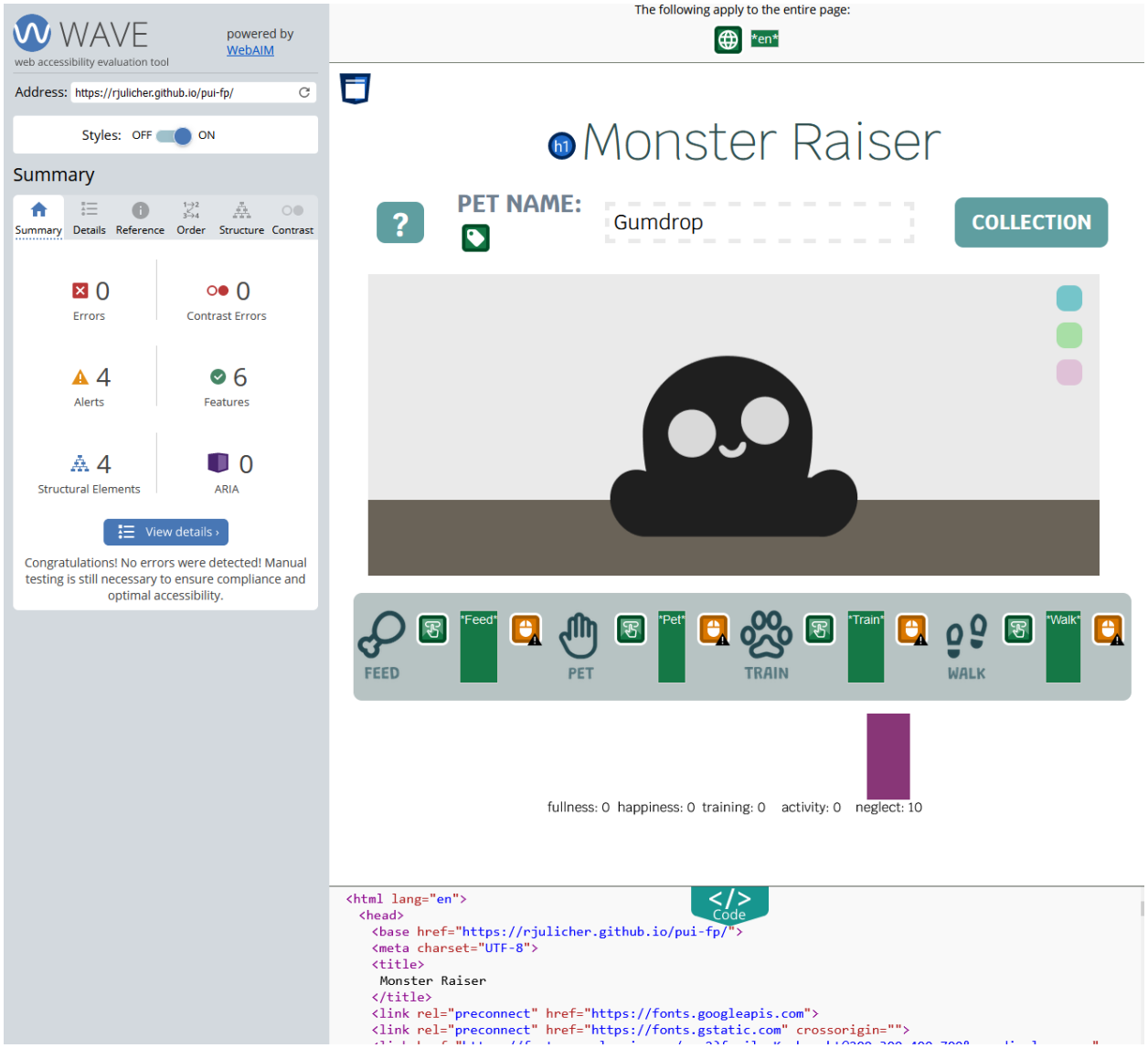
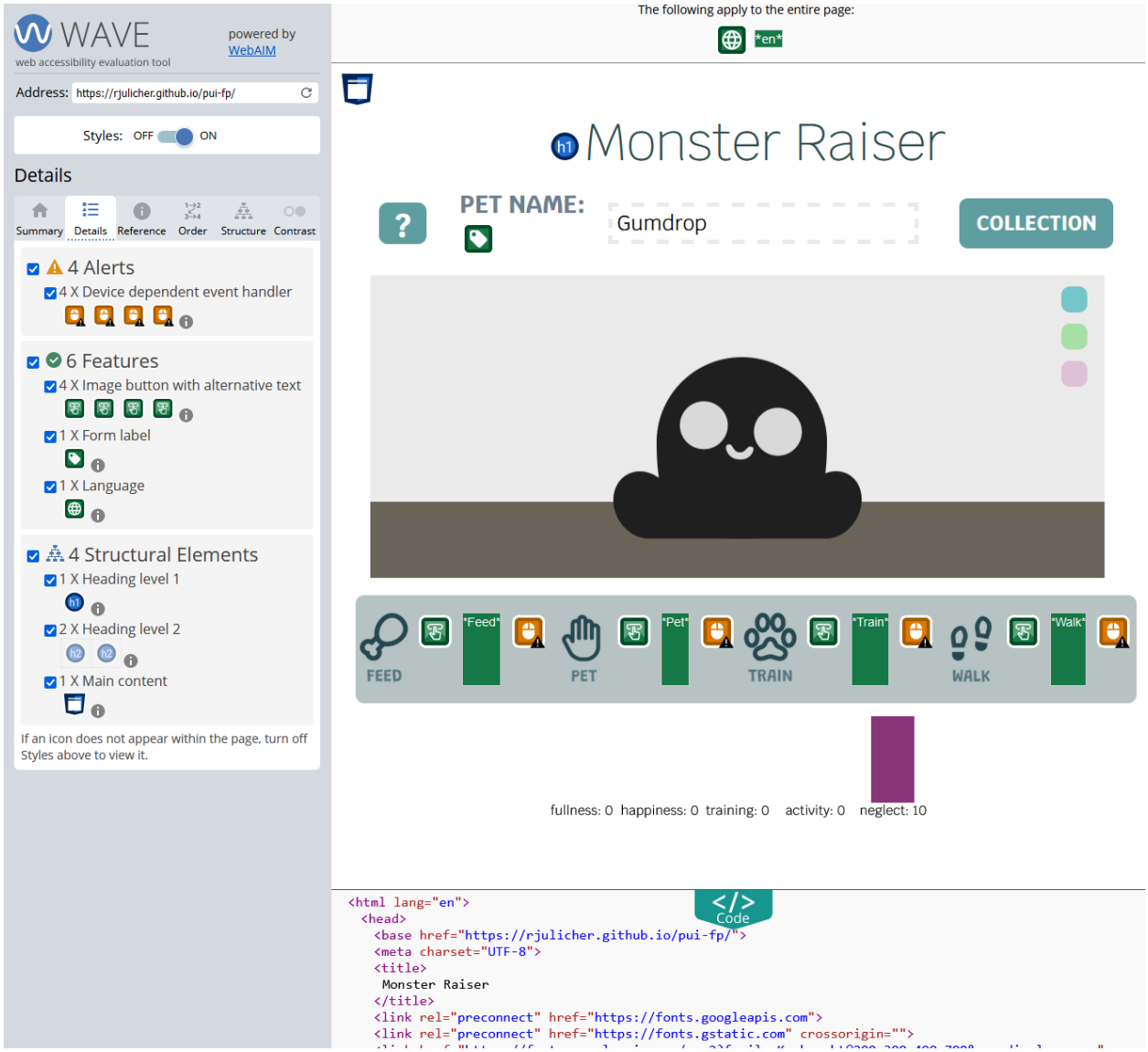
PUI Final Project Writeup
Rebekah Julicher

Screen sizes to test:

- iPad Mini - 768 x 1024
- 1200 x 1200

Accessibility screenshots:





Part 1:

The purpose of my website is to provide a fun, soothing browser game. I'm not conveying any important information, just the game elements and the ending flavor text. It is interesting and engaging because it is both cute and provides multiple different endings, each with their own monster type, and that combined with the naming, background colors, and monster collection features provide the user with an incentive to replay. It is specifically designed to be added onto in the future with more monster combinations should I have time to do so. The target audience is technically anyone with an interest in cute, simple monster games, but realistically that audience is reduced to primarily those who are able to read or have something available to read the ending text for them. Small kids can still have fun with the art, though.

Part 2:

- Help button
    - Creates a popup with helpful gameplay information
    - Select the “?” button on the top left of the screen
- Pet naming
    - Names the pet for the collection record
    - Select the Pet Name input box at the top middle of the screen, type a new or updated name
- Monster collection viewing
    - Displays all monsters currently in the collection
    - Select the “Collection” button at the top right of the screen
- Monster actions/stat increasers (feed/pet/train/walk)
    - Increases one of four out of the five total monster stats (fullness, happiness, training, activity) for weighing in the end screen
    - Select one of the four action buttons (feed, pet, train, walk) in the bar at the bottom of the main game screen, just above the stat bar graph
- Neglect
    - Increases the Neglect stat meter, which may affect the ending the user gets
    - Wait without using any of the main four monster action buttons
- Keep monster in collection
    - Saves a monster card to the Collection screen for later
    - Get the monster to the end screen (20 total action points on the graph) and select the “Keep” button at the bottom
- Restart
    - Restarts the game
    - Get the monster to the end screen (20 total action points on the graph) and select the “Restart” button at the bottom

Part 3:

- Name: Two.js
- I chose to use it because it seemed like a simpler library for 2D vector animations than WebGL, and it involved less setup than some others I checked out. I also wanted to try using a library actively being updated and hosted on GitHub to potentially offer feedback, documentation, or development assistance in the future to grow the number of projects on my portfolio.
- I used Two.js for my entire main animated screen (monster, wall/floor areas). It creates the graphics and keeps them animating and updating at constant framerates on every device without the need for odd html canvas workarounds or sleep cycles.
- It adds all of the main graphical content to my website while maintaining a relatively simple update loop in the code.

Part 4:

My original design did not include the top row of elements currently present in the game (title, help button outside the screen, pet naming, or collection). These were added after feedback. It

was originally planned to potentially be a Chrome extension, until I realized that making (and maintaining) an extension for a browser that is about to completely revamp how all of their extensions work (and which I no longer actively use personally due to recent events) is borderline a waste of time.
Originally this game was going to also have a room decoration feature, but this was scrapped due to feedback on that being an odd inclusion in its current state. The monster was also intended to have many more ending options, taking into account all stats and not just one.

Part 5:
The monster's decrease in ending results was due to the process of making unique art for all of these options being too much of a time commitment. Early on in the process, I also experimented with WebGL, which was scrapped due to involving a ridiculous amount of setup just to render a 2D square. I did have some issues for a while with the game screen not growing and shrinking properly when browser windows were resized, but I fixed that by applying checks in the update loop so that I could still keep persistent sprite animations on the page without constantly redrawing them.