View Transitions

Information

- Epic: https://mozilla-hub.atlassian.net/browse/FFXP-2525
- Spec:
  - [SPA] https://drafts.csswg.org/css-view-transitions-1/
  - [MPA] https://drafts.csswg.org/css-view-transitions-2/
- Meta:
  - css-view-transitions-1 (Bug 1823896)
  - css-view-transitions-2 (Bug 1860854)
- Tech doc: View Transitions
- Slack: #project-view-transitions
- Other browsers:
  - Blink:
    - [SPA] https://chromestatus.com/feature/5193009714954240
    - [MPA] https://chromestatus.com/feature/5118874666663936
  - WebKit
    - [SPA] https://bugs.webkit.org/show_bug.cgi?id=259055

Meeting Notes

Oct 24, 2024

Attendees: Boris, Emilio, Macro, Nicolas, Valerie

- Triage one new bug: Bug 1926021
- Go through graphics bugs and assign points

Oct 17, 2024

Attendees: Boris, Emilio, Macro, Nicolas, Valerie

- No bug needs to be triaged today
- Emilio: The dependency of Create pseudo-element tree
- Boris: Generalize the pseudo-element requests to also carry an identifier (for both getComputedStyle and Web Animations)

Oct 10, 2024

Attendees: Boris, Emilio, Macro, Martin, Nicolas, Valerie

- Assigned story points for each bug

    - Layout backlog
- Discussed https://bugzilla.mozilla.org/show_bug.cgi?id=1922332

Oct 3, 2024

Attendees: Boris, Emilio, Macro, Nicolas, Valerie

- Go through the bugs and status of our breakdown
- Development phase: 6 weeks after Oct 10
    - We use an extra week to prepare and do labeling
    - Target: What are we actually able to do in these 6 weeks?

Oct 2, 2024

Attendees: Boris, Emilio, Nicolas

- Go through the bugs, assign the t-shirt size
    - Notes - Nical/Boris/Emilio go through View Transitions spec/bugs
- Let's use the (tentative) tshirt sizes
    - S: ~ 1-2 day
    - M: ~ 3 days
    - L: ~ 5 days
    - XL: ~10 days
    - XXL: ~20+ days
- **1847089:** Detect UA transitions on same-document navigations: S
- **1914014:** Implement selector matching for view transition pseudo elements
    - **1922076:** Implement selector matching for ::view-transition: S
    - **1922078:** Implement selector matching for ::view-transition-group(..): M
    - **1922079:** Implement selector matching for ::view-transition-image-pair(..): S
    - **1922080:** Implement selector matching for ::view-transition-old(..): S
    - **1922082:** Implement selector matching for ::view-transition-new(..): S
- **1914321:** [css-view-transitions] Create pseudo-element tree.: L
    - **1918400:** Capture old state for view transitions: M (excluding snapshotting)
    - **1918402:** Capture new state for view transitions: S
    - **1919333:** Implement right inheritance in view transitions pseudo-elements: M
- **1914322:** [css-view-transitions] Hook into the refresh driver.: S
- **1914323:** [css-view-transitions] Import static UA style sheet.: S
- **1914324:** [css-view-transitions] Implement dynamic UA stylesheet.: XL
- **1918404:** Detect pending animations on each view transition frame: M
- **1918405:** Figure out why some test failures from bug 1916311 only happened after merge on central, rather than on autoland and try.: S
- **1919341:** [css-view-transitions] Web animations integration
    - **1921109:** Make Element.getAnimations({subtree: true}) return animations on view-transition pseudo elements: S
    - **1921112:** Add the handling of view-transition pseudo elements into Element.animate(): S

- **1922095:** Support CSS Animations on view-transition pseudo-elements
    - **1921116:** Make AnimationData store different kind of pseudo elements (especially for view transition pseudo elements): S
- **1922315**: Generalize the pseudo-element requests to also carry an identifier (for both getComputedStyle and Web Animations): M
    - **1921553:** Use (PseudoStyleType, nsAtom) as a parameter to look up the pseudo element for animations: S
    - **1919347:** Make getComputedStyle work with view-transitions pseudo-elements: S
- **1919356:** [css-view-transitions] Skip transition on viewport resizes.: S
- **1919358:** [css-view-transitions] Implement invalidation for view transition pseudo-elements: M
- **1922298:** Implement the rule that elements with non-none view-transition-name become a stacking context.: S
- **1922299:** Implement the snapshot containing block concept: M
- **1922300:** Implement the view transition layer: L
- **1922301:** Make sure view transition pseudo-elements are not exposed to a11y.: S
- **1922322**: Add a fast way to check whether an element is a captured element for layout.: M
- **1922302:** Make captured elements stacking contexts: S
- **1922304:** Make captured elements behave as visibility:hidden and pointer-events: none: S
- **1922305:** Make live captured element rendering work: XXL (needs investigation)
- **1922332**: Make sure we have a story for captured elements outside of the displayport XL (needs investigation)
- **1922333**: Integrate the displaylist building logic with WebRender's snapshotting functionality for view-transitions S
- **1922335**: Implement a fallback code-path for view-transitions based on SurfaceFromElement.: M

Oct 1, 2024

Attendees: Boris, Emilio, Nicolas

- Go through the bugs and spec:
  Notes - Nical/Boris/Emilio go through View Transitions spec/bugs

Notes (copied from the notes above)

- Dependency tree
- Nical's breakdown so far: View-transitions Nical's task breakdown
- Animation breakdown (also in the dependency tree)
    - In order to support the CSS animations on view transition pseudo (Bug 1922095)
        - Use (PseudoStyleType, nsAtom) as a parameter to look up the pseudo element for animations (Bug 1921553)
        - Make AnimationData store different kind of pseudo elements (especially for view transition pseudoe elements) (Bug 1921116)

    - Web animations integration (Bug 1919341)
        - Add the handling of view-transition pseudo elements into Element.animate() (Bug 1921112)
        - Make Element.getAnimations({subtree: true}) return animations on view-transition pseudo elements (Bug 1921109)
- Starting a transition turns some things into stacking context. Which sort of forces us to re-create the display list. (spec link)
- Emilio’s breakdown: Mostly looked at the algorithms in the spec and filed bugs to cover them.
- Adding pseudo elements: normally they'd hang off of elements, but this is tricky because theoretically you can swap out the document element and these should remain.
    - Unclear if any test-case cares about it. Certainly not any realistic use-case of the feature so we might be able to simplify a bit.
    - https://bugzilla.mozilla.org/show_bug.cgi?id=1914321 tracks generating the pseudos (unclear if it’s splittable into smaller chunks)
    - To work around that we should hang them off of the document which adds a bit of complexity for css matching but it should be manageable
- Small capture and mix-blend-mode stuff:
    - https://drafts.csswg.org/css-view-transitions-1/#capture-the-old-state captures some properties by rendering the element, and some others by actually taking the computed value, and applying it later.
- Layout vs. webrender bounds:
    - https://drafts.csswg.org/css-view-transitions-1/#capture-the-image (non document element) checks the ink-overflow rect, which is a layout concept.
- Large element capture:
    - We could fallback to whatever -moz-element does: https://searchfox.org/mozilla-central/rev/1b90936792b2c71ef931cb1b8d6baff9d825592e/layout/painting/nsImageRenderer.cpp#161

Action items

- ☑ ~~File a bug for https://drafts.csswg.org/css-view-transitions-1/#named-and-transitioning (making named elements a stacking context and co).~~
    - ::view-transtion // a stacking context, view-transtion-layer. Tree root
    - – ::view-transition-group(abc) // a new stacking context
    - – ::view-transition-group(root) // a new stacking context
- ☑ ~~File a bug to make sure that view-transition pseudos are not exposed to a11y: https://drafts.csswg.org/css-view-transitions-1/#view-transition-pseudos~~
- ☑ ~~Try to split https://bugzilla.mozilla.org/show_bug.cgi?id=1914014~~
    - Root
    - Group(..)
    - image-pair(..)
    - old(..)/new(..)
- ☑ ~~Track implementing the snapshot containing block: https://drafts.csswg.org/css-view-transitions-1/#snapshot-containing-block~~
- ☑ ~~Also https://drafts.csswg.org/css-view-transitions-1/#view-transition-layer (semi-related)~~

        - Implementation should be very similar to the current top layer implementation.
        - Might need extending -moz-top-layer (but hopefully not)
        - It creates a CB for fixed-pos and abspos, which simplifies things significantly.
- ☑ ~~Track making captured elements visibility: hidden + pointer-events: none:~~
    - https://drafts.csswg.org/css-view-transitions-1/#view-transition-stacking-layer
- ☑ ~~Track making the new snapshot live. Kinda falls off from:~~
    - https://drafts.csswg.org/css-view-transitions-1/#update-pseudo-element-styles
    - If capturedElement's new element is not null, then:
        - Let new be the ::view-transition-new() with the view transition name transitionName.
        - Set new's replaced element content to the result of capturing the image of capturedElement's new element.

Sep 26, 2024

Attendees: Boris, Emilio, Macro, Nicolas, Valerie

- Mostly talked about the breakdown today
    - Probably two pools for bugs: contingency pool and defect pool
    - It's hard to predict the time we need for each test in WPT, and some edge cases may take longer.
    - Share your breakdowns with each other, and know it's a clear, confident message, and so this act is good to go.
- Will meet each other some other time to go through the breakdown

Sep 17, 2024

Attendees: Boris, Emilio, Marco, Martin, Valerie

- The scope of our current milestone
- Emilio: Break down items, and nical break the graphics work down. Track tests in WPT and fix them.
- Valerie: Do the current wpt cover the spec level 1?
    - Emilio: Probably 90%
- Boris: will break down more things for other CSS part and animations
- Emilio: there are some other work we may have to work on as part of view transitions
    - https://bugzilla.mozilla.org/show_bug.cgi?id=1871584
    - https://bugzilla.mozilla.org/show_bug.cgi?id=1770591

Sep 9, 2024

Attendees: Boris, Emilio, Hsin-Yi, Marco, Martin, nical, Valerie

- [Marco] Provide the Forecasting lighting talk
    - Break down the work into actionable items (in JIRA)
        - Try to get small items
    - Story points (i.e. the size of the item) for each actionable item

    - So we can see what is going on to happen, for our projects
    - May have micro risk tasks (i.e. plan B)
- Break our work into small tasks in the following meetings
- Change the meeting time to 30min weekly.

Sep 3, 2024

Attendees: Boris, Emilio

- Quick sync on some working items at the current stage
    - Emilio: DOM API support, and the creation of pseudo elements tree
        - The implementation about the creation of pseudo elements may be a little bit complicated, so we will land the patches step by step. At some points we could continue to work in parallel with other CSS stuff
    - Boris: continue to work on selectors matching, CSS integration (and stylesheet) once we have some implementation of pseudo elements.
- Also, we may have to add some special handling of the animations because the pseudo element names are dynamically. (This also depends on the structure of pseudo elements tree)
    - Note: our current CSS Animations only apply to ::before, ::after, and ::marker. We hard-coded the pseudo element types/names. We have to tweak it.

Aug 13, 2024 (MozWeek)

- Notes - View transitions

Jul 16, 2024

Kick off for View Transitions project.
Agenda:

- Overview of project framework and process
    - DOM API: [`startViewTransition()`]
    - View Transition Pseudo-elements:
    - Basically, we snapshot the old DOM state and new DOM state, before and after calling `document.startViewTransitions()`. And create pseudo elements for the old state and new state, so we can implicitly apply CSS Animation to these Transition pseudo-elements
        - `::view-transition`
        - `└─ ::view-transition-group(root)`
        - `   └─ ::view-transition-image-pair(root)`
        - `      ├─ ::view-transition-old(root)`
        - `      └─ ::view-transition-new(root)`
    - Support CSS animations for these new pseudo elements (e.g. `::view-transition-old` and `::view-transition-new`)

  - Support new selector, pseudo class, and new css property (i.e. `view-transition-name`)
- Meta bug for VT level 1: https://bugzilla.mozilla.org/show_bug.cgi?id=1823896
- Goal and scope of the project
  - Single page in H2 2024
  - Multiple pages, need more DOM stuff, e.g. Navigation API, later
    - FFXP-1948 - Navigation API (in progress)
- Roles & responsibilities (who's involved, etc)
  - SPA
    - CSS parsing and selector-matching
    - DOM:
      - Actual API
      - Pseudo-element setup
      - Olli Pettay has the most of the context to do the architecture review; Andreas Farre is happy to take any review requests coming to his plate.
    - Graphics
      - Need an API for snapshotting the DOM (or reusing the -moz-element / drag&drop / etc). We're going to need WR integration for L2, so it'd be great if GFX can get that working before end of H2 so that we don't have to write this twice.
      - @Nicolas Silva , but need to coordinate with him for the bandwidth
    - Standards
      - No blocking issues for now, though we may find while we're implementing.

- Communication and status updates
- Discuss what time works for everyone for bi-weekly meeting
  - 30min - 45min
  - 1st: 9am PDT, 6pm Berlin (CEST)
  - 2nd: 8am PDT, 5pm Berlin (CEST)
  - Not Wed (CEST)
  - Mon or Thurs?
  - https://www.timeanddate.com/worldclock/meeting.html ?
  - https://app.rallly.co/new
  - And MozWeek
- Nical's questions:
  - What should happen if we need to snapshot something very large?
    - Could cancel the animation?
  - What should happen if we need to snapshot something that is outside the displayport?
    - Need to support this properly, means we have to provide content in the DL that would have been clipped out otherwise.
  - Complicated clipping cases?

        - Should be made work as well. In general it should work somewhat like the layerization that happens for, say, a filter, except that the bounds of the layerized content are not affected by clips, and the clips are applied on top of the resulting image. We'd have to make sure to account for things that can inflate the bounds of the layer, for example shadows.
    - Would it make sense to, instead of representing the transitionable elements as images, represent them as standalone display lists?
        - Would give us the ability to rasterize on demand and at the resolution we want during transitions. More general but probably more complicated and not what chrome is doing.
- Action for emilio:
    - Play with clipping and other edge cases to see how the spec and Blink/WebKit deal with it.
        - Does it render un-clipped stuff?
        - Does it apply ancestor effects like opacity?
        - How do shadows and such get treated?