70 Conditional and Action Breakpoints in C++

本讲内容是一个简单的VS开发和调试的技巧，不过不仅仅是断点，而是关于条件与操作应用在断点上。

1. 条件断点 Condition

通过条件或条件断点，我们可以告诉调试器想在这里放置一个断点，但我希望断点在特定条件下触发，比如内存中的某些东西满足了条件就触发这个断点。

2. 操作断点 Action

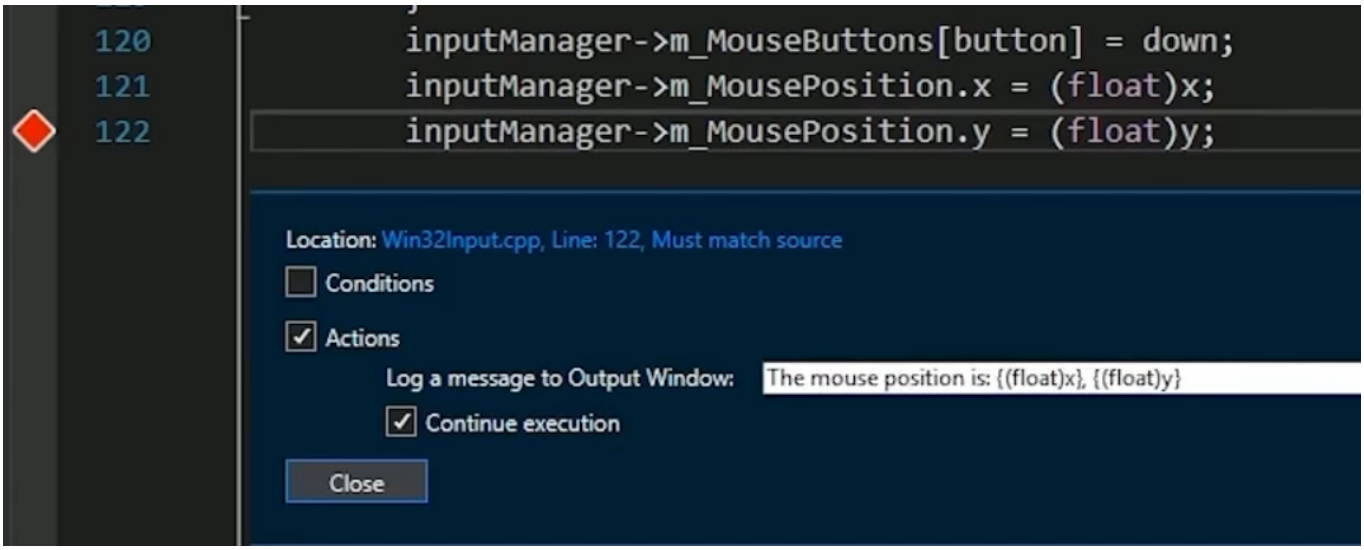
操作断点是允许我们采取某种动作，一般是在碰到断点时打印一些东西到控制台。

这里有两种类型的操作断点：

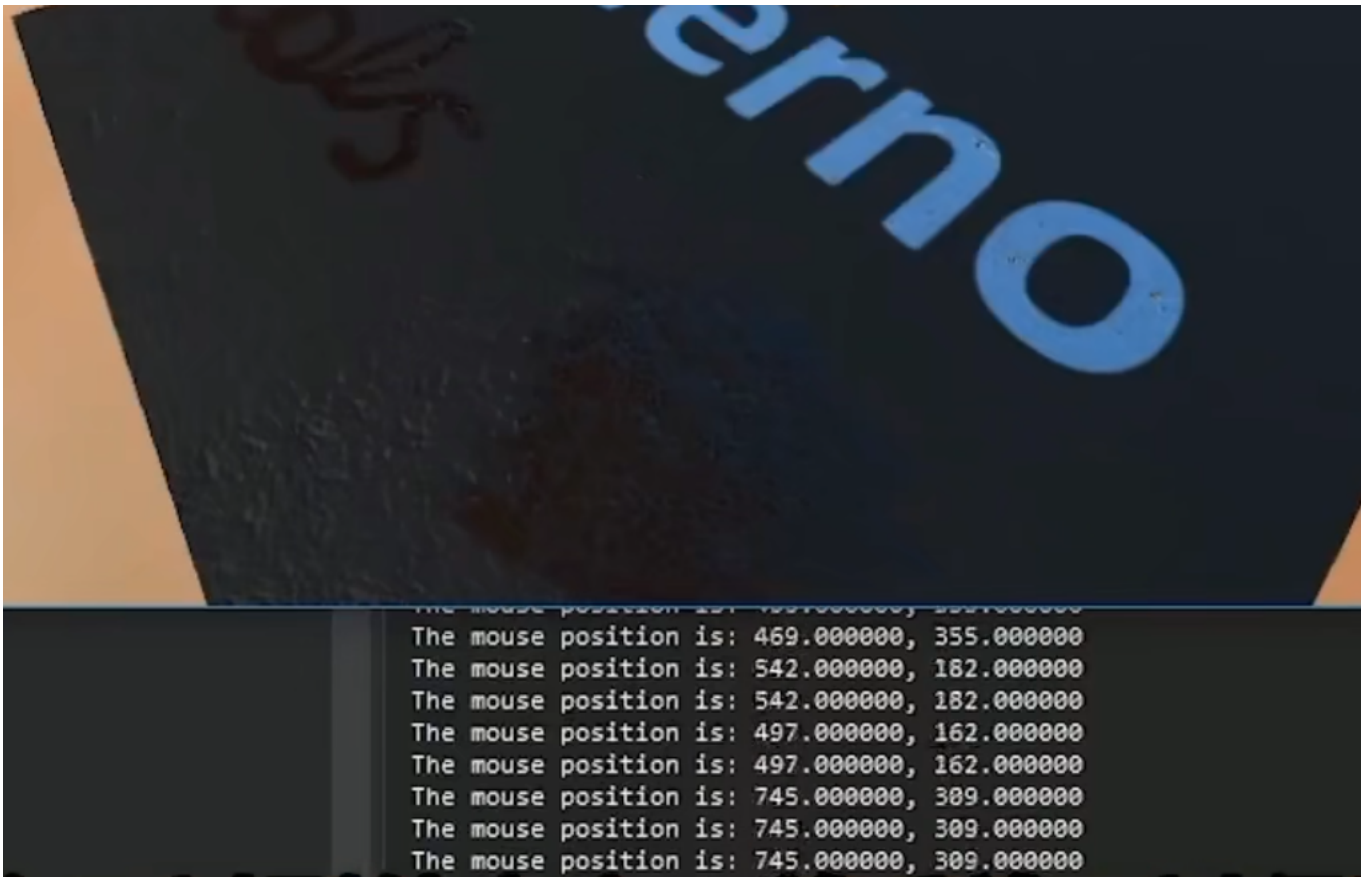
- 一是让你在打印你想要的东西时继续执行，比如你想记录鼠标位置，每次鼠标移动，移动事件（打印鼠标位置）就会发生，可以让那个断点打印一些东西到控制台但保持程序运行；
- 二是打印一些东西，但仍然中断程序，暂停程序的执行，这样我们就可以检查内存中的其它东西。

3. 现实中的用法

右键设置的断点，选择Conditions或者Actions（可以一起使用，打开哪个都可以设置另一个），这里先测试 *Action breakpoint*：

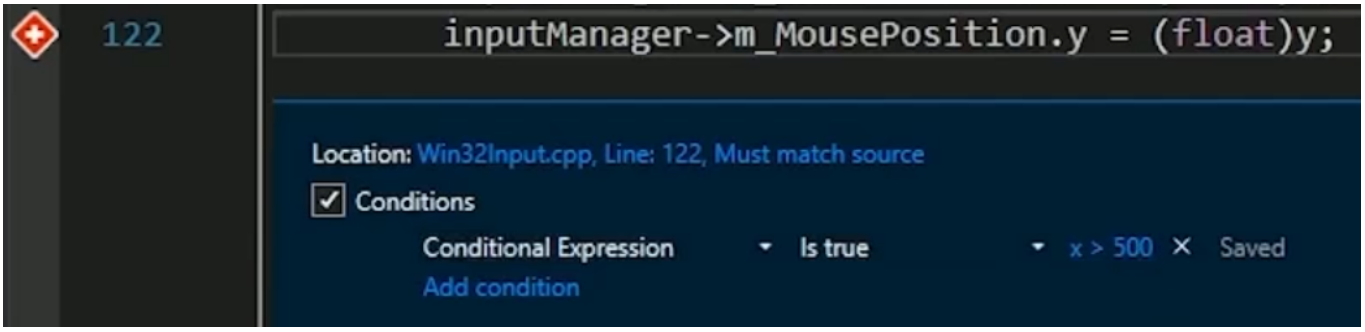


要将变量名写在`{}`内，还可以做类型转换或者加一些文本（如示例）



你当然也可以通过在源代码中添加代码的方式来打印这些内容，但这里最酷的事情是，你**不需要终止你的应用，不需要重新编译代码**，只是在初始引用程序运行时设法打印了额外的数据到控制台（Output Window），提升了效率。
另一种方式就是打普通断点，然后看中断的程序再找变量，这样很耗时。

再测试一下*Condition breakpoint*：



这里的条件可以是任何布尔语句。

条件断点也是非常有用的，比如你想在一个循环中找到一个特殊的条件对于普通断点来说是基本不可能的，比如只想在一个特定编号的敌人的节点（出问题的实体）那里中断程序，如果要手工完成你得一直按多次F5。尽管会降低性能，但是这只是一个调试工具罢了。