

Ignoring clicks on recently-moved iframes

ajuma@chromium.org

May 2016

Background

This document considers the problem of [iframes that move suddenly](#) in an attempt to get users to click on them when they really intend to click on something else, leading to a bad user experience. This is mostly a problem for iframes containing ads.

Web authors might not be sufficiently incentivized to fix this problem themselves, since they benefit from clicks on ads. This means that addressing this problem may require user agents to intervene. One way to do this is to ignore clicks on iframes that have moved so recently that it's unlikely that a user could have intended to click on them. The purpose of this document is to turn this idea into a concrete proposed intervention. For Chrome-specific implementation ideas, see [this doc](#).

Which iframes?

Ads are usually contained in cross-origin iframes. Since interventions should be as narrowly-targeted as possible, the initial intervention should be limited to cross-origin iframes. This approach may help avoid breaking legitimate use cases for clicking on recently-moved iframes (e.g., games).

Identifying movement

There are several ways that an iframe's position on screen can change. These include main-thread-driven animations of properties like transform, top, and left, compositor-driven transform animations, javascript-driven changes to scroll position, and user-initiated scrolling. With the exception of user-initiated scrolling, all of these can be used by a page to quickly move an iframe over a short period of time, so any intervention that doesn't cover all of these will be easy to work around.

That leaves user-initiated scrolling. Obviously, movement caused by the user isn't intended to deceive the user. But from an implementation standpoint, it might be cleaner to treat this as any other kind of movement. One way to mitigate the impact of this is to allow a click on an iframe that has recently moved if the click would also have been inside the iframe's bounds without the recent movement, that is, if the iframe's position for the past X ms (for some X to be determined) also contains the click.

Identifying *recent* movement

At a high level, we want to identify movement that must have happened after the user formed the intent to click. That is, we want to make sure that when the user decided to click, the position they decided to click was inside the same iframe that they ultimately did click. This means approximating the minimum latency between a user's decision to click and the delivery of the click event. Experimenting with [this page](#), latency is roughly 200 ms in my testing, both when using a mouse and when using a touchscreen. However, at least one person found somewhat lower latency (between 110 ms and 170 ms), suggesting there's some variation caused by different device characteristics or by different human reaction time. Defining recent movement to be movement that happened less than 200 ms before the arrival of the click event seems like a reasonable starting point, but it may turn out that the definition of recent needs to be broadened (perhaps to something like 500 ms) to address most of the problematic cases, or reduced (perhaps to 100 ms) to avoid false positives.

Clicks that happen during fling scroll

It is likely that clicks that occur during a rapid fling on an "ad-sized" iframe will not be inside the iframe's position from 200 ms earlier. However, experimenting on Android, taps during a fling only inconsistently get treated as clicks; most of the time they instead simply stop the fling. Stopping the fling may, in fact, be the interpretation of the user's intent that's most likely to be correct. In other words, during a rapid fling, it's unlikely that a user will see an iframe enter the viewport, anticipate its position 200 ms in the future, and then tap on this anticipated position.

Using the union of multiple recent positions instead of the intersection

Instead of checking whether a click that's inside an iframe was also inside the iframe for the past 200 ms, what if we check whether it was inside the iframe at any point between 200 ms and, say, 500 ms earlier? This only makes a difference when the click was outside the iframe at some time less than 200 ms ago but was inside the iframe at some time t ms ago, where $200 < t \leq 500$. Since the click is also inside the iframe now, this means that the iframe [moved away from its current position and moved back](#) within the past 500 ms. Without good examples of use cases where users really do intend to click on such abruptly-moving iframes, using this approach doesn't seem justified.

The proposed intervention

Ignore clicks on cross-origin iframes when the location of the click would not have been within the bounds of the iframe for all draws that happened over the last 200 ms.

Measurement

Ideally, we would measure the percentage of false positives and false negatives in the click rejection logic. But of course, being able to precisely compute these values would imply having

logic to identify exactly which clicks we should be rejecting, and if we had such logic we would just use that directly in the intervention. Nevertheless, we can measure how often a rejected click is immediately followed by an accepted click on the same iframe (suggesting that the user did in fact intend to click on the iframe, so the first click should not have been rejected). Also, as a sanity check, we can measure the percentage of clicks on cross-origin iframes that get rejected, to make sure this is neither too close to zero (suggesting that the intervention is a no-op) nor too large (suggesting that the intervention is too broad). These measurements can be used to help fine-tune the intervention (e.g., to change the 200 ms threshold to something larger or smaller).

Implementation ideas

These are now in a [separate doc](#).