
Subject: Re: Holding a Button down

Posted by [mrjt](#) on Sun, 05 Apr 2009 07:39:05 GMT

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

I can think of two ways.

1- Use the WhenRepeat callback:

```
CtrlLibTest::CtrlLibTest()
{
    CtrlLayout(*this, "Window title");
    btn <<= THISBACK(OnPushEnd);
    btn.WhenRepeat = THISBACK(OnRepeat);
    count = 0;
}
```

```
void CtrlLibTest::OnRepeat()
```

```
{
    ++count;
    label = AsString(count);
}
```

```
void CtrlLibTest::OnPushEnd()
```

```
{
    count = 0;
}
```

2- Create your own button type and add a new Callback:

```
struct HoldButton : public Button {
    Callback WhenPush;
    virtual void LeftDown(Point p, dword keyflags) {
        Button::LeftDown(p, keyflags);
        if (IsPush())
            WhenPush();
    }
};
```

The first method is nice because it doesn't require you to start a timer to determine how many 'ticks' the button has been pushed for, but you don't have much control over the repeat speed (you can set it globally but this might make other bits of the GUI work less well). There is also a delay. The second method will require a timer (SetTimeCallback) to count the ticks. The normal WhenAction callback is triggered on LeftUp so you can use that to determine when the user stops pushing the button.

Or you can combine both approaches for the best of both worlds (keeping the count internally in your new Ctrl class).

James
