**KBE for UE4使用说明**

目录

[一、 登入登出 2](#_Toc454541330)

[二、 事件处理 4](#_Toc454541331)

[三、 Entity的进入和离开 7](#_Toc454541332)

[四、 Entity的位置同步 9](#_Toc454541333)

[五、 Entity的属性同步 11](#_Toc454541334)

[六、 Entity的远程方法 13](#_Toc454541335)

[七、 Player的位置同步 15](#_Toc454541336)

[八、 注意事项 16](#_Toc454541337)

## 登入登出

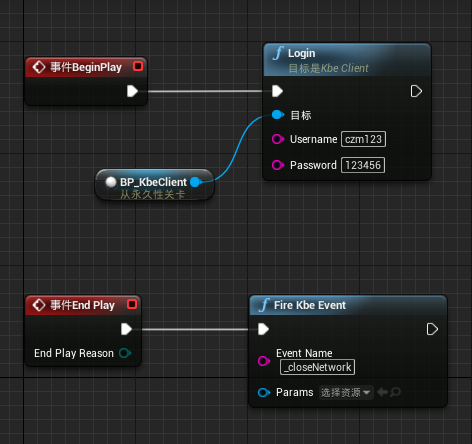
1. 使用默认的Pawn类型（避免在登录游戏之前出现角色）。



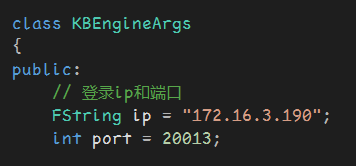
1. 将BP\_KbeClient蓝图拖入场景，创建一个该类型的实例。

1. 打开关卡蓝图，分别在BeginPlay事件中请求登录，及EndPlay事件中关闭网络连接：



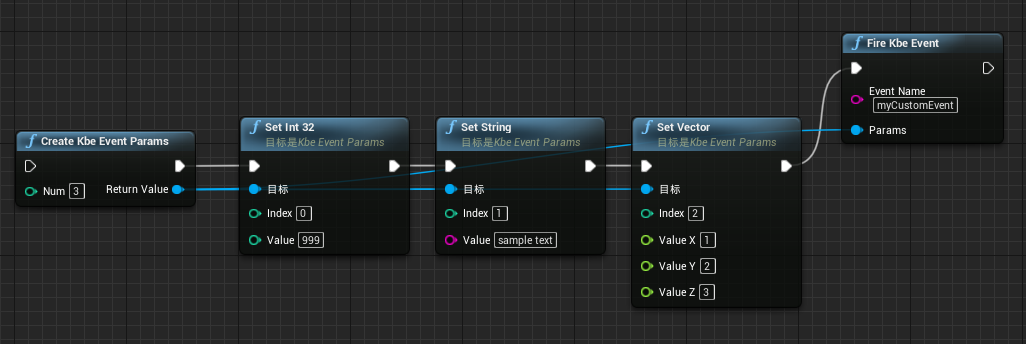
1. 目前，服务器的ip和端口在KBEngineArgs.h中定义：



## 事件处理

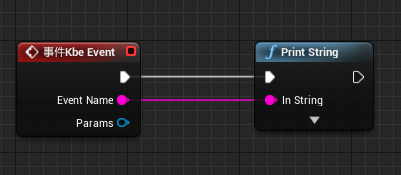
按事件的发起方进行分类，可以为：FireIn（UE4 to KBE）和FireOut（KBE to UE4）两种。

上个章节中，Login和FireKbeEvent属于FireIn类型，也就是从UE4层主动传达到KBE层的事件。当需要传递事件参数时可参考：

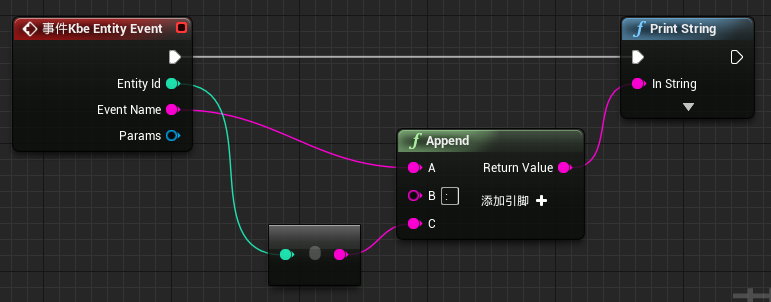


（创建一个包含3个参数的“KbeEventParams”对象，并分别为每个参数进行赋值，然后将它传递到“myCustomEvent”中）

所有FireOut类事件从KBE层发出，转由BP\_KbeClient蓝图对象进行处理（比如登录结果、Entity的属性更新等）。打开BP\_KbeClient蓝图，可以看到：

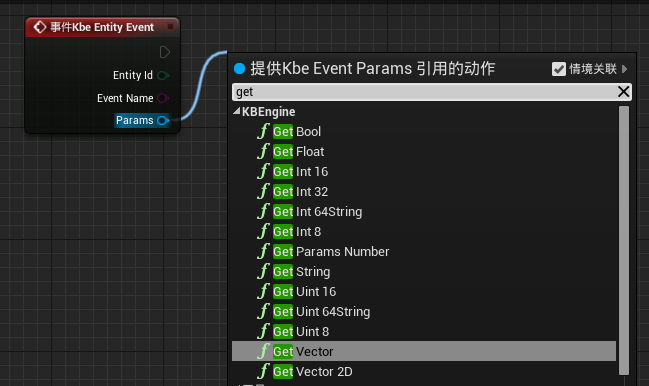


（全局性事件）



（Entity事件）

其中Params对象包含了该事件所关联的参数集，可以通过GetParamsNumber()获得参数的总个数，Get<Type>()获得指定参数的值。



如需发出自定义的事件到蓝图，可以调用：

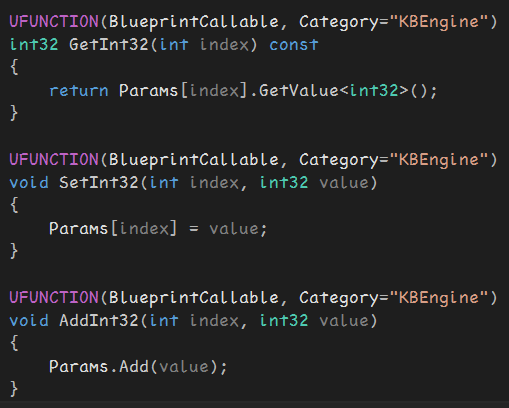
C:\Users\new\AppData\Roaming\Tencent\Users\514485685\QQ\WinTemp\RichOle\[YQXW2)PM~WR7T4Z4O5C]7C.png

或：

C:\Users\new\AppData\Roaming\Tencent\Users\514485685\QQ\WinTemp\RichOle\80%`85I1ZI65A2L]9OEPQS8.png

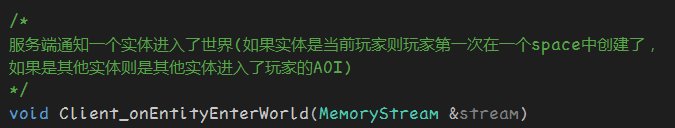
这将触发对应的“KbeEvent”和“KbeEntityEvent”。

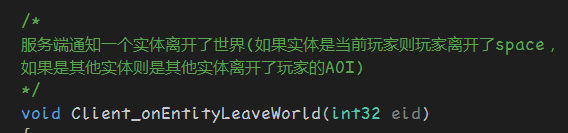
如需扩充KbeEventParams类型，使其支持更多的参数种类，可以参考KbeEventParams.h中的代码，添加Get<MyCustomType>()、Set<MyCustomType>()、Add<MyCustomType>()：



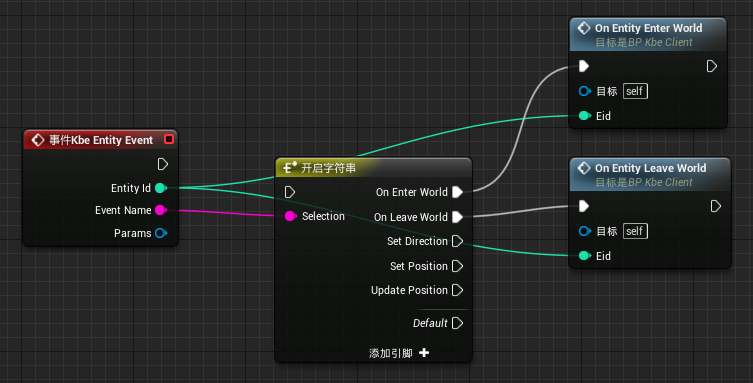
## Entity的进入和离开

当Entity（玩家或怪物）进入/离开（AOI）时，将触发客户端的EnterWorld/LeaveWorld事件。

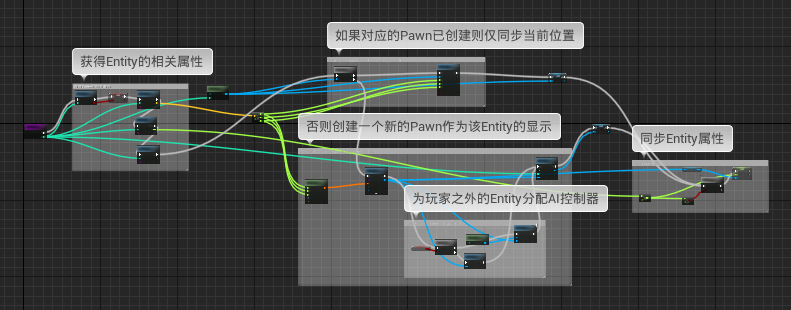




C++中的代码做好对应的处理后，将继而触发UE4蓝图中的“KbeEntityEvent”事件，在这里做相关的客户端表现：

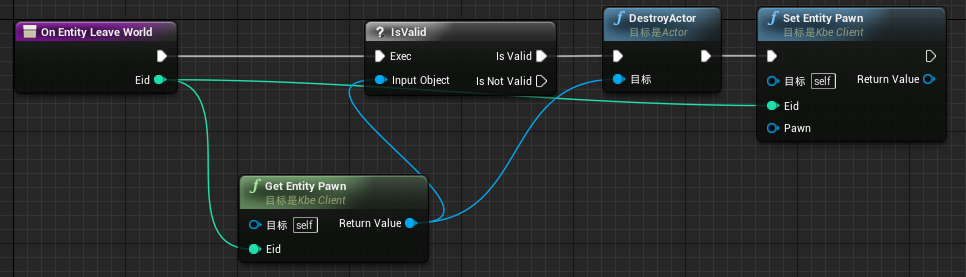


在Demo中，OnEntityEnterWorld中做了以下处理：

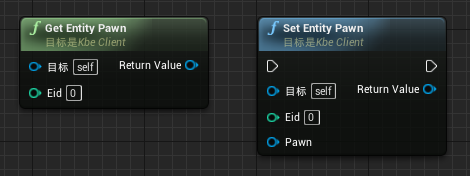


（主要是创建一个相应的Pawn作为Entity的显示）

而OnEntityLeaveWorld中则简单地将该Entity对应的Pawn销毁掉：



在BP\_KbeClient中，会记录Entity（KBE）和Pawn（UE4）的对应关系，通过一对函数进行操作：

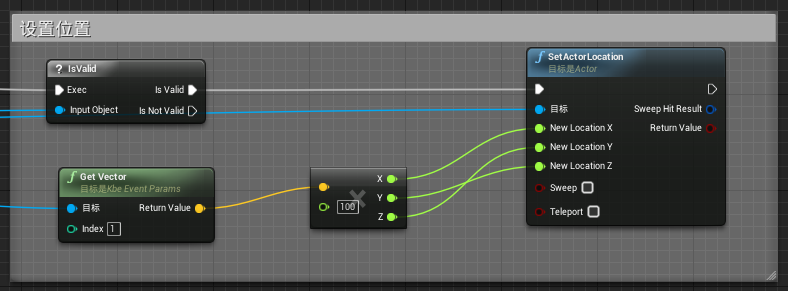


## Entity的位置同步



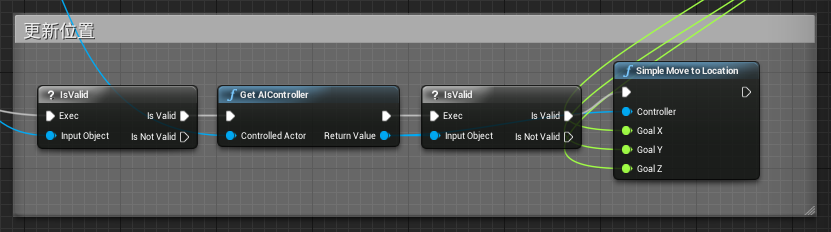
Demo中通过处理SetDirection（由于UE4的MoveTo也会控制角色的方向，暂时可以不做处理）、SetPosition、UpdatePosition事件来同步Entity的位置和朝向信息。

先看下SetPosition的处理方式：



（简单地将对应的Entity放到指定位置）

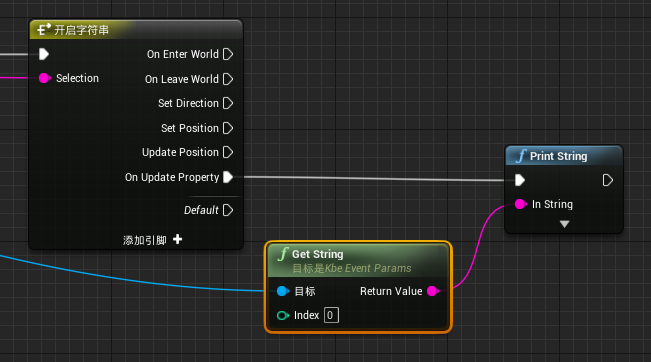
而UpdatePosition就需要通过MoveTo来做角色的移动过程：



(场景中需要设有导航区域)

## Entity的属性同步

当服务器更改Entity的属性，并将其同步到客户端之后，将会触发UE4蓝图中的“onUpdateProperty”事件：

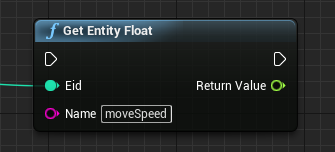


关联的Params中包含三个参数，依次是：

* PropertyName（String）
* NewValue（entitydef中定义的类型）
* OldValue（同上）

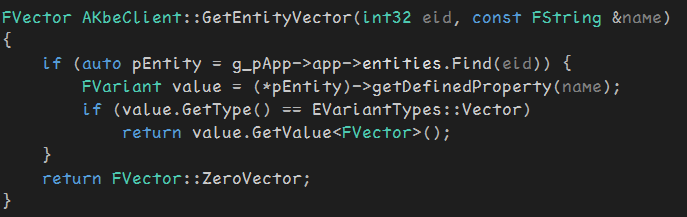
如需主动访问特定的Entity属性值，则可以通过BP\_KbeClient对象的GetEntity<Type>()系列接口获得：





（获得指定Entity的“moveSpeed”属性）

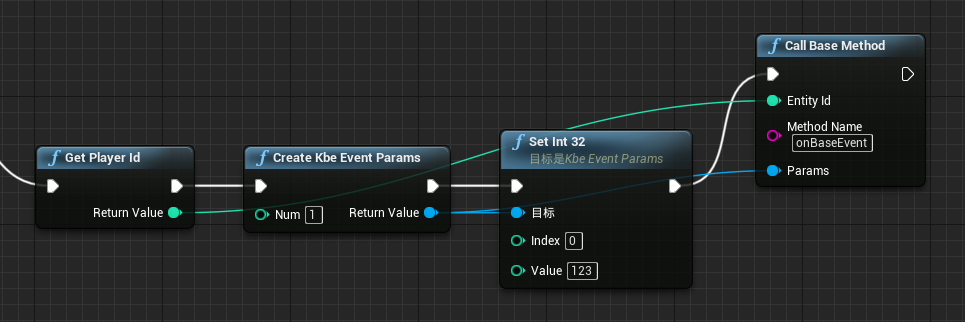
如需扩充更多的属性类型，请参考KbeClient.h/KbeClient.cpp中相关代码进行添加：



## Entity的远程方法

###### Client -> Base

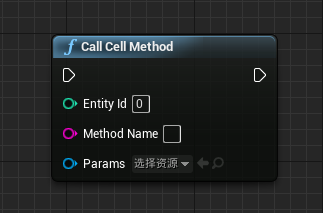
可以通过以下方式从Client端调用Base上的方法：



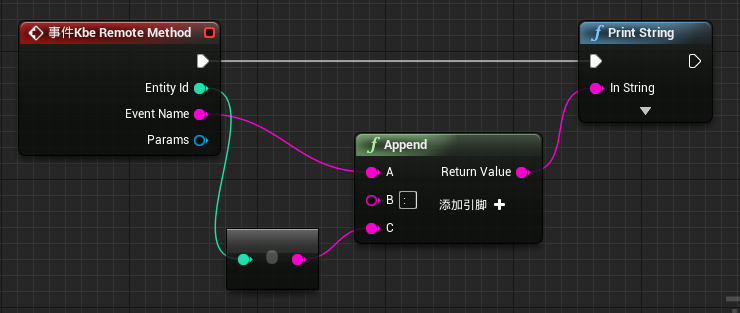
（调用Base端的onBaseEvent方法）

###### Client -> Cell

Cell端的方法调用则通过：

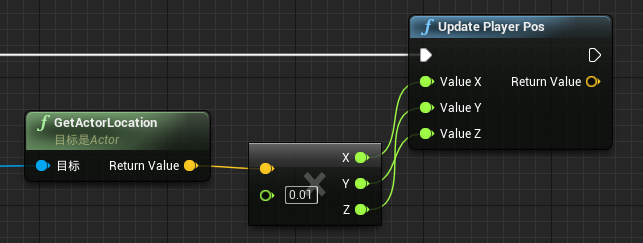


###### Remote -> Client



在Base或Cell中对Client的Entity方法调用，会触发蓝图中的“KbeRemoteMethod”事件，请按需进行处理。

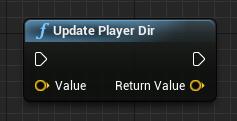
## Player的位置同步



(如果服务判定为非法移动，可以调用SetPosition()方法，将玩家传送回之前的位置)

Demo的玩家位置同步功能，在BP\_MainGameController的“Tick”事件中实现。

其工作原理是当侦测到玩家位置发生变化时，调用UpdatePlayerPos()方法来提交位置信息，类似的方法还有UpdatePlayerDir()：



## 注意事项

* UE4与KBE的单位不一致，UE4是厘米，而KBE是米。
* UE4与KBE的向量组成不一致，y轴和z轴位置相反。
* UE4的蓝图不支持无符号整形、64位整形、双精度浮点数等基本类型。如有必要，需额外处理。