
Subject: Re: Lack of KeyMods parameter in the ::MouseUp function call -> Feature request

Posted by chris_torrence@NOSPAM on Wed, 06 Jan 2016 16:31:34 GMT

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

On Tuesday, January 5, 2016 at 2:20:35 PM UTC-7, Helder wrote:

> On Tuesday, January 5, 2016 at 9:16:07 PM UTC, Chris Torrence wrote:

>> On Wednesday, December 30, 2015 at 4:25:08 PM UTC-7, Helder wrote:

>>> Hi,

>>> I'm intensively using the FG event handler class. In doing so, I've come to a problem where I would like the ::MouseUp function to include information about the current keyMods. Currently, the functions ::MouseDown, ::MouseMove and ::MouseWheel have it...

>>>

>>> Let me explain why I think this is useful.

>>> I have a polygon on an image. I would like to add a point to the polygon. To distinguish from the translation case, I use the "alt" key to identify modifications (or alt-erations) to the polygon. Therefore, pressing the alt-key and the left mouse button results in an added point. Since I'm not always good with my hand, I would like to be able to:

>>>

>>> 1) press down alt-key

>>> 2) press down left mouse button (a point is added)

>>> 3) move mouse (and see the added point moving with the mouse)

>>> 4) release the left mouse button

>>> 5) release the alt-key

>>>

>>> Besides being useful, it is also "cool".

>>>

>>> However, if I would like to "cancel" the add point operation if the alt-key is released *before* the release of the left mouse button. In other words, the following sequence should delete the point after step 4):

>>>

>>> 1) press down alt-key

>>> 2) press down left mouse button (a point is added)

>>> 3) move mouse (and see the added point moving with the mouse)

>>> 4) release the alt-key (the previously added point is deleted)

>>> 5) release the left mouse button

>>>

>>> Since I cannot detect if alt is pressed when releasing the button, I cannot do this.

>>>

>>> Can this please be added? I don't think that it is that much of an effort...

>>>

>>> Cheers,

>>> Helder

>>

>> Hi Helder,

>>

>> It probably isn't too hard to add, but the problem is backwards compatibility. If we just add another argument then it will break existing event handlers. We can't add a keyword for the same

reason. There might be a way to check the number of accepted arguments and make the appropriate call, but then the documentation becomes ugly.

>>

>> So, I'll add it to the list of possible enhancements, but I wouldn't count on it making the cut.

>>

>> But keep making suggestions!

>>

>> -Chris

>

> Hi Chris,

> thanks for the answer. Just out of curiosity, what would be the problem of using a keyword? I thought that that would be the way to go to avoid backwards compatibility?

>

> Cheers,

> Helder

Hi Helder,

Well, if your routine doesn't currently accept keywords, and it's called with a keyword, then IDL will throw an error...

Cheers,

Chris