
Subject: Re: Focus problem

Posted by [Lance](#) on Sun, 19 Dec 2021 13:22:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Silvan:

The way I define WhenSuspiciousKey is a typical way to define an Event. First, the Ctrl or its direvative need to provide a chance for the event in a (relevant) overrided virtual function. In my case, I overrided

```
bool Key(...)
```

In my MainWindow::Key(...)

I allowed WhenSuspiciousKey be called at certain time. What is WhenSuspiciousKey? I define it as a member variable of MainWindow of type Event<>

By add this member variable to MainWindow, the user of MainWindow can assign a callback to an instance(object) of this type by

```
MainWindow m;  
m<<[]{};
```

There is no magic here. C have simlar (but much limited) mechanism. WhenSuspiciousKey is like a function pointer to which you can assign a callback. In order for this callback to actually be called, some where, you need to check and call it(in our case, it's in the MainWindow::Key(...) virtual function).

Now, Key is a virtual function, you cannot assign a callback to it. On the other hand, an object of MainWindow, m, has a public member variable name WhenSuspiciousKey, which is of type Event<>. You can access this member variable. One way is by something like
m.WhenSuspiciousKey<<[]{};

PS: On a second thought, use C function pointer as example:

```
void (*WhenWhat)(int);
```

```
void func1(int p)  
{  
    if(WhenWhat)  
        WhenWhat(p);  
}
```

```
void callback(int p)  
{  
}
```

```
WhenWhat=callback; // this is good
// just like
// WhenSuspiciousKey<<[]{};
// is fine
func1=callback; // this won't compile
// just like
// Key<<[]{};
// won't compile.
```
