# 事件调度器（Dispatcher）

也叫做分发器。操作的目的是，一点执行调度器调用，N点可以响应调用逻辑。与设计模式中的观察者模式相同。

**观察者模式：**对象之间存在一对多的关系，在构建时，通过被观察者主动执行操作，观察者随之响应的一种软件设计模式。

### 事件调度器构建

调度器只有输入参数列表，主要原因是由于调度器需要绑定到事件节点，事件节点是无返回值的，所以我们只需提供输入参数值即可。

注意：谁希望知道某个人的某个事情是否发生，那么就由希望知道的人发起绑定。动作响应在发起人的身上。

调度器的优势，可以不用关心希望接受目标对象类型，可以广播数据给不同类型的对象。

# 调试

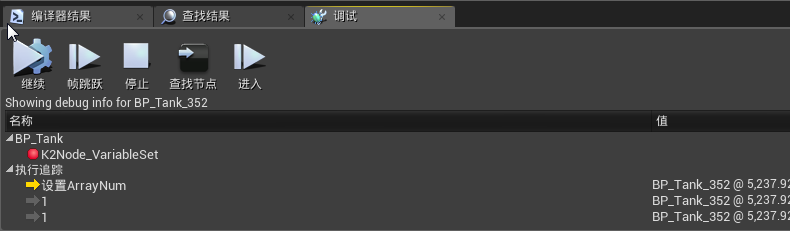
将程序执行停止到某一个逻辑点，通过观察堆栈数据信息，逐步运行程序，以推演程序的执行流程，达到排查错误的目的。调试主要为了修正语法和逻辑错误。可以有效的保证程序的稳定性和正确性。

注意：调试过程是不可以逆转的，无法进行后退。

### 断点

用于打断程序执行流程，将程序锁住到标记点处

### 执行堆栈



用于观察程序执行顺序，和程序执行过程中数据的变化情况（数据需要进行观察标记）

### 观察标记



右键你的属性节点，均有Watch this value

### 调试工具按钮说明



* 继续 Resume：用于跳过当前断点，进入到下一个执行断点，如果没有下一个执行断点，则直接继续运行游戏
* 帧跳跃 Frame Skip：与进入相似，就是逐步骤调试程序，调试逻辑完成后，将游戏的运行模式改为逐帧运行
* 停止：直接终止游戏
* 查找节点：快速跳跃到当前调试节点
* 进入：逐步运行程序节点（F10）
* 调试过滤器：用于标记当前调试对象目标是谁