤害”函数中，计算可能的物理伤害和魔法伤害的实际伤害，取最大值作为实际伤害。

真实伤害是直接从当前生命值处扣除。不会给护盾造成伤害。

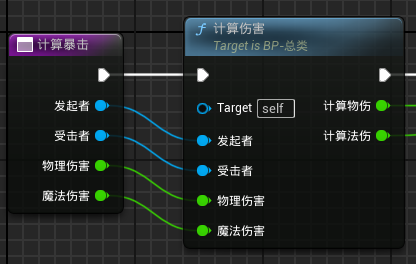
绝对伤害是忽视双抗和双穿直接造成伤害。会给护盾造成伤害。

同时也要考虑到暴击、生命偷取、全能吸血、生命恢复强度

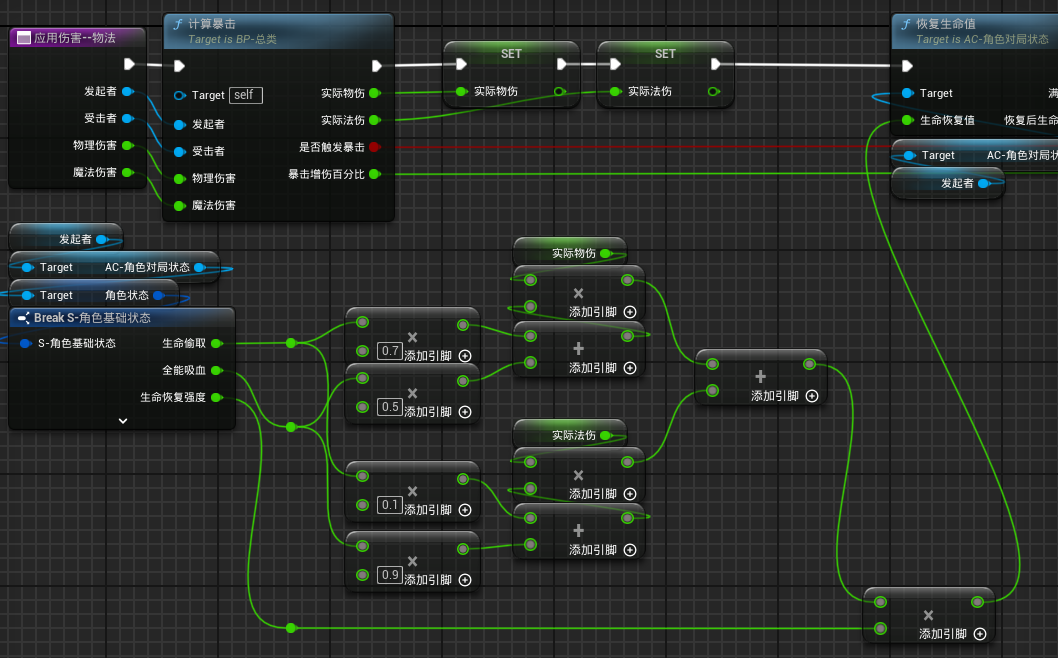
首先实现计算伤害函数

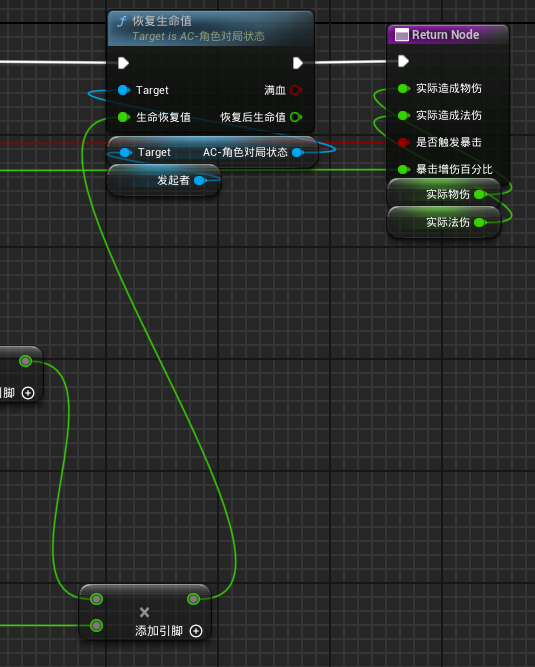


然后实现计算暴击函数



然后在应用伤害中实现生命偷取等效果。





设定：所有的生命恢复效果都要基于生命恢复强度来体现，默认值为1，装备提供的生命恢复强度的数值体现为百分比。重伤是应用在生命恢复强度上的。

然后在该应用伤害的地方应用伤害。

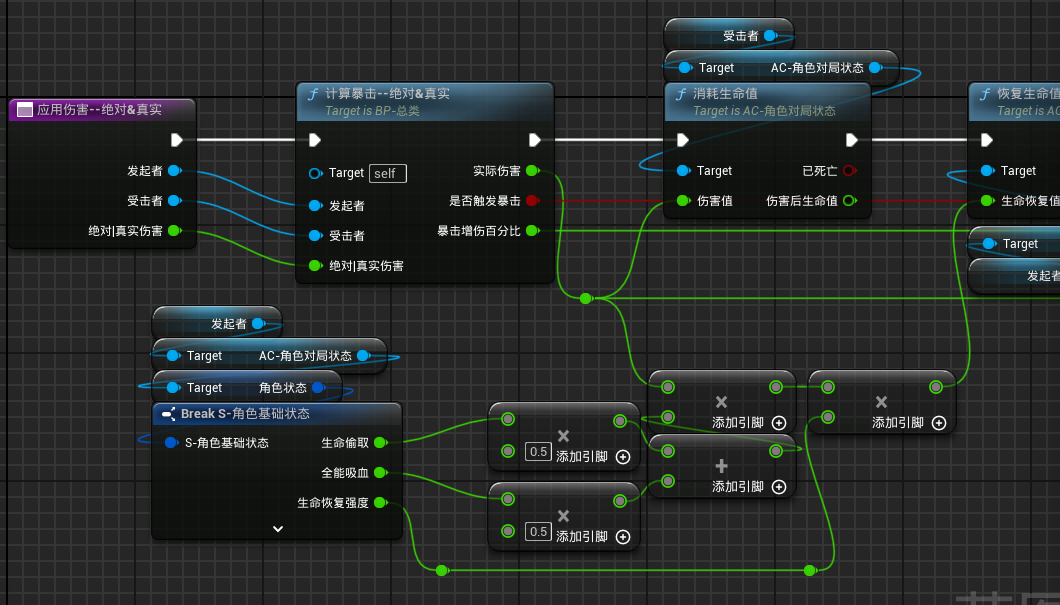
我们也可以实现一下转化、绝对、真实伤害。

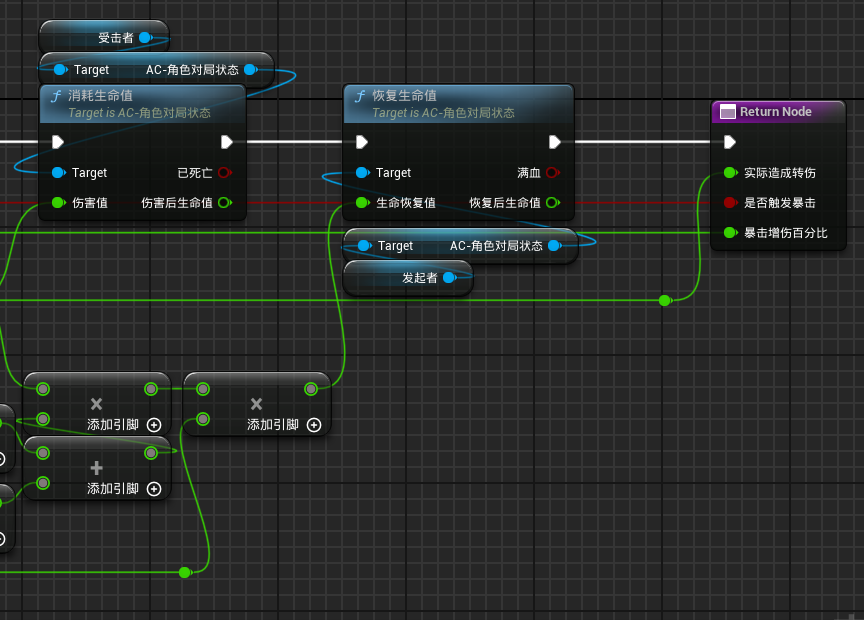
因为护盾机制并没有创造，所以先不实现绝对伤害。

实现转化伤害。



实现真实伤害。





按照上述的思路，在英雄的基础属性表中的对应数值，应当是小数形式，而不是整数形式（如穿透应该是0.1，代表10%，而不是10。吸血同理）

英雄的生命恢复强度应当设置到默认为1。

暴击伤害保留50，代表50%。

接下来实现护盾

护盾思路：给角色挂一个假人，这个假人实现护盾生命值的更迭。

实现英雄：灺

技术突破尝试：生成的发射物不在其他蓝图中实现，只在角色蓝图中实现。

实现护盾效果。

实现近程攻击。

实现眩晕效果与击飞效果。

实现蓄力释放技能