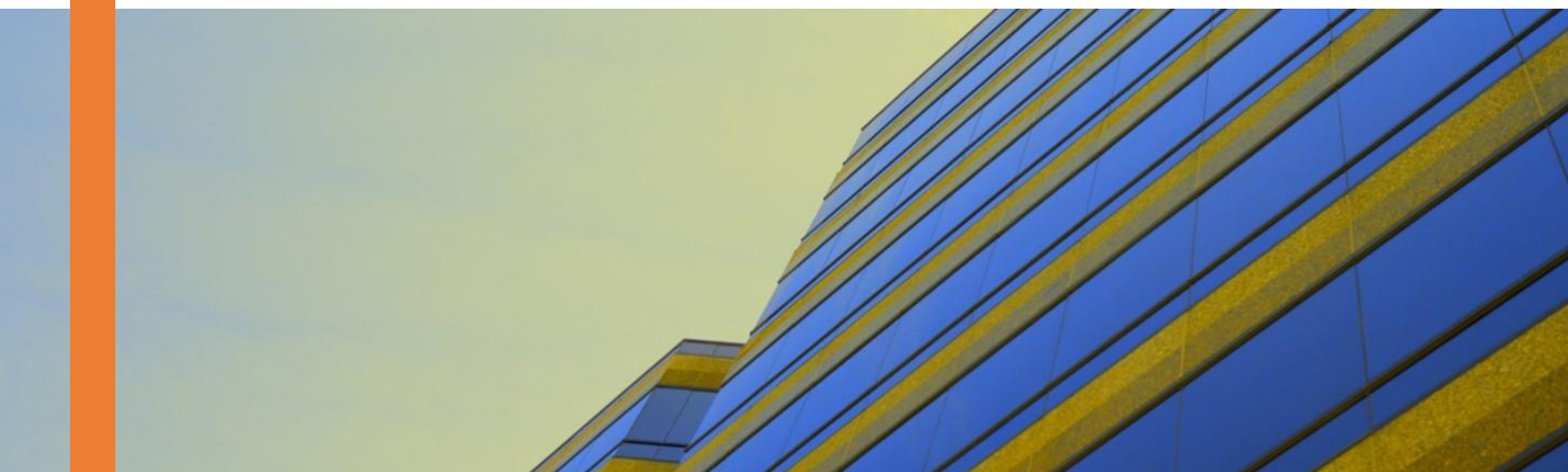


# ZEROCLICK, LLC. V. APPLE INC.



**KRIS RHU**  
HARRITY & HARRITY LLP  
11350 Random Hills Road  
Suite 600  
Fairfax, VA 22030

*Zeroclick, LLC. v. Apple Inc.*

Zeroclick sued Apple alleging that Apple infringed claims 2 and 52 of U.S. Patent No. 7,818,691 and claim 19 of U.S. Patent No. 8,549,443, which relate to modifications of graphical user interfaces (GUIs) that allow the GUIs to be controlled using pre-defined pointer or touch movements instead of mouse clicks. Apple responded by asserting invalidity of those claims. The U.S. District Court for the Northern District of California found that the claims recited means-plus-function terms for which the specification did not have sufficient structure, and therefore, were invalid for indefiniteness. However, the Court of Appeals for the Federal Circuit (CAFC) vacated and remanded, holding that the district court did not undertake the relevant inquiry when making factual findings to support its conclusion that the claims recited means-plus-function terms.

Claim 2 of the '691 patent recites:

2. A graphical user interface (GUI), which may comprise an update of an existing program, that may fully operate a GUI by a two step method of movement of a pointer (0) to operate one or more functions within the GUI,  
wherein, said existing program is any existing **program that can operate the movement of the pointer (0)** over a screen (300) and has one or more functions operated by one or more other methods apart from said two step method,  
and/or one or more functions operated by said one or more other methods in said existing program can be updated to operate by said two step method,  
wherein said GUI executes one or more functions within the GUI by the completion of the following said two step method:  
first said pointer (0) is immediately adjacent or passes within a control area (1), which is an area of the screen (300) that may be any size including from a pixel on the screen (300) to occupying the whole screen (300), and  
second by the completion of a subsequent movement of said pointer (0) according to a specified movement generates a 'click' event, thereby triggering one or more functions within the GUI.

Claim 52 is directed to the "method of operating a graphical user interface" described in claim 2.

Claim 19 of the '443 patent recites:

19. A device capable of executing software comprising:  
a touch-sensitive screen configured to detect being touched by a user's finger without requiring an exertion of pressure on the screen;  
a processor connected to the touch-sensitive screen and configured to receive from the screen information regarding locations touched by the user's finger;  
executable user interface code stored in a memory connected to the processor;  
the user interface code executable by the processor;  
the **user interface code being configured to detect one or more locations touched by a movement of the user's finger on the screen without requiring the exertion of pressure and determine therefrom a selected operation;** and  
the user interface code is further configured to cause one or more selected operations, which includes one or more functions available to the user interface code of the device, to deactivate while the user's finger is touching one or more locations on the screen.

During claim construction, the district court found that the limitation “program that can operate the movement of the pointer (0)” in claim 2 of the ‘691 patent is a means-plus-function term. Specifically, it found “program” to be a means that performs the function of operating the movement of the pointer. The district court also found that “user interface code being configured to detect one or more locations touched by a movement of the user’s finger on the screen without requiring the exertion of pressure and determine therefrom a selected operation” in claim 19 of the ‘443 patent is a means-plus-function term. Specifically, it found “user interface code” to be a means that performs the functions of detecting one or more locations touched by a movement of the user’s finger on the screen without requiring the exertion of pressure and determining therefrom a selected operation.

The CAFC cited *Williamson v. Citrix Online, LLC*, which stated that the failure to use the word “means” creates a rebuttable presumption that 35 U.S.C. § 112, ¶ 6 does not apply, but that the presumption can be overcome “if the challenger demonstrates that the claim term fails to recite sufficiently definite structure or else recites function without reciting sufficient structure for performing that function.” 792 F.3d 1339, 1348 (Fed. Cir. 2015) (en banc). *Williamson* further stated that the essential inquiry in determining whether a claim limitation involves § 112, ¶ 6 is “whether the words of the claim are understood by persons of ordinary skill in the art to have a sufficiently definite meaning as the name for structure.” *Id.* The CAFC found that the district court failed to undertake this inquiry and make related factual findings.

The CAFC noted that the claims did not include the word “means,” which invokes the presumption. The CAFC also noted that Apple provided no evidentiary support for their position that the claim limitations must be construed under § 112, ¶ 6, and therefore, failed to rebut the presumption. By relying on Apple’s arguments without evidentiary support, the district court legally erred by not giving effect to the unrebutted presumption.

The CAFC further found that the district court effectively treated the features “program” and “user interface code” as nonce words that operate as a substitute for “means,” and found such treatment to be erroneous for three reasons: 1) just because the disputed limitations incorporate functional language does not necessarily convert those words into means for performing those functions. There are plenty of devices that take the names of the functions they perform (e.g. brake, clamp, screwdriver); 2) the district court removed those terms from their context, which otherwise strongly suggests the plain and ordinary meaning of the terms. Claim 2 of the ‘691 patent recites a “graphical user interface” which may “comprise an update of an existing program” using a two-step method. Claim 19 of the ‘443 patent tethers “user interface code” to the code “stored in a memory connected to the processor.” That processor, in turn, is configured to perform the indicated functions. Thus, a person of ordinary skill in the art could reasonably discern from the claims that the words “program” and “user interface code” are not used as generic terms, but rather as specific references to conventional GUI programs or code existing previously. The written description further bolsters this conclusion because it describes the distinction drawn between GUIs in the prior art and the improvement to such interfaces in the claimed invention; and 3) the district court made no findings that compel the conclusion that a conventional GUI program or code is commonly used as a substitute for “means.”

This opinion illustrates the importance of avoiding the use of the word “means” in order to create a rebuttable presumption that 35 U.S.C. § 112, ¶ 6 does not apply to a claim. The opinion also illustrates the importance of tethering software-esque terms like “program” or “code,” or other terms that may not inherently have structure, to a specific structure in the claims and/or providing

context to such claim limitations. Such strategies can help prevent claim limitations from being construed under 35 U.S.C. § 112, ¶ 6.