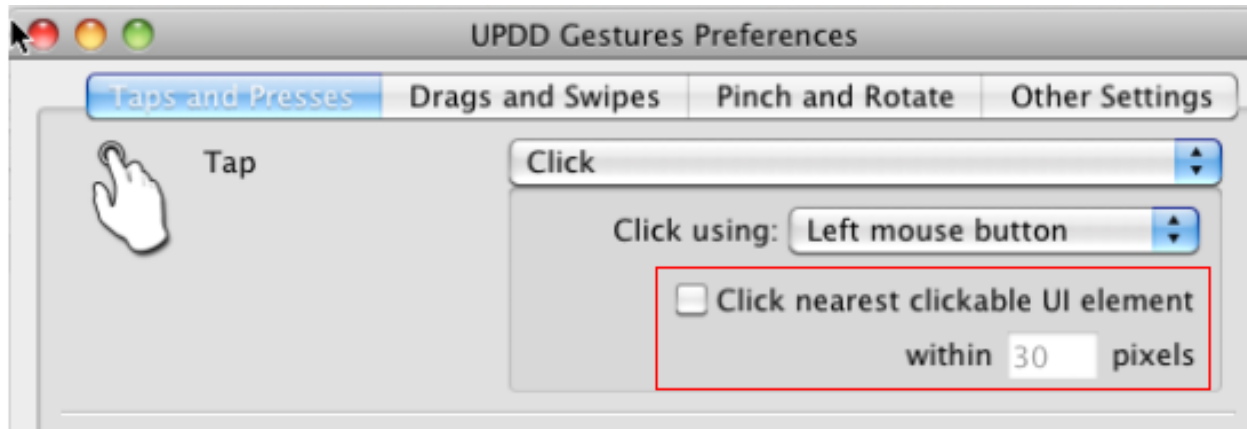


## Clickable UI element feature

Some MacOS controls are small and can be difficult to accurately select with a touch screen, especially when using a large stylus such as a finger, so we have introduced an option for each click action to locate the nearest clickable UI element within a defined pixel radius to the point of touch and click it.



Since gesture version 2.3.5 this is enabled by default as it is a very useful feature and improves touch usability, especially when dealing with small controls not designed for touch interface.

However, please note with this enabled the gesture software deliberately distorts tap and press touch locations that occur near a UI control element.

Should you wish to disable this feature uncheck the checkbox in the above dialog *or run the command `upddutils nodevice set "gesture action option tap click nearest" 0`*.

This option is very useful for selecting small UI controls difficult to accurately touch, such as the dialog close, minimise and maximise UI buttons, as shown, and similar desktop and application controls:



### Notes when using this feature:

1. This feature makes use of the Accessibility API and there are a few quirks when using this API interface which we have to code around.
2. This feature will activate any GUI button within a certain radius, even under the dialog

## Clickable UI element feature

being touched, and this behaviour is by design. The click nearest feature will consider every UI element within a radius of the touch, regardless of what window it is in and what window contains the touch. This caters for the situation whereby a click is required on a button near the edge of a window, like for example the close or minimize buttons. If the nearest UI element is restricted to only the window underneath of the actual touch, then touching just slightly above the top edge won't result in any of those buttons getting clicked, and it's pretty easy to miss. We understand that not restricting the search to within the touched window can sometimes yield undesirable results. Further, GUI buttons can be enabled or disabled and Gestures restricts clicks to enabled buttons only.

3. In our tests this feature really improved the touch experience and makes the selection of small buttons and UI elements a lot easier. However, this is an 'unproven' feature and will probably have a few eccentricities and issues to iron out as we have had to code in some decision making into Gestures in order for it to intelligently determine what it should click, when, and where. We suspect there's probably going to be a few edge cases where the wrong item gets clicked, or something doesn't get clicked that should. We would really appreciate any constructive feedback you may have when using this feature that may help us improve functionality in future releases.
4. We believe there are a few improvements we could make already which may come in future releases. Specifically, this feature doesn't work as well when browsing the web in Safari, as Safari reports practically every element in the browser window as being clickable, even when they aren't. We think we could improve its detection of 'real' clickable items to make it work properly in Safari, including finding text links.
5. We also briefly tested in Logic Pro X and found that it works partially as Logic Pro X supports the Accessibility API, but it reports odd information about some of its UI elements, similar to Safari. We think we could code round these issues and may do so in a future release.
6. UPDD Gestures is searching for the closest clickable UI element using the Accessibility API but in some cases there can be a considerable delay in some applications reporting the UI information. This was especially observed in the case with many games, including those created with Cocos2D-X. To cater for this we have added a timeout waiting for the information and by default it will timeout after 250 ms. This value can be changed to a smaller (or larger) value using the following Terminal command:  
  
upddutils nodevice set "gesture click nearest timeout" ms where ms is the new timeout value in milliseconds.
7. If searching for a nearest GUI element on an application that is totally unresponsive or frozen (perhaps because it crashed), then the request will never be fulfilled. Apple's Accessibility API gives up after six seconds, but in that time UPDD Gestures will be frozen and unresponsive.

## Clickable UI element feature

8. Finally, this feature won't readily work in any applications that do not support Apple's accessibility API, such as Chrome, Firefox, and Qt applications build using a Qt version prior to 5.2. It'll only recognize a small subset of their UI elements, like the menu bar.

Touch-Base Support

<http://support.touch-base.com/Documentation/50183/Clickable-UI-element-feat...>