# UE4: Event Dispatcher

[目前接触到的Version: UE 4.16.2]

## 一.简介

UE4 中Event Dispatcher是非常强大的功能，非常适合在各个蓝图之间实现通信。通过将一个或多个事件绑定到事件调度程序，可以在调用事件调度程序后使所有这些事件触发。这些事件可以在Blueprint Class中绑定，但是Event Dispatcher也可以在Level Blueprint中触发事件。

## 二.原理

Event Dispatcher其实就是C++（C#）中Delegate机制。但是Delegate机制的涵盖范围比事件调度器要大一些（C#中有delegate关键词，可以用来实现delegate机制； event关键词，对应的就是Event Dispatcher）。在C# 中，delegate是一种类型，它实际上是一个能够持有对某个方法的引用的类。与其它的类不同，delegate类能够拥有一个签名（signature），并且它"只能持有与它的签名相匹配的方法的引用"。

Delegate所实现的功能类似于C++中的函数指针。它允许将一个方法作为变量传递给一个对象，使得这个对象能够调用该方法。但是delegate与函数指针有很大不同，具备更多的优点：

（1）函数指针只能指向静态函数，而delegate既可以引用静态函数，又可以引用非静态成员函数。

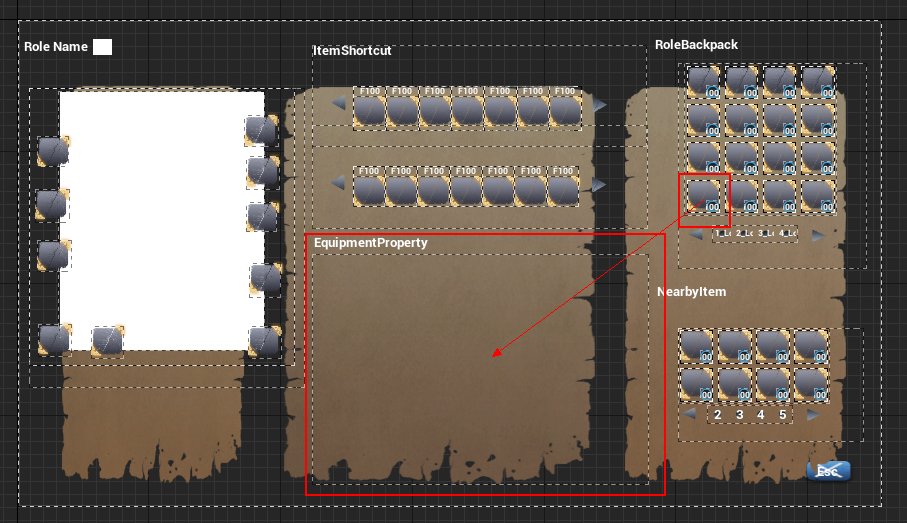
（2）在引用非静态成员函数时，delegate不但保存了对此函数入口指针的引用，而且还保存了调用此函数的类实例的引用。

（3）与函数指针相比，delegate是面向对象、类型安全、可靠的受控（managed）对象。

Delegate机制是一种常见的设计模式。 Event是一种特殊的delegate实例，是受限制的委托，是委托的一种特殊应用，只能在类的外部施加+=,-=操作符，二者本质上是一个东西。

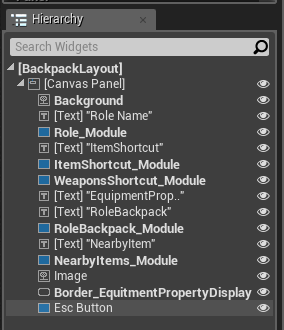
UE4种 delegate按照类型，分为单播代理和多播代理。其中，多播代理与事件调度器的功能类似，而单播代理更接近单纯的代理，也就是只能绑定一个执行函数。要让多播代理像是事件调度器一样使用的话，需要将其定义为Dynamic。对于单播代理也是一样的。

## 三.蓝图间通信

我们要实现的其中一个功能是在角色背包每个背包物品被点击之后，在装备属性区域显示相应的装备有关的信息。如图一所示：  


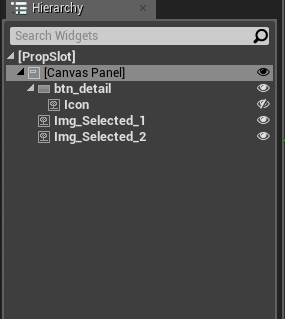
图一，UI界面

整个界面的模块组成为：



图二，UI界面模块组成

角色背包模块里面利用UniformGridPanel控件来显示每一个物品，其中最底层的物品槽模块，控件组成为：



图三，装备槽模块组成

整个界面架构组成：

这种结构决定，在btn\_detail 点击之后，需要将该点击事件传递到UI界面最顶层，并传递所需要的装备ID等一些信息。这时候就可以采用Event Dispatcher来进行消息传递。

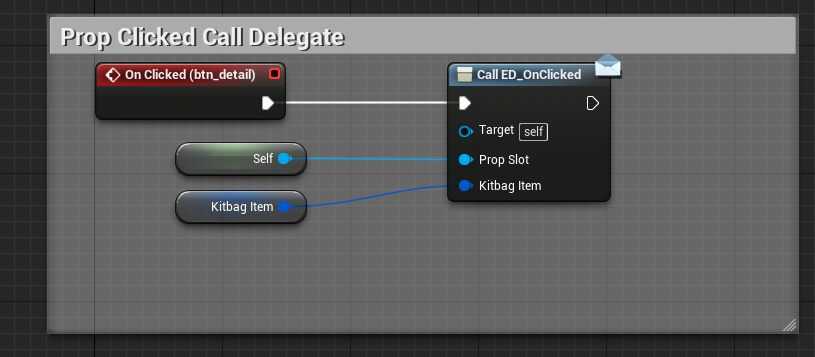
主要分为四步

1.首先在最底层装备槽Blueprint 中定义一个Event Dispatcher：



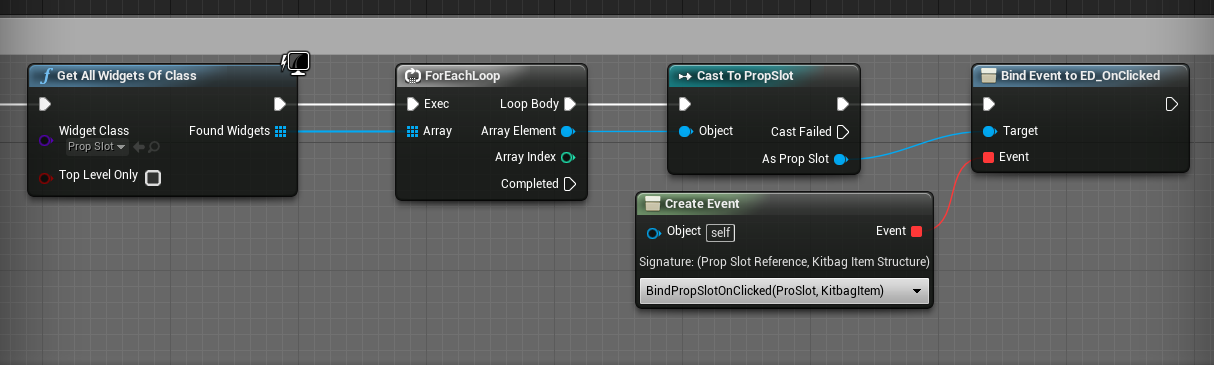
图四，装备槽中Event Dispatcher

2.设置在Button点击之后，Call该Event Dispatcher：



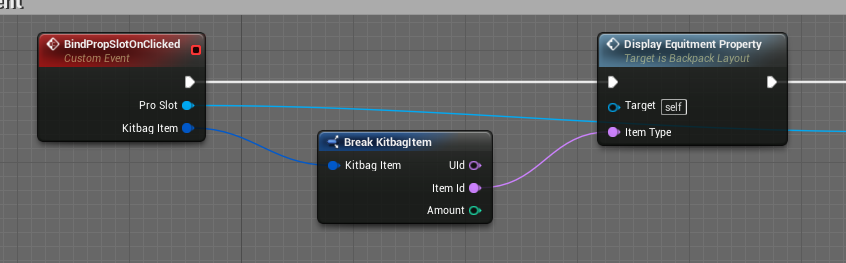
图五，调用Event Dispatcher

3.在最顶层UI中设置绑定事件：



图六，绑定处理事件

4.设置绑定的事件要做数据处理：



图七，对该事件所做的处理