Creating Accessible Mouse Pointer Interactions

In this module, you'll learn:

- The characteristics of accessible mouse pointer interactions.
- Why accessible mouse pointer interactions are important.
- How to ensure mouse pointer interactions are accessible.
- Best practices for creating accessible mouse pointer interactions.

What are the characteristics of accessible mouse pointer interactions?

Accessible mouse pointer interactions (also called pointer interactions, pointer gestures, or pointer events) are intended to help people with disabilities interact with an application without using complex mouse functions, such as multipoint or path-based gestures. When creating these interactions, a designer should try to:

- Provide the same functionality through multiple interaction options, one of which should be a simple pointer interaction.
- Provide predictable ways to interact and to stop, reverse, or undo an action as needed.
- Display all content, where possible, when an end-user clicks instead of using mouse hover.

Definitions:
Simple pointer interaction (also known as single pointer interaction): an interaction with a device that requires a single point of contact with an application.

- Examples: a single, double, or right-click (or tap), and long presses.

Multipoint gesture: an interaction with a device that uses two or more points of contact.

- Examples: two-finger pinching to zoom, two- or three-finger swipes or taps, and split taps where one finger rests while the other taps.

Path-based gesture: an interaction (usually single) that requires an end-user to follow a path to complete an action.

- Examples: drawing, sliding, dragging.

Why are accessible pointer interactions important?

Your role as a MURAL designer creates exciting opportunities to "experiment like an owner." Using accessibility guidelines, you're tasked with finding innovative and creative ways to support all collaborators, whether they're using a mouse, keyboard, touch, or assistive device.

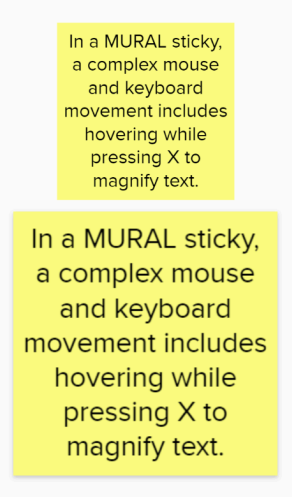
As you'll see through this module, many conventional mouse movements and pointer interactions pose challenges for people with disabilities:

- Complex mouse functions, such as multipoint and path-based gestures, are often difficult to initiate for end-users who have tremors, lack fine motor skills, and experience other disabilities.
- Many end-users with disabilities may not have the mobility or dexterity to perform actions such as sliding, dragging, and using simultaneous mouse-keyboard movements.
- Others may have a device mounted to a wheelchair arm in a fixed position that prevents more complex gestures.

Additionally, the manner in which content appears on the page can also become inaccessible:

- Content that displays on hover can be challenging for people with differing mobility, vision, and cognition.
- Tremors sometimes unintentionally cause motion when trying to hover over an object.
- Those with low vision may not see content that displays on hover if it's out of their field of vision.
- Others may have trouble remembering content once the hoverable information is no longer displayed.

Example: The screenshot below shows the MURAL pop-out feature, an example of a simultaneous mouse-keyboard movement, or multipoint gesture. The end-user must hover over the sticky note or textbox while pressing X to engage the pop-out. The pop-out is designed to magnify elements without zooming in.



How can I create accessible pointer interactions?

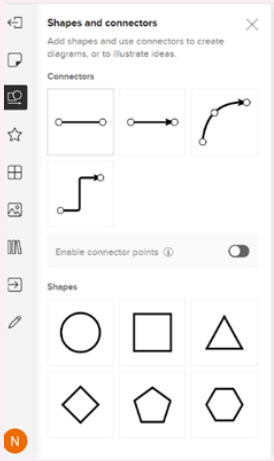
As a MURAL designer, you can use the following guidelines to create accessible pointer interactions:

- **As a first step, question whether dynamic content is appropriate for your design.** If end-users need information to complete a task, does it make more sense to use a persistent label or instruction in the UI?

    - If information is dynamically displayed on the screen, it's best to associate it with an intentional action by the end-user, such as activating or updating a UI element.
        - A designer should provide buttons for a user to click when, for example, an accordion opens and changes content on hover or a carousel moves based on hover.

- **Make sure end-users can perform every action, where possible, using simple pointers and *separate* keyboard commands.** Simple pointer interactions may require a designer to provide multiple ways of interacting instead of relying on a one-size-fits-all solution. In addition, using simple pointers and keyboard commands at the same time isn't accessible, so be sure these are used separately.

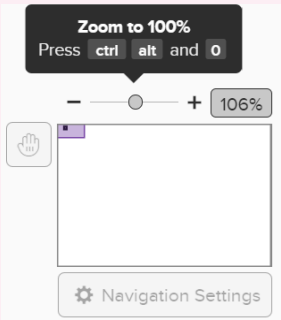
Example: Create a simple pointer solution by adding items to a context menu that opens with a single click or right-click, like in the example below. The MURAL shapes and connectors menu opens with one click and continues to appear without hovering over it.



    - Create as many simple pointer and keyboard command alternatives as possible.
        - Be sure end-users aren't required to use multipoint gestures like drag and drop or pinching to zoom elements.

Example: MURAL designers have created multiple ways for you to interact with a MURAL canvas. Though you can zoom in and out by pinching, the same result can be achieved with the simple gestures featured in the screenshot below.

You may click **[+] and [-] buttons** to adjust the scale or press **Control + Alt + 0 on a PC or Control + Command + 0 on a Mac**.

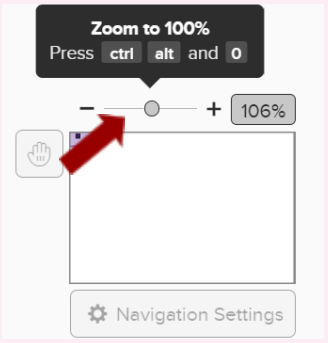


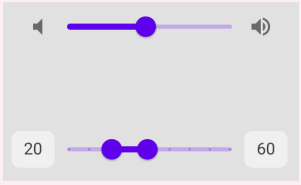
You can also drag elements on a canvas, but the same result can be achieved when pressing **Shift + the up, down, left, or right arrow keys** (depending on the direction you want to go). To use a simple interaction, click on the sticky note to move it across the screen using buttons rather than dragging.

- Provide alternatives to swiping or path-based gestures for elements.
  - One exception to this is the MURAL drawing tool. Draw mode enables collaborators to create their own widget using a path-based gesture. MURAL allows multiple ways to add images and content to a canvas that function as alternatives to draw mode. Draw mode would become an accessibility issue if there were no comparable alternatives.

Example: Continuing with the previous example, you'll see in the first screenshot below a closer look at the MURAL zoom slider. A common feature of sliders is the clickable **[+] and [-] buttons** on each end, which are similar to what's frequently used for volume control sliders, shown in the second screenshot.

However, sliders are even more useful when the entire slide is clickable, providing a larger target area. You can also allow the end-user to input numerical values to control an action.





- Provide alternatives to device motion. Motions such as shaking or tilting may be difficult for end-users with limited mobility or dexterity, as these motions can be inadvertently triggered or difficult to turn off. Always provide a clickable button or other conventional forms of interaction to achieve the same result.

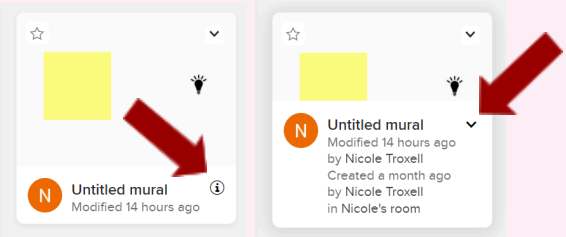
- **Design interactions that follow expected patterns so that end-users know how to interact with all items.**
    - Use common and familiar conventional controls to initiate pointer events, such as buttons, links, and inputs, that allow the user to perform a single pointer interaction.
    - Provide a means for pointer actions to be canceled or undone. Accidentally initiating a pointer event, especially when it can't be undone, can result in frustration.
    - Ensure that a pointer event is always initiated, as expected, on the down event (when the click is pressed) and aborted or canceled on the up event (when the click is released).

Example: One alternative to a task that requires an end-user to reorder items via **drag and drop** is to initiate the **sort function** with a long press of up or down arrows. The action is then undone when the press is released.

This feature is an alternative that can help collaborators reorder MURAL outlines in place of dragging items.

- **Create solutions that display all content whenever possible.** Always look for opportunities to have content appear with a click. Items that receive focus with a click are more keyboard accessible.
    - Build a responsive design that doesn't truncate text or require hovering and enables end-users to view all content.
    - When more content is necessary to the design, hover may be the only option. In these cases, the content should be displayed on hover until the end-user dismisses it.

Example: If you provide an information icon, make it clickable rather than hoverable. In the screenshots below, the first image features the information icon on MURAL cards in the MURAL dashboard. When you click (instead of hover over) the icon, as shown in the second screenshot, the information appears and stays present until you click the down arrow to dismiss it.



Resources:

- For more information on responsive design, see the **Creating Accessible Layouts** module.

- For information on text truncation, see the **Avoiding Text Truncation** module.
- For information on creating keyboard equivalents to mouse gestures, see the **Keyboards** module.
- Learn more about **Pointer Cancellation** on the Web Content Accessibility Guidelines (WCAG) website.
- Understand the requirements for hovering and dismissing content that appears on hover by visiting the WCAG **"Content on Hover or Focus"** page.

- **Design large target areas so that an end-user can easily click links, buttons, and controls.**
    - Design interactive elements to have minimum target sizes and touch targets that have an affordance of at least 44px by 44px.
        - Though the minimum recommendation from WCAG 2.2. is 24px by 24px, 44px by 44px helps prevent collaborators from accidentally clicking another element and takes into account different finger sizes.

Resource:
- Watch the video **"Large Links, Buttons, and Controls**" for more context on the benefits of creating larger target areas.

Best practices, in a nutshell

To create accessible pointer interactions:

- As a first step, question whether dynamic content is appropriate for your design.
- Make sure end-users can perform every action, where possible, using simple pointer interactions and keyboard commands.
- Design interactions that follow expected patterns so that end-users know how to interact with all items.
- Create solutions that display all content whenever possible.
- Design large target areas so that an end-user can easily click links, buttons, and controls.

Test your knowledge

Choose one or more answers to the following questions.

1. As a MURAL designer, you're tasked with creating an accessible alternative to drag and drop. What's the most accessible solution?
    A. Allow an end-user to initiate a pointer event when a click is released (also called the up event).
    B. Provide an alternative path-based gesture such as sliding.
    C. Create a smaller target area so that the end-user can grab elements more easily.
    D. Provide a clickable option, such as up or down arrow keys to move items.

2. One accessible alternative to using a slider is to:
    A. Provide a drag and drop option.
    B. Provide the option to input numerical values.
    C. Allow an end-user to use a drawing tool to accomplish the same task.

3. The most accessible option for displaying content in a design is to:
    A. Create hoverable content that continues to display once clicked.
    B. Truncate text not immediately relevant to the content on the page.
    C. Always display all content whenever possible.