XR Navigation thoughts

Why?
Because “Metaverse”

(Obviously)

Fundamental challenge: Define a “link” in XR

There’s a whole bunch of ways that navigation could be presented, but fundamentally it all boils down to defining a new type of link for the browser.

WebXR is (currently) entirely imperative

So we have to have something that dynamically communicates to the browser “The user is interacting with a link now”.

Proposal: Navigation Contexts

Navigation Context == “If the user navigated right now, this is where they would go.”

- User is looking at an object
- User is pointing at an object
- User is holding an object

- Party organizer is suggesting a destination
- Currently viewed slide has an associated URL

- User is standing in a specific place
- User has stepped through a portal/door

Navigation destination is shown to the user as trusted UI

Up to the UA to decide how to present it (head locked, waist locked, wrist locked?)

No actual navigation takes place until the user uses a secure gesture (ie: Oculus menu button)

Important!

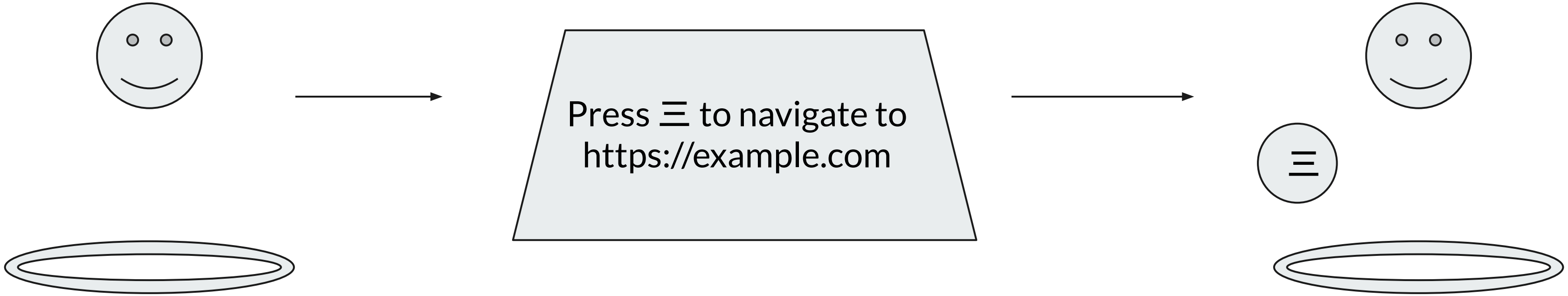
Navigation is ALWAYS initiated by a trusted gesture and handled by the UA.

Example user experience

User stands on teleport pad

Trusted UI is shown to the user Indicating they can navigate

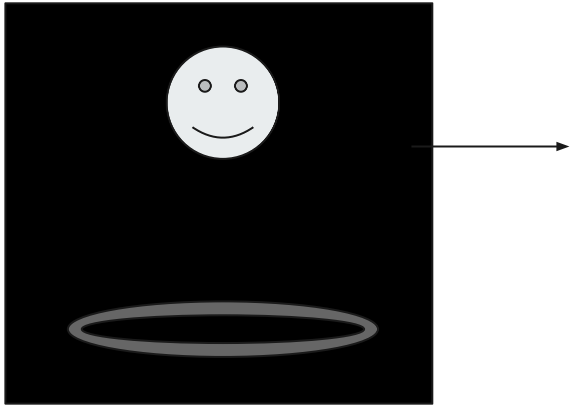
User slaps that trusted button!



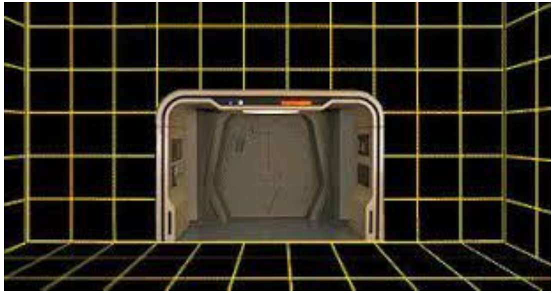
Press ☰ to navigate to https://example.com

Example user experience, Part 2

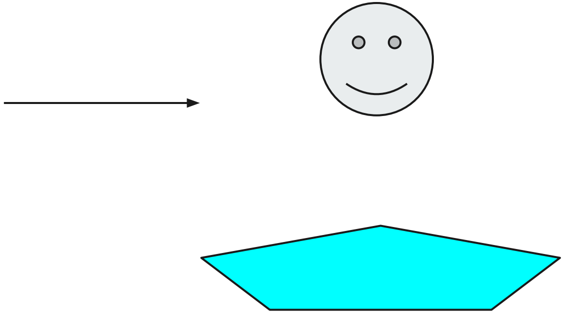
Quick fade to black



User shown some UA-provided intermediate space while navigation occurs



When destination page is ready fade back in from black.



Transition between pages should start simple

We can add more complex transitions, with page-provided interstitial environments, later. Initially letting the destination page take over rendering as quickly as possible should be the goal.

Proposed API, Navigation Contexts

```
// Set a navigation context
xrSession.setNavigationContext ({
  // Required
  href: 'https://example.com' ,

  // Optional/ignorable?
  text: 'Example Page' ,
  pose: rigidTransform,
  space: referenceSpace,
});

// Clear the navigation context
xrSession.clearNavigationContext ();
```

Open Questions, Navigation Contexts

- What would a reliable, unspoofable gesture be for hand-based navigation?
- Would a pose for the NC be useful? (Or an avenue for abuse?)
- What about a text description?
- Mitigations for rapid switching.
- If navigation gesture is overloaded with exit immersive mode gesture, how to mitigate spamming navigation contexts to block exiting immersive mode?
- Any explicit support needed for supporting a “back” gesture that retains context?

The other side:
Accepting Navigation Requests

Accepting navigation requests

- Needs to signal early, but also be dynamic. May rule out tags?
- Page needs to specify session options like required features.
- Want to be able to early-out on pages with no XR capabilities.
  - Require it be called before the document DOMContentLoaded event is finished firing.
  - (That fires before window.onload, when all dom is parsed but not CSS/images/etc)
  - Allows you to call offersSession in the event to guarantee you catch the signal.
- As mentioned previously, could be the same mechanism as a “Easy entry” UI button.

Proposed API, Accepting Navigation Requests

```
// Accepting a navigation request
navigator.xr.offersSession('immersive-vr', {
  requiredFeatures: 'hand-tracking',
}).then((session) => {
  /* Onward! */
});
```

Using isSessionSupported as the signal?

- Interesting idea to dissuade fingerprinting.
- Adds an unexpected side effect to a shipped API.
- May train users not to trust the "easy" WebXR button, rely more on in-page buttons.
- Probably best to have a purpose-build API as a result.

Open Questions, Accepting Navigation

- How to catch both an AR or VR session?
- Should you be able to navigate between AR/VR sessions?
- Communicating readiness to display. Maybe even a progress percentage?