

Why do user interface actions tend to occur on the release, not on the press?

devblogs.microsoft.com/oldnewthing/20061216-13

December 16, 2006



Raymond Chen

If you pay close attention, you'll notice that most user interface actions tend to occur on the release, not on the press. When you click on a button, the action occurs when the mouse button is released. When you press the Windows key, the Start menu pops up when you release it. When you tap the Alt key, the menu becomes active when you release it. (There are exceptions to this general principle, of course, typing being the most notable one.) Why do most actions wait for the release? For one thing, waiting for the completion of a mouse action means that you create the opportunity for the user to cancel it. For example, if you click the mouse while it is over a button (a radio button, push button, or check box), then drag the mouse off the control, the click is cancelled. But a more important reason for waiting for the press is to ensure that the press won't get confused with the action itself. For example, suppose you are in mode where objects disappear when the user clicks on them. For example, it might be a customization dialog, with two columns, one showing available objects and another showing objects in use. Clicking on an available object moves it to the list of in-use objects and vice versa. Now, suppose you acted on the click rather than the release. When the mouse button goes down while the mouse is over on an item, you remove it from the list and add it to the opposite list. This *moves the items the user clicked on*, so that the item beneath the mouse is now some other item that moved into the original item's position. And then the mouse button is released, and you get a `WM_LBUTTONDOWN` message for the new item. Now you have two problems: First, the item the user clicked on got a `WM_LBUTTONDOWN` and no corresponding `WM_LBUTTONUP`, and second, the new item got a `WM_LBUTTONUP` with no corresponding `WM_LBUTTONDOWN`. You can also get into a similar situation with the keyboard, though it takes more work. For example, if you display a dialog box while the Alt key is still pressed rather than waiting for the release, the Alt key may autorepeat and end up delivered to the dialog box. This prevents the dialog box from appearing since it's stuck in menu mode that was initiated by the Alt key, and it's waiting for you to finish your menu operation before it will display itself. Now, this type of mismatch situation is not often a problem, but when it does cause a problem, it's typically a pretty nasty one. This is particularly true if you're using some sort of windowless framework that tries to associate mouse and keyboard events with the corresponding windowless objects. When the ups and downs get out of sync, things can get mighty confusing.

(This entry was posted late because a windstorm knocked out power to the entire Seattle area. My house still doesn't have electricity.)

Raymond Chen

Follow

