Frames() Timing Function

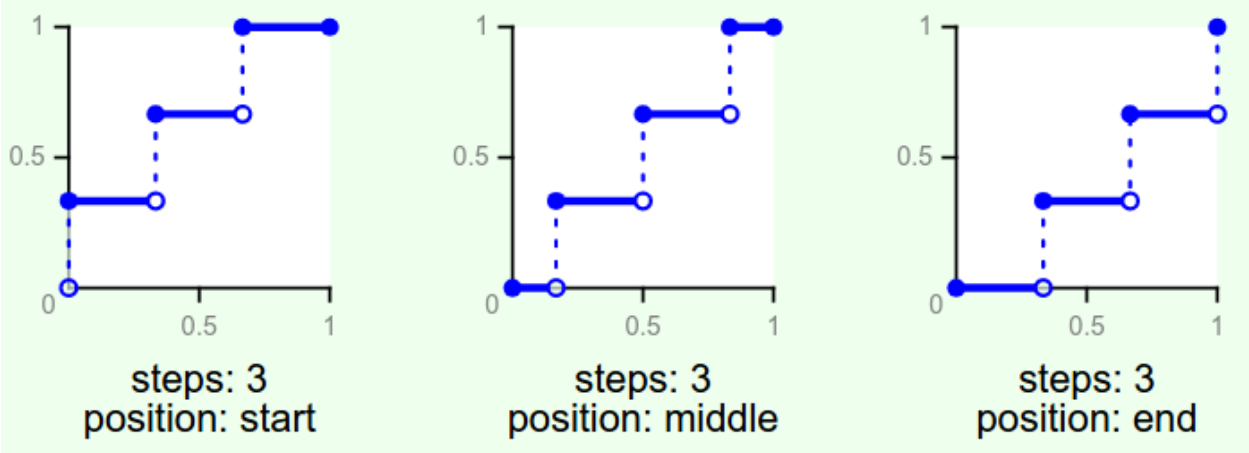
suzyh@chromium.org
Last updated: 16 Mar 2017
http://crbug.com/646265

Objective

Bring Chromium's timing function capabilities in line with the CSS Timing Functions spec by implementing the frames timing function.

Background

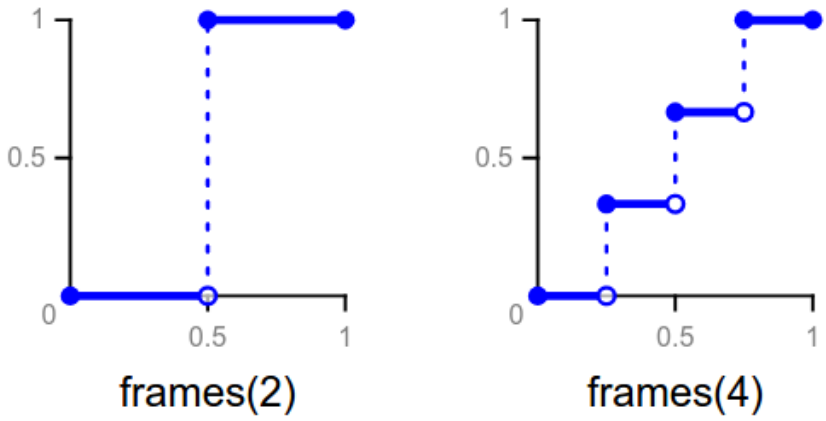
The Web Animations spec defines a step timing function that allows an animation to progress in a number of discrete steps from start to end. It defines three step positions---start, middle and end---which specify at which point during the interval the value change occurs. The following excerpt from the diagram from the spec (linked above) illustrates these functions:



A W3C discussion in Mar 2016 proposed replacing the step timing function, in particular with *position: middle*, with a new frames timing function.

The definitions of the step and frames timing functions, along with the linear and cubic Bézier timing functions, have been transferred to the CSS Timing Functions spec. The *position: middle* variation of the step timing function is not included.

The following diagram from the CSS Timing Functions spec [section link] illustrates the frames timing function:



In particular, notice the difference in length of the start and end frames when compared to the step timing function with *position: middle*.

Chromium currently implements support for the step timing function, including *position: middle*, and does not support the frames timing function. Code locations: platform/animation/TimingFunction.h, cc/animation/timing_function.h.

Implementation Plan

Strategy

The most significant difference between the step and frames timing functions is in the length of the start and end frames. In the frames timing function, all frames are of equal duration, while in the step timing function, the start and/or end frame is shorter than the other frames. For this reason, the implementation of the frames timing function cannot be achieved by simple aliasing of the existing step timing function implementation.

Option A: Step and frames timing functions parsed separately but share implementation

- Less duplication of code
- Implementation must be more general than the definition of either timing function on its own

Option B: Step and frames timing functions are parsed and implemented separately

- Enables alignment between code and spec, so function names etc. can be clearer

- Duplicate code is mostly the boilerplate of the timing function class
- Code does not reflect the similarity between the two timing functions

I propose to go with option B because:

- option A requires the implementation to be more general (probably unnecessarily),
- in option B, the structure of the code will more clearly align with the spec, and
- option B's duplicated code is not likely to contain complex logic.

Code

Adding an implementation for the frames timing function requires:

- A new subclass of TimingFunction
  - in Source/platform/animation
  - in src/cc/animation
- A new CSS*TimingFunctionValue
- Hooks into the CSS and parsing code
  - Source/core/css/ComputedStyleCSSValueMapping.cpp
  - Source/core/css/parser/CSSPropertyParser.cpp
  - Source/core/css/resolver/CSSToStyleMap.cpp

Tests

The web-animations suite of the Web Platform Tests currently contains one test that exercises the frames timing function [test, failure expectation].

The Mozilla copy of this web-animations suite contains additional tests with invalid frames() values, which will be synced to the Web Platform Tests in the near future. After sync, these will also be available in the Chromium repository.

Since the frames timing function is defined in the CSS Timing Functions spec, rather than the Web Animations spec, the best location for additional tests are in a test suite for the CSS Timing Functions spec. Some such tests exist in the Mozilla copy of the Web Platform Tests, and once they have been synced to the Web Platform Tests, we will be able to import them to Chromium. This may happen automatically, as long as the W3CImportExpectations does not skip that directory (css-timing-1).

To test the frames timing function the compositor thread, one or more additional tests should also be added to LayoutTests/animations/resources/composited-animations-data/timing-function.js.