Spatial Navigation on the Web

TPAC 2018

Breakout Session

Jihye Hong, LG Electronics

Overview

- Spatial Navigation
- Use cases
- Spec
  - Heuristic processing model
  - API
- Next Steps
- Demo

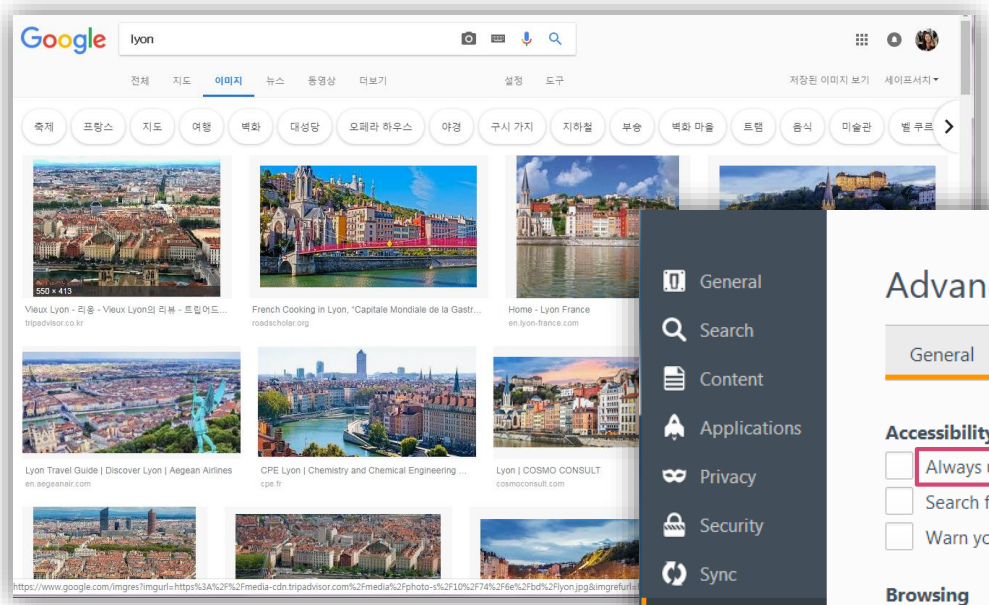
Spatial Navigation?

- Two-dimensional navigation within the pages of web app.
- Navigate around your pages in an intuitive and robust way in various d evices. (TV, PC, IVI, etc)

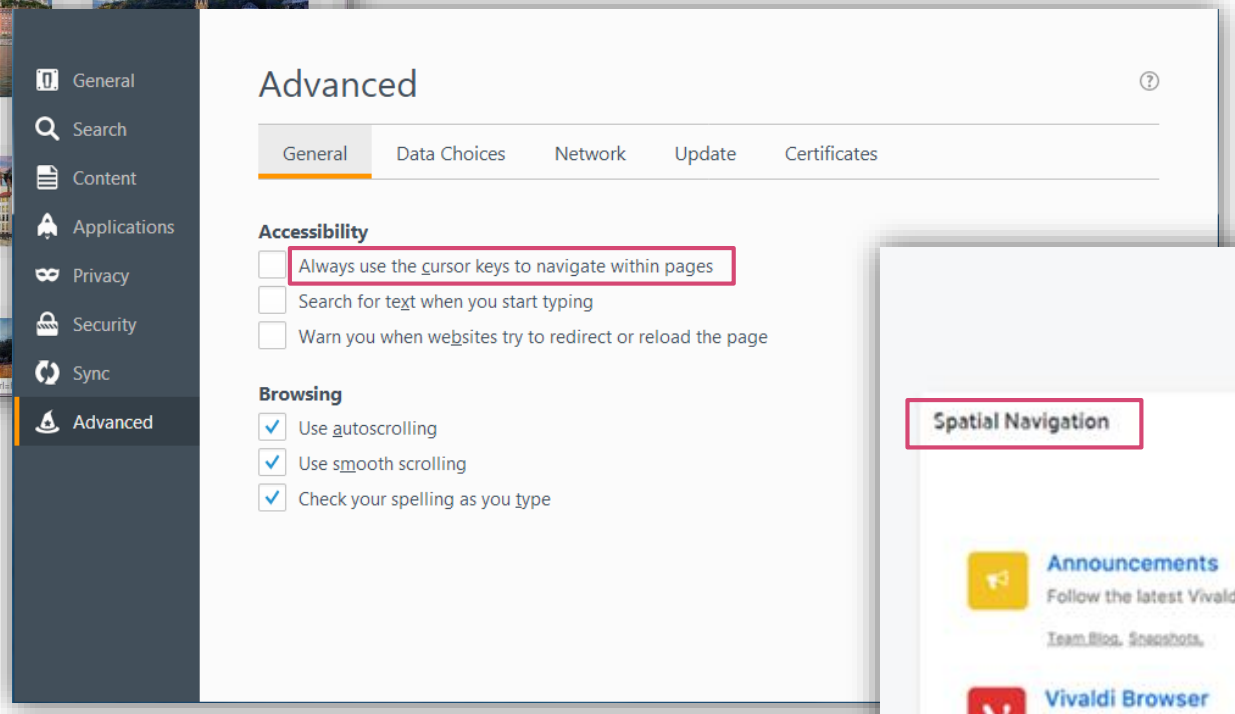
TV Web Application

Grid-like Web Application

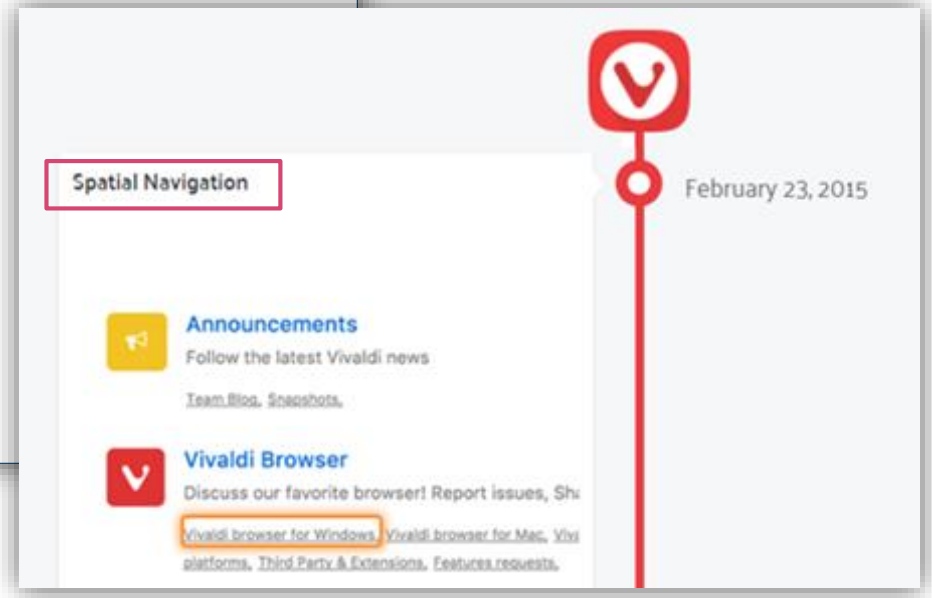
Use cases



Google's Product
(Image Search,
Gmail,
Google Docs)

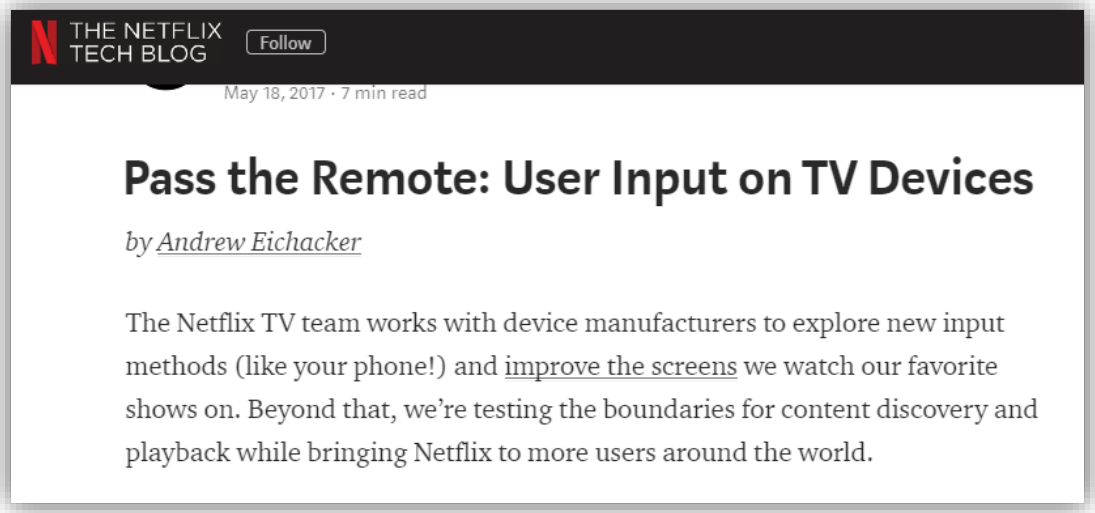


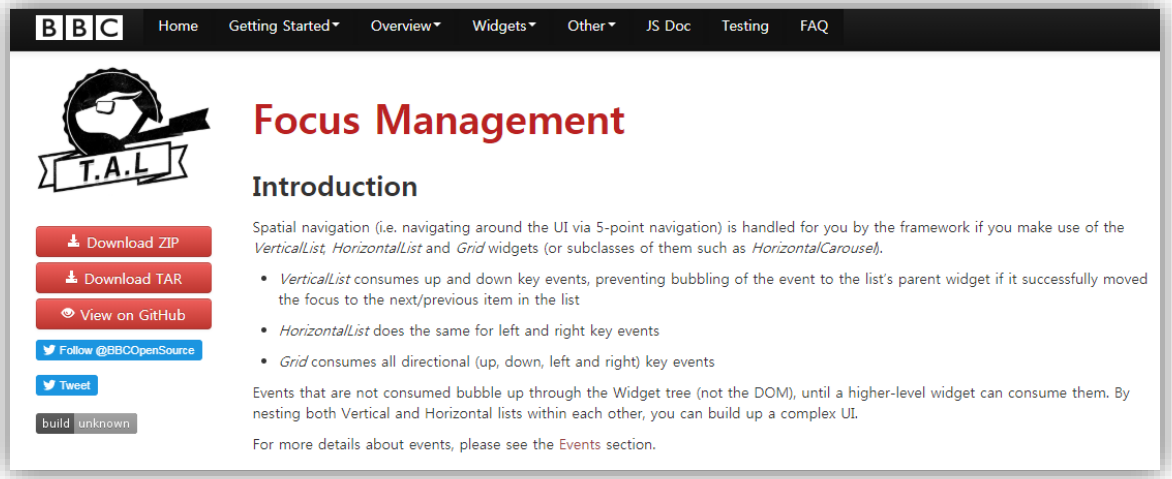
Directional Cursor Navigation
in Firefox

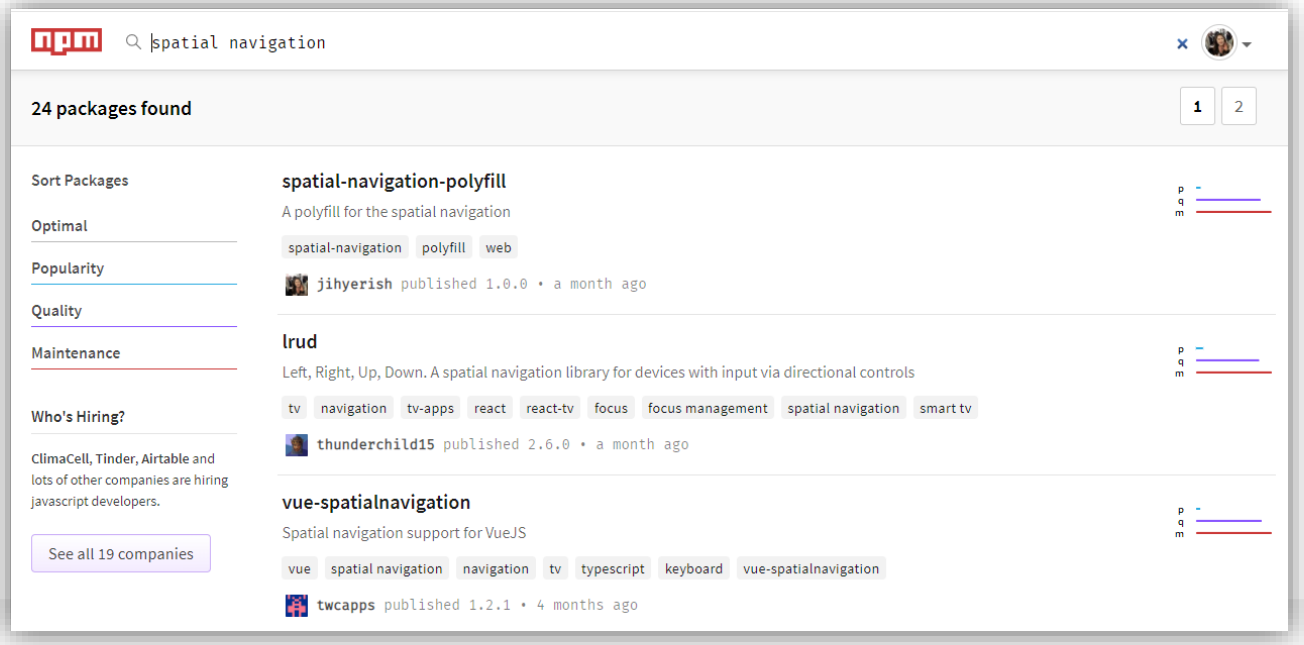


Spatial Navigation Feature
in Vivaldi

Use cases


THE NETFLIX TECH BLOG
Follow
May 18, 2017 · 7 min read
Pass the Remote: User Input on TV Devices
by Andrew Eichacker
The Netflix TV team works with device manufacturers to explore new input methods (like your phone!) and improve the screens we watch our favorite shows on. Beyond that, we're testing the boundaries for content discovery and playback while bringing Netflix to more users around the world.

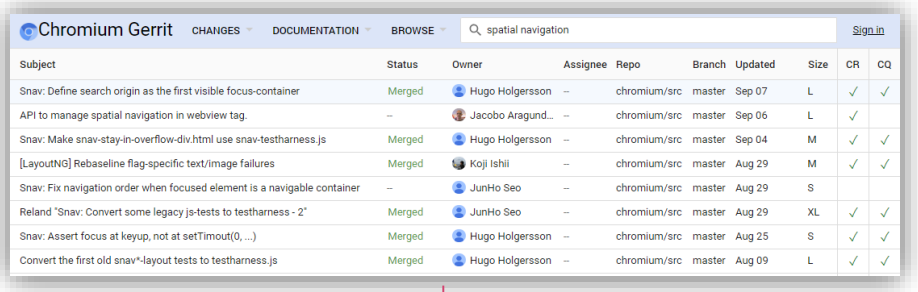

BBC
Home
Getting Started
Overview
Widgets
Other
JS Doc
Testing
FAQ
T.A.L
Download ZIP
Download TAR
View on GitHub
Follow @BBCOpenSource
Tweet
build unknown
Focus Management
Introduction
Spatial navigation (i.e. navigating around the UI via 5-point navigation) is handled for you by the framework if you make use of the VerticalList, HorizontalList and Grid widgets (or subclasses of them such as HorizontalCarousel).
VerticalList consumes up and down key events, preventing bubbling of the event to the list's parent widget if it successfully moved the focus to the next/previous item in the list
HorizontalList does the same for left and right key events
Grid consumes all directional (up, down, left and right) key events
Events that are not consumed bubble up through the Widget tree (not the DOM), until a higher-level widget can consume them. By nesting both Vertical and Horizontal lists within each other, you can build up a complex UI.
For more details about events, please see the Events section.


npm
spatial navigation
24 packages found
1
2
Sort Packages
Optimal
Popularity
Quality
Maintenance
Who's Hiring?
ClimaCell, Tinder, Airtable and lots of other companies are hiring javascript developers.
See all 19 companies
spatial-navigation-polyfill
A polyfill for the spatial navigation
spatial-navigation
polyfill
web
jihyerish published 1.0.0 • a month ago
lrud
Left, Right, Up, Down. A spatial navigation library for devices with input via directional controls
tv
navigation
tv-apps
react
react-tv
focus
focus management
spatial navigation
smart tv
thunderchild15 published 2.6.0 • a month ago
vue-spatialnavigation
Spatial navigation support for VueJS
vue
spatial navigation
navigation
tv
typescript
keyboard
vue-spatialnavigation
twcapps published 1.2.1 • 4 months ago

Need for Standard!

- Evolving navigation for evolving layout in the web
- Limitation of JS lib for handling iframe, shadow DOM
- Less customization for consistent UX

History



17.11

Introduce the Spatial Navigation @W3C TPAC 2017

18.04

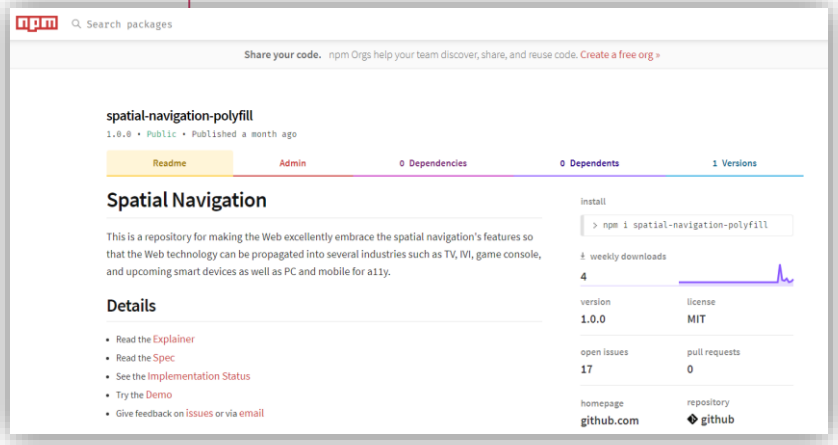
Introduce Spatial Navigation Spec Draft @CSS WG F2F

18.08

Contribute patch (#8) to Chromium upstream about the Heuristic Behavior of Spatial Navigation

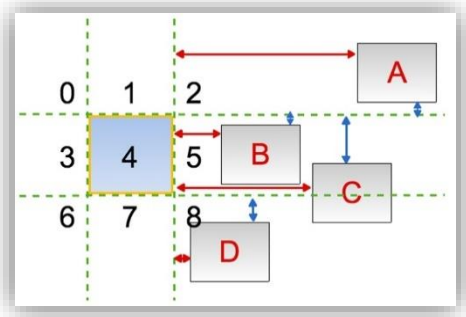
18.09

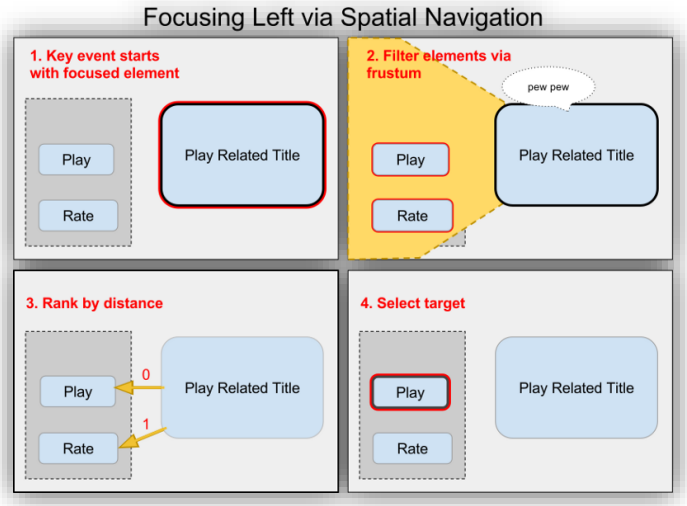
Release polyfill



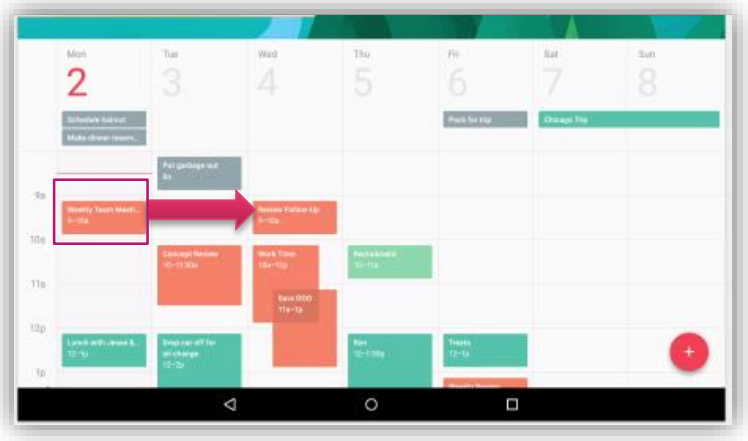
Find the Primitive Features

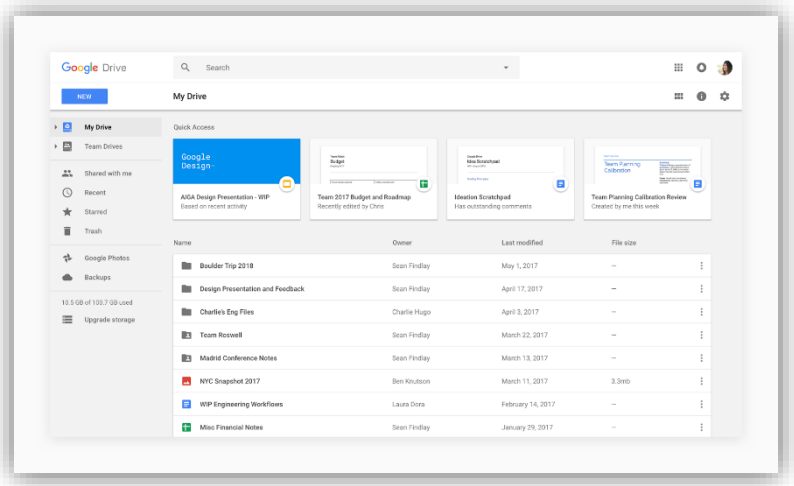
Processing Model





APIs





Spatial Navigation in WICG

- Processing Model
- APIs

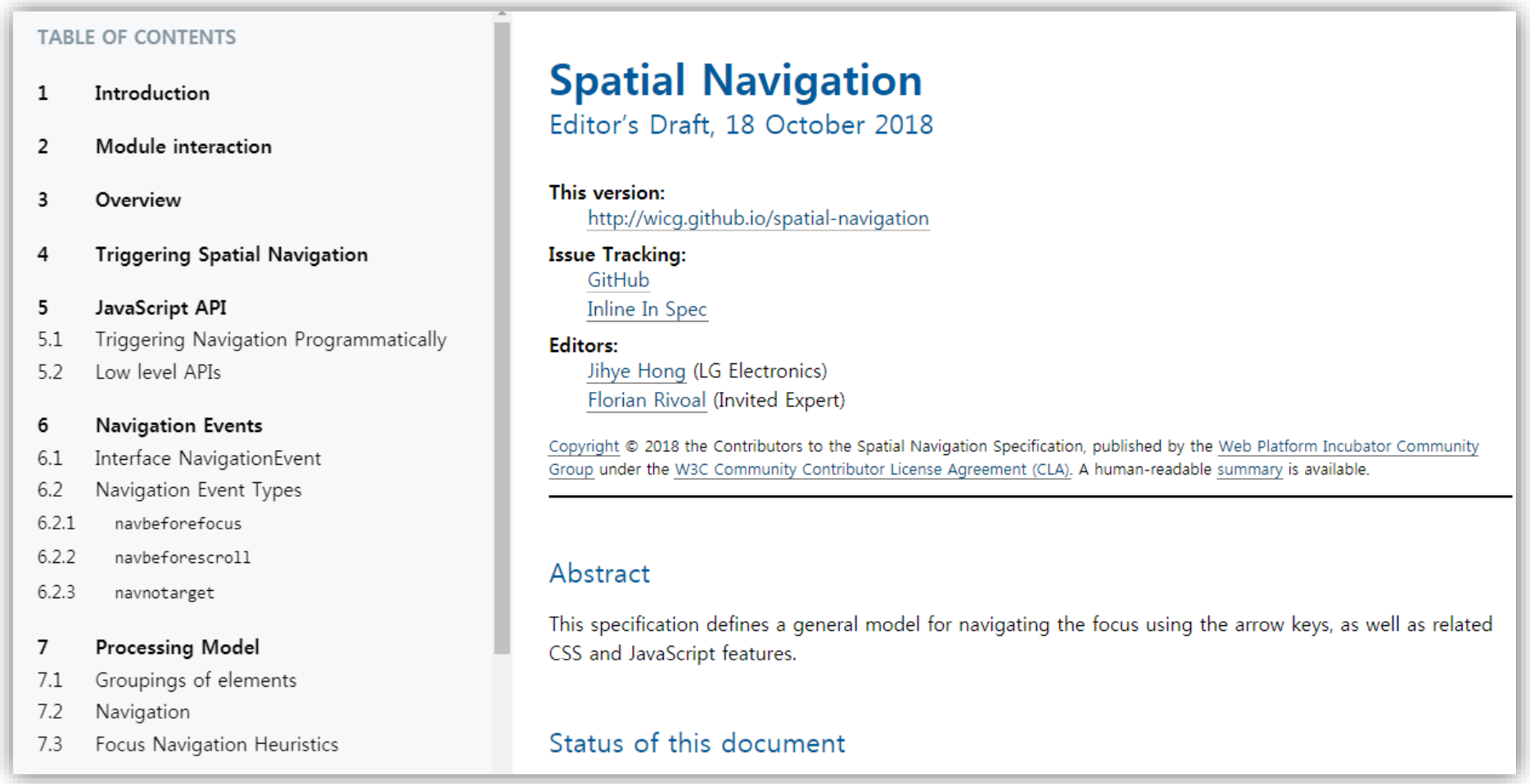


TABLE OF CONTENTS

Spatial Navigation

Editor's Draft, 18 October 2018

This version:
http://wicg.github.io/spatial-navigation

Issue Tracking:
GitHub
Inline In Spec

Editors:
Jihye Hong (LG Electronics)
Florian Rivoal (Invited Expert)

Abstract

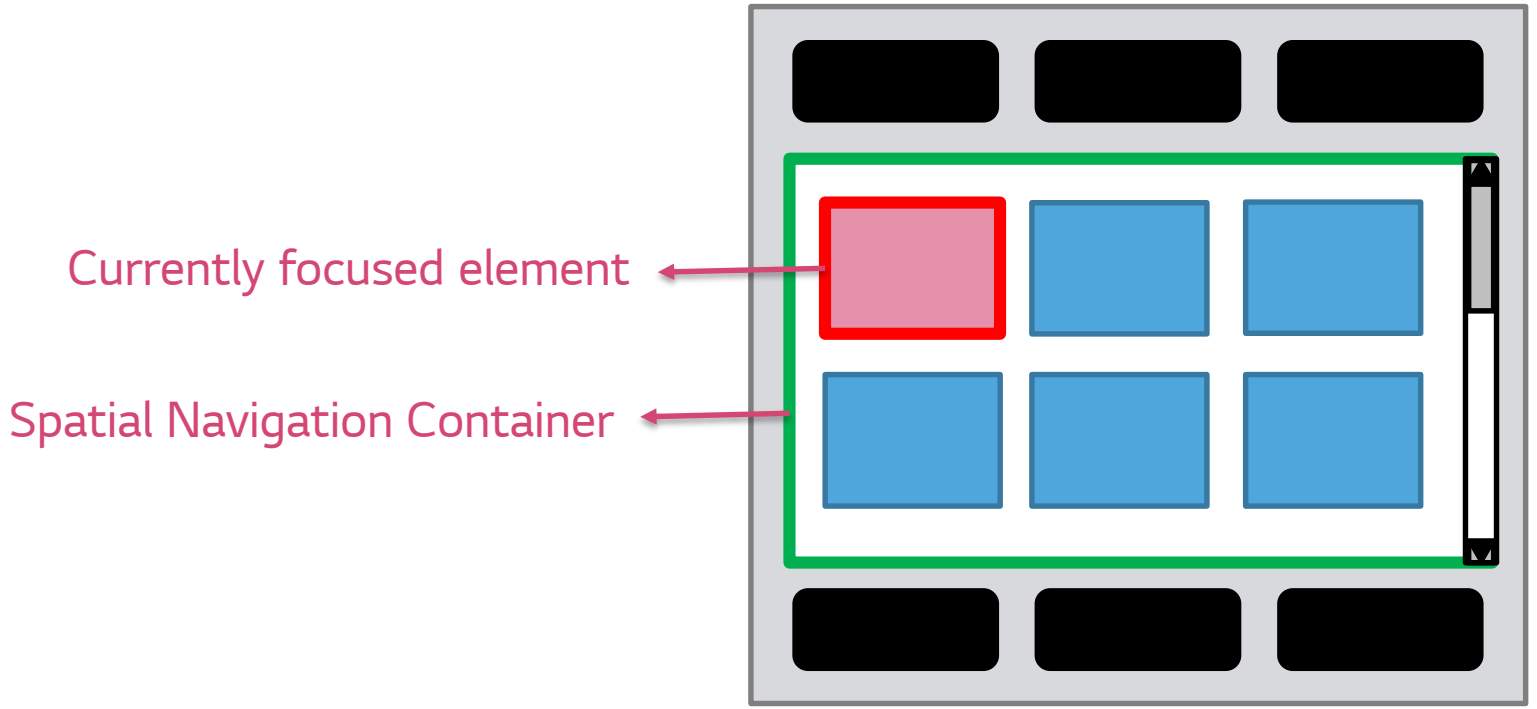
This specification defines a general model for navigating the focus using the arrow keys, as well as related CSS and JavaScript features.

Status of this document

https://wicg.github.io/spatial-navigation/

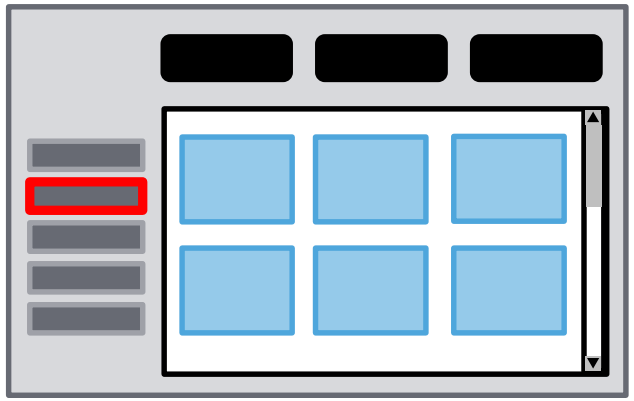
Spatial Navigation Container

- Scope for searching the focusable elements from the currently focused element
- Document, scrollable element (by default)

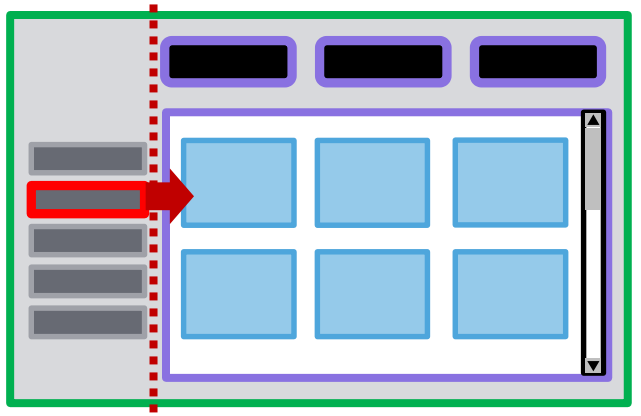


Basic Spatial Navigation Heuristics

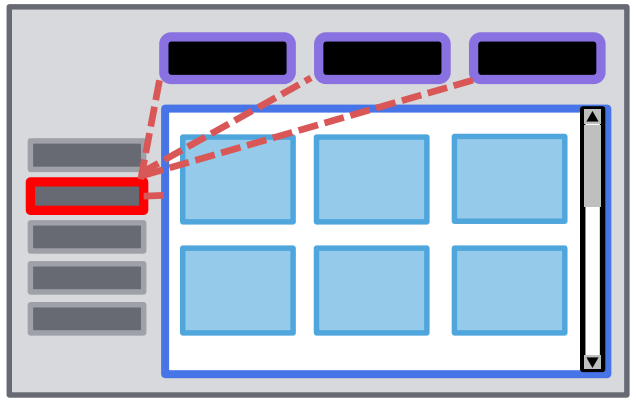
1. Pressing → from an element



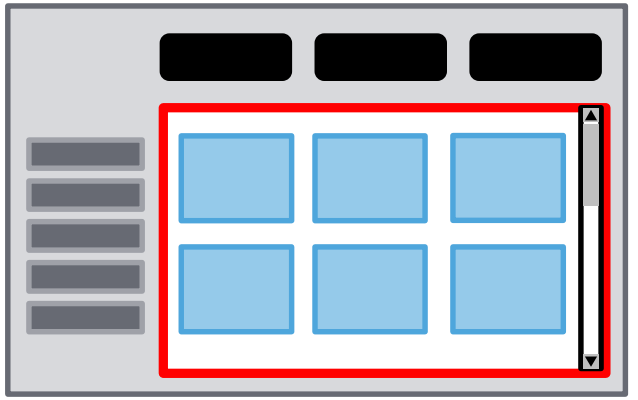
2. Find candidates within the spatial navigation container



3. Select the best candidate



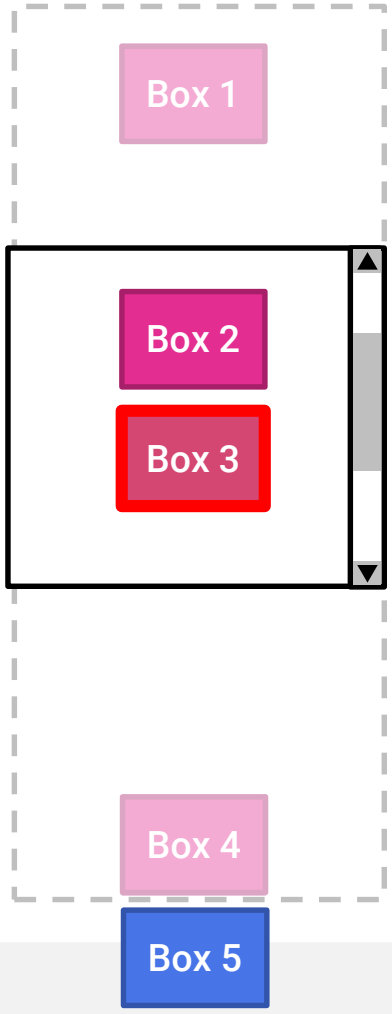
4. Focus the best candidate



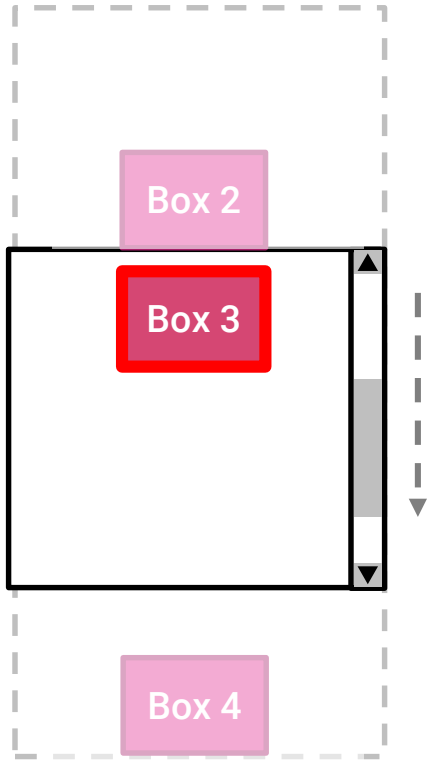
Spatial Navigation Heuristic Behavior

- Combining the UA-defined arrow key behaviors

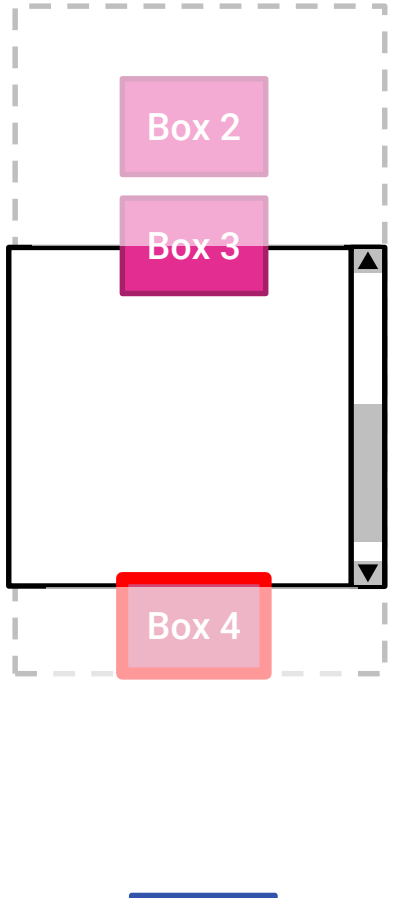
1. Box 3 gains the focus



2. Pressing triggers the scrolling



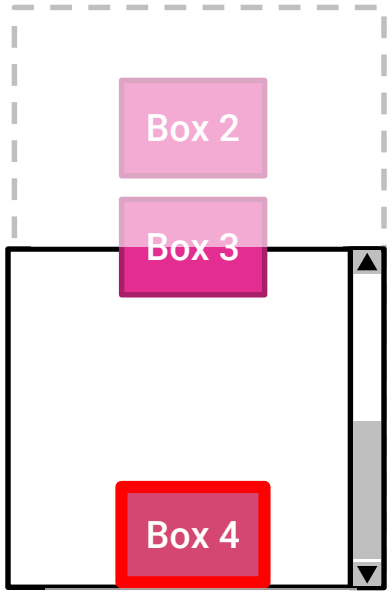
3. Box 4 gains the focus when it comes into the view



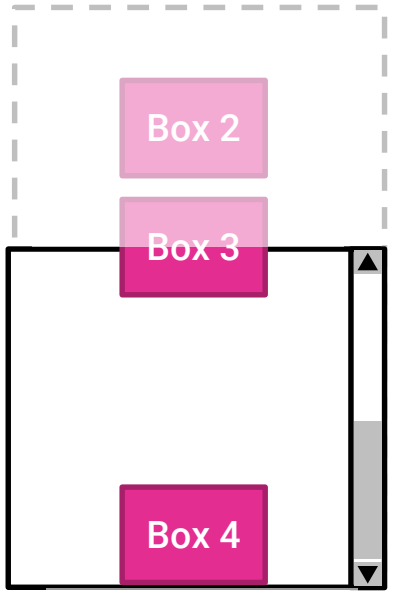
Spatial Navigation Heuristic Behavior

- Combining the UA-defined arrow key behaviors

4. Pressing ↓ moves down the scrollbar until the end



5. Pressing ↓ moves the focus out of the scroll area and Box 5 gains the focus

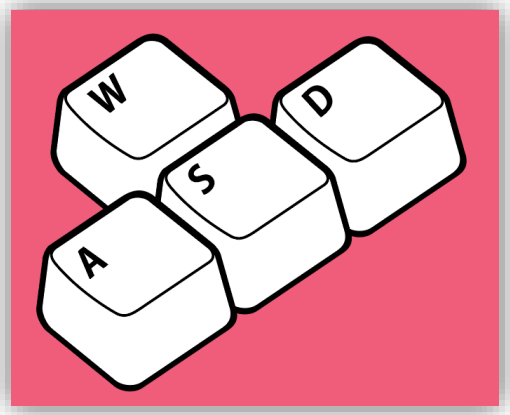


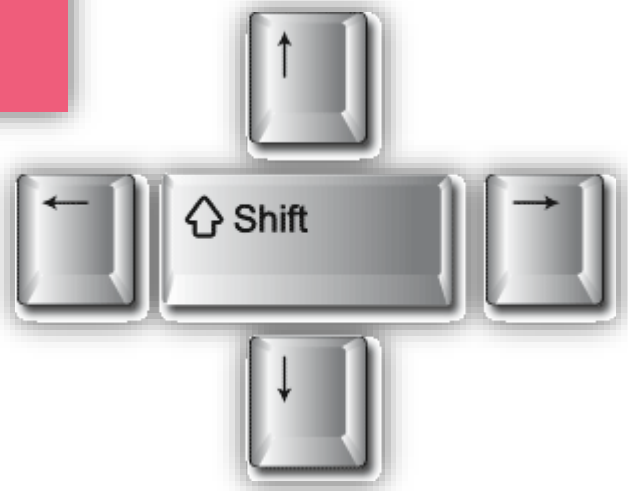
Box 5

Box 5

APIs

- window.navigate(direction)
  - Move the focus to the best candidate in the corresponding direction
  - Mapping other key combinations



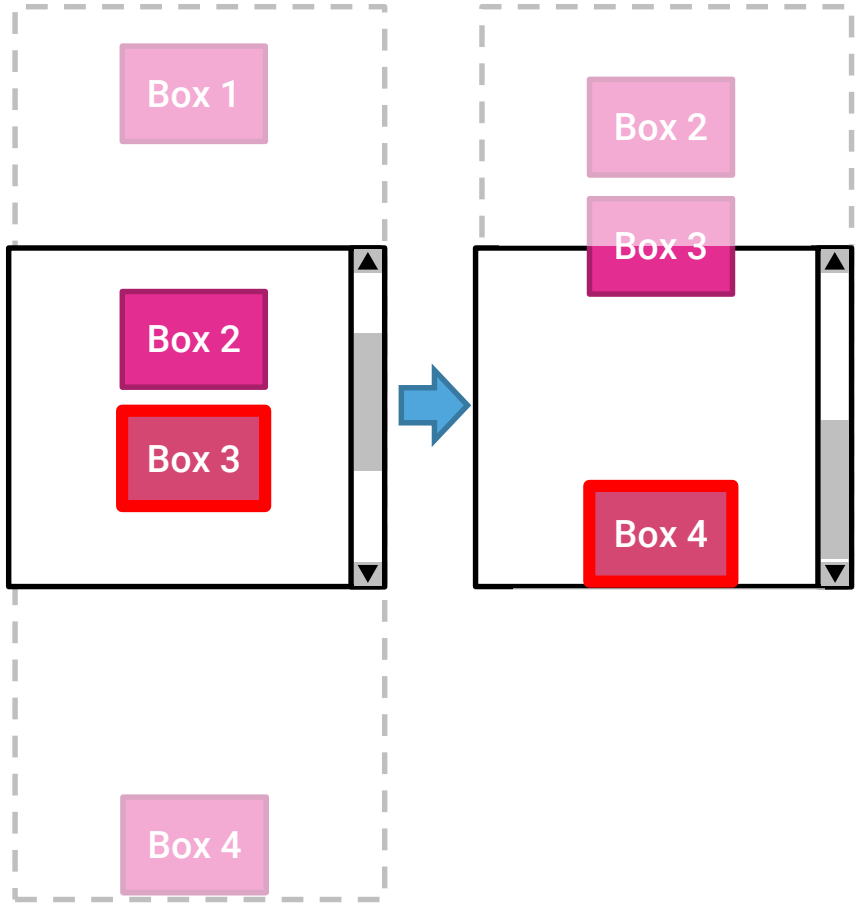



```
const redContainer = document.getElementsByClassName("container c1")[0];
redContainer.addEventListener('keydown', function(e) {
  const dir = WASD_KEY_CODE[e.keyCode];
  if(window.navigate && dir) {
    window.navigate(dir);
    e.preventDefault();
  }
}, true);
```

APIs

- element.focusables(option)
  - Finds all the focusable elements inside the container
  - Directly move the focus without manually scrolling

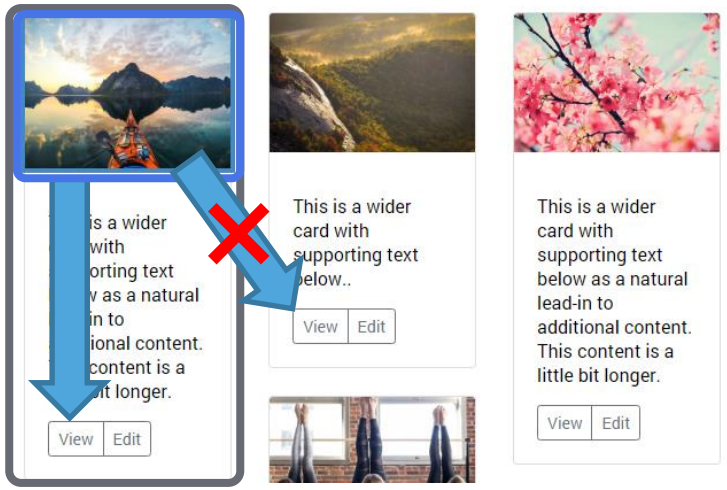
```
enum FocusableAreaSearchMode {
    "visible",
    "all"
};

dictionary FocusableAreasOptions {
    FocusableAreaSearchMode mode;
};
```



APIs

- Customize finding the best candidate
  - spatialNavigationSearch(), getSpatialNavigationContainer()
  - Navigate Events
  - spatial-navigation-contain CSS property

Infinite scroll



Grouped elements

APIs

Customize finding the best candidate

- CSS UI nav-* properties
  - Auto
    "The user agent determines which element to navigate the focus to in response to directional navigational input."

Spatial Navigation Heuristics

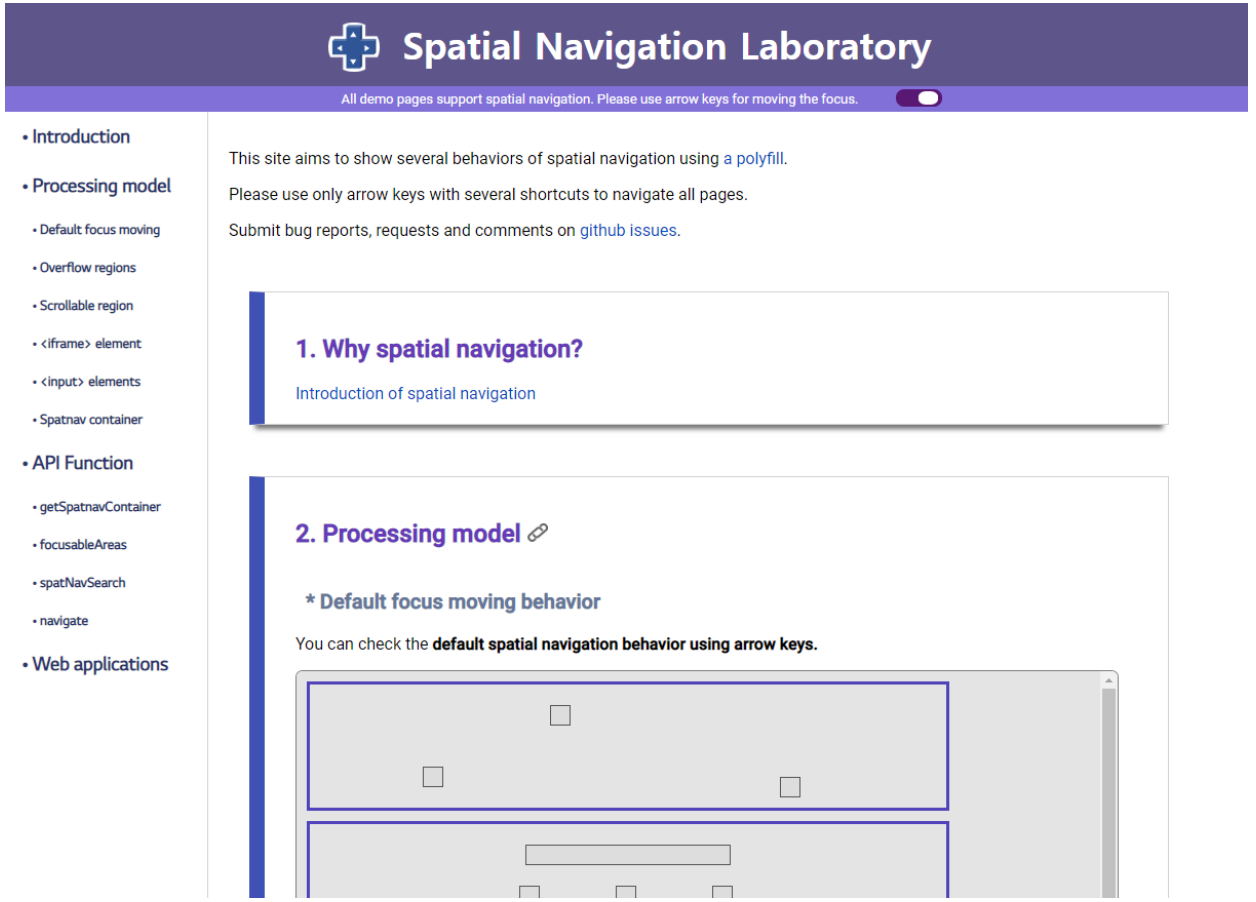
§ 5.3. Keyboard control

§ **5.3.1. Directional Focus Navigation: the 'nav-up', 'nav-right', 'nav-down', 'nav-left' properties**

Name:	***'nav-up', 'nav-right', 'nav-down', 'nav-left'***
Value:	auto | <id> [current | root | <target-name>]?
Initial:	auto
Applies to:	all enabled elements
Inherited:	no
Percentages:	N/A
Computed value:	as specified
Canonical order:	per grammar
Animation type:	discrete

https://drafts.csswg.org/css-ui/#nav-dir

Try the Spatial Navigation!



https://wicg.github.io/spatial-navigation/demo/

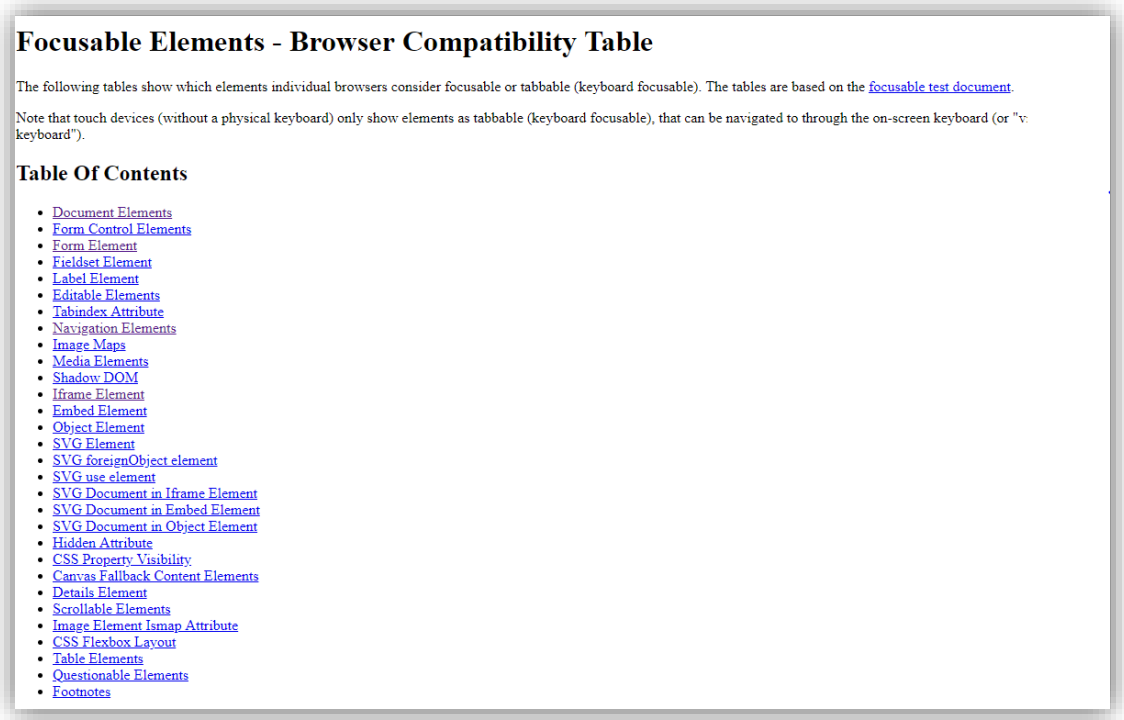
https://www.npmjs.com/package/spatial-navigation-polyfill

Next Steps

- Implementation
  - Google
    - Chromium Open Source, Accessibility
  - Vewd
    - Chromium Patch and Spec Reviews
  - Vivaldi
    - Investigating about applying the Spatial Navigation

Next Steps

- Spec Extension
  - Handling the various focusable elements
  - Improving the distance function
  - Managing the focus



Focusable Elements - Browser Compatibility Table

The following tables show which elements individual browsers consider focusable or tabbable (keyboard focusable). The tables are based on the focusable test document.

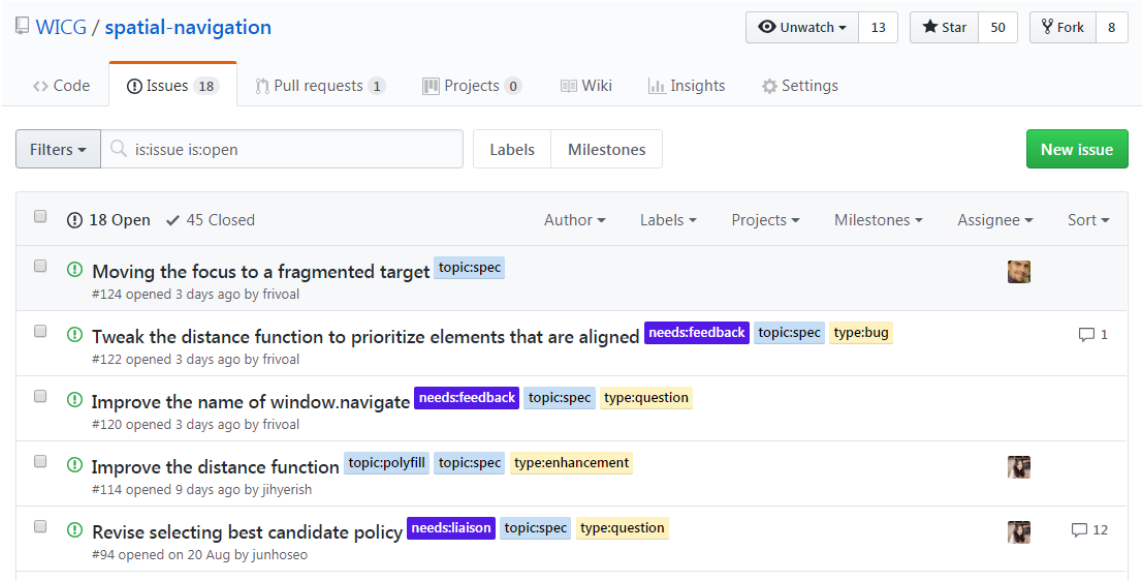
Note that touch devices (without a physical keyboard) only show elements as tabbable (keyboard focusable), that can be navigated to through the on-screen keyboard (or "v keyboard").

Table Of Contents

https://allyjs.io/data-tables/focusable.html

Discussion

- Open Questions
  - https://github.com/WICG/spatial-navigation/blob/master/explainer.md#open-questions
- Github issues
  - https://github.com/WICG/spatial-navigation/issues



Thank you! :)