Responsive Sizes:

Macbook Pro + iPad Mini (Horizontal)

Part 1:

The purpose of this website is to educate designers on machine learning libraries and methods to design for accessible interactions—a keyboard is often hard to use for those with disabilities that affect mobility, and hardware designed for accessibility is often expensive and uncomfortable. The information I am conveying on this website is a variety of experimental interactions that users can activate using their hands, faces, random objects around them, and sound. Users are encouraged to combine different interaction methods to replace using the keyboard with other interactions. To test these experimental interactions, users are prompted to move around and color a 20x20 module grid, engaging themselves with methods to create pixel artwork. The target audience is designers interested in incorporating ML in their interface designs to design for accessibility.

Part 2:

Using the panel on the right, users can activate a variety of interactions to move around and color the grid.

Interactions for moving around the grid:

- Keyboard (default): use the keyboard to move around the grid correspondingly
- Face: move your face up, down, left, and right to move around the grid correspondingly

Interactions for coloring modules:

- Keyboard (default): use keys 1, 2, and 3 to assign colors to the modules. Use the backspace bar to clear a module.
- Hands: hold up 1, 2, or 3 fingers to assign colors to the modules. Hold up 4 fingers to clear a module.
- Object: Hold up different objects to the camera to assign them to different colors. To assign/clear color to a grid, simply hold up the object to the camera.
- Sound: Clap, crinkle paper, or knock on the table to assign different colors to a module.

Part 3:

- ML5.js library: I used this library because it provides many ML visual detection methods that could be accessed through a webcam. This way, as long as users have a webcam through their computers, I incorporate ML models to recognize different objects and create interactions in my interface. From the ML5 library, I incorporated the Handpose, Facemesh, and ObjectDetection models to detect hands, faces, and objects and combine them with the interactions of movement and assignment. This adds to the different interactions I can incorporate into my interface.

- Google Teachable Machine: I used this library because I can train my own Visual and Auditory ML models to implement into a web application. With this, I can choose if I want to create visual or auditory interactions, and have the freedom to design my interactions. I used this to incorporate the auditory interactions in my website, training the models to be able to detect knocks, crinkling of paper, and claps. Implementing the Google Teachable Machine increased my opportunities to design for accessibility since I could train my own Visual and ML models.

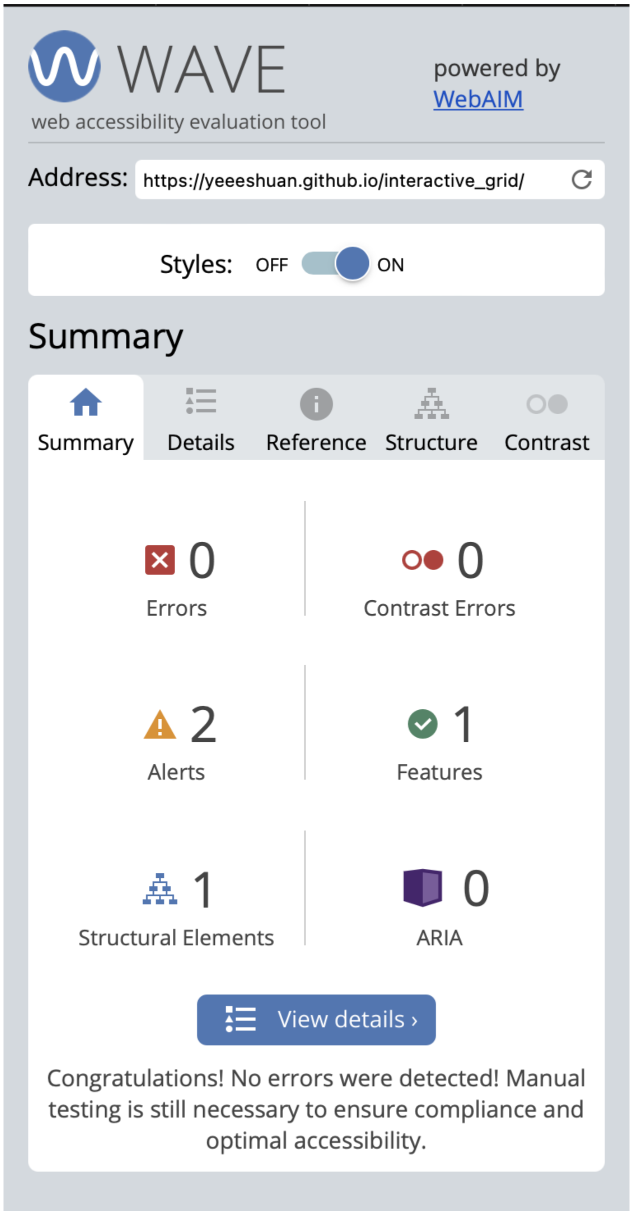
Part 4:

Many of my design changes happened thinking about the purpose of my interface. I was designing an interface meant to educate users and not a game. As a result, the idea to incorporate an information panel listing directions to navigate the interface and Machine Learning models used to create the interaction could allow users to understand how to incorporate the interactions into their websites. In addition, a big design change that I had to work on was the process of onboarding—many of these interactions are experimental, and users are not expected to know how to interact with the interface on first use. As a result, pop-up instructional panels were implemented to ensure that users always had a resource to consult in case they were confused about how to operate the interface.

Part 5:

The biggest challenge was to not lose sight of the main purpose of the website. While other interactions may add to how engaging the interface is, they may also be distracting, and distract users from learning how to incorporate ML into designing for accessibility. In addition, it was also difficult, but incredibly rewarding to think about how to design interactions for accessibility, utilizing auditory and visual interactions to communicate with the computer.

Appendix: WAVE - “Home Page”



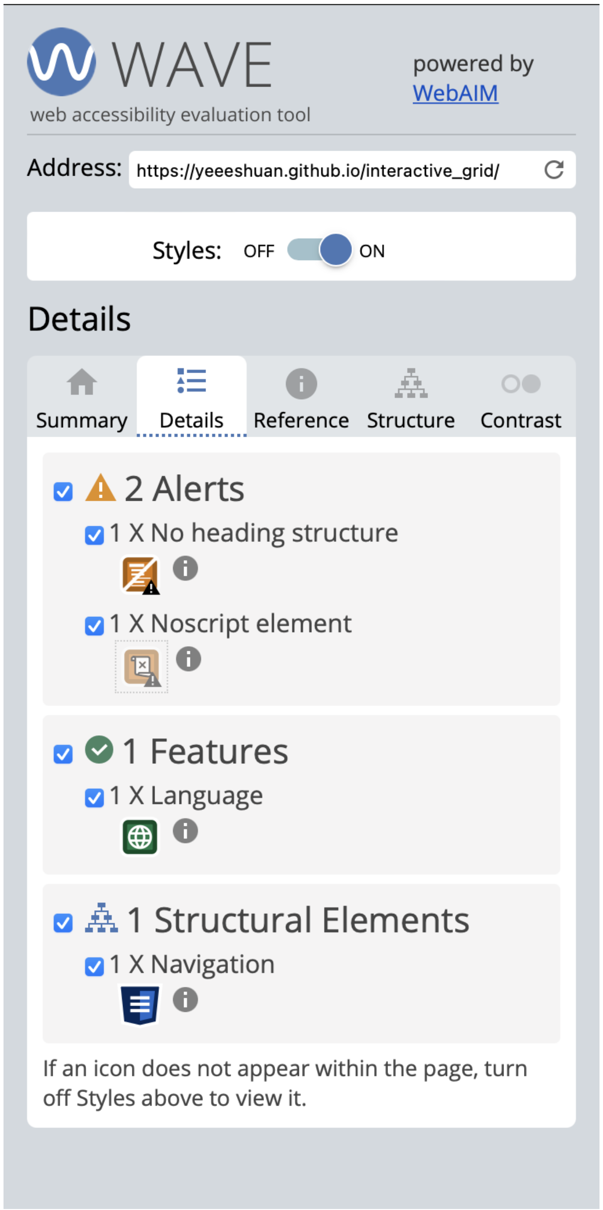
Address: https://yeeeshuan.github.io/interactive_grid/

Styles: OFF ● ON

Summary

☒ 0 Errors	0 Contrast Errors
⚠ 2 Alerts	✓ 1 Features
1 Structural Elements	0 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.



Address: https://yeeeshuan.github.io/interactive_grid/

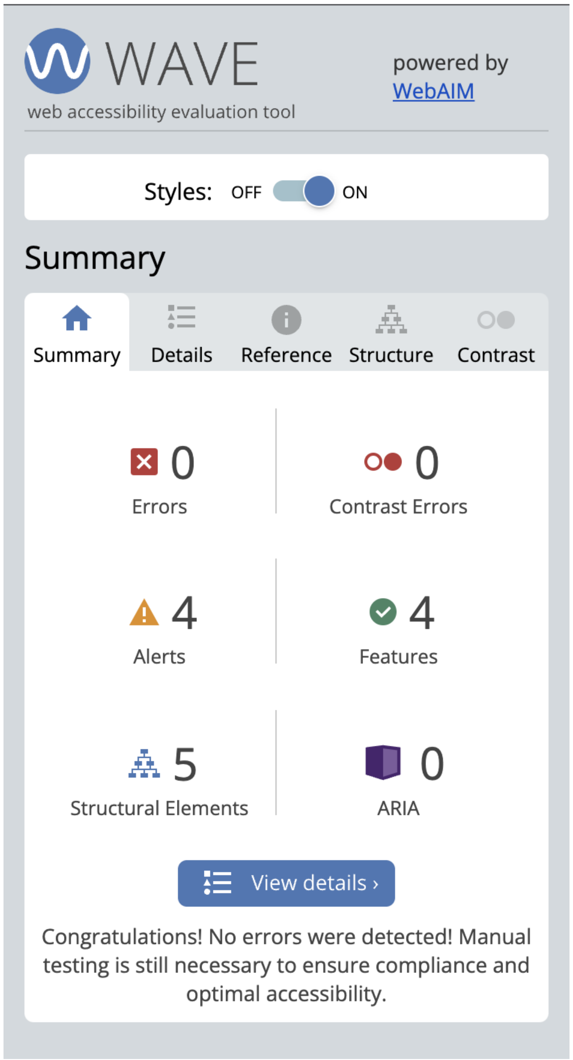
Styles: OFF ● ON

Details

- ☑ ⚠ 2 Alerts
  - ☑ 1 X No heading structure
  - ☑ 1 X Noscript element
- ☑ ✓ 1 Features
  - ☑ 1 X Language
- ☑ 1 Structural Elements
  - ☑ 1 X Navigation

If an icon does not appear within the page, turn off Styles above to view it.

“Demo Page”



WAVE
web accessibility evaluation tool

powered by WebAIM

Styles: OFF ON

Summary

0 Errors

0 Contrast Errors

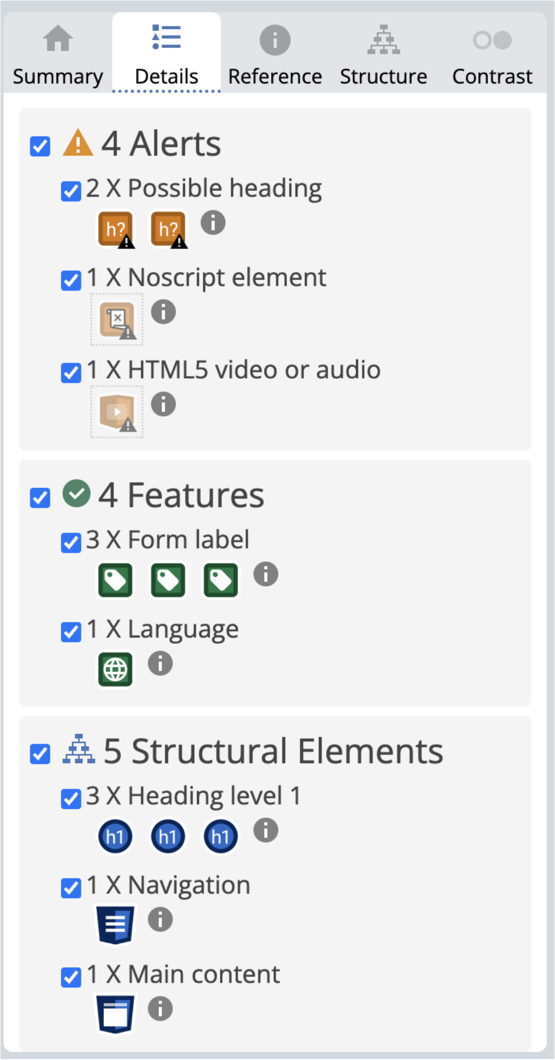
4 Alerts

4 Features

5 Structural Elements

0 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.



☑ 4 Alerts

☑ 2 X Possible heading

☑ 1 X Noscript element

☑ 1 X HTML5 video or audio

☑ 4 Features

☑ 3 X Form label

☑ 1 X Language

☑ 5 Structural Elements

☑ 3 X Heading level 1

☑ 1 X Navigation

☑ 1 X Main content