Why do up-down controls have the arrows backwards?

devblogs.microsoft.com/oldnewthing/20051222-12

Raymond Chen

When you create an up-down control (some people call a "spinner" control) in its default configuration, the up-arrow decrements the value and the down-arrow increments it. Most people expect the up-arrow to increment and the down-arrow to decrement. Why is it backwards?

The up-down control is a victim of Windows' reversed y-axis.

Mathematically, the (0, 0) coordinate should be at the bottom left corner of the screen (and in fact that's where OS/2 puts it), with y increasing as you move up the screen. Windows, on the other hand, puts the (0, 0) coordinate at the **upper** left corner, with y increasing as you move **down** the screen.

What does that have to do with anything?

The up-down control can be positioned horizontally or vertically. Let's first look at the less problematic horizontal configuration. Windows and mathematics agree that the x coordinate increases to the right, and the right-arrow (the arrow with higher x-coordinate) causes the value to increase. (Let's leave right-to-left languages out of the picture for the purpose of this discussion.)

Once you have the horizontal version of the control working, converting it to a vertical control is a simple matter of interchanging the x– and y-axes.

That's why the up-arrow decreases the value. The up-arrow points towards smaller y-coordinates and consequently decrements the value.

It's perfectly logical and simultaneously counter-intuitive. (It's slightly more intuitive if you imagine the value attached to the up-down control as controlling the y-coordinate of an imaginary object on the screen. In that case, clicking the up-arrow causes the y-coordinate to decrease and the object moves up the screen.)

Fortunately, this wackiness doesn't last long, because the moment you change the range of the up-down control from the (not very useful) default of 0–100 to whatever you actually need, the arrows behave "normally" again.

Perhaps intuitiveness should have won out over logic. But what's done is done, and, as noted above, the problem goes away soon enough.

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

Follow