

Martina Tan

PUI - HW8: Final Project Prototype

December 3, 2021

Treehugger - Companion Web Space

View the coded web prototype here: <https://martinatan.github.io/treehugger/>

View the original Figma prototype here:

https://www.figma.com/proto/jReRUpJYtkR3duqcqE3277/MartinaTan_PUI_HW7?node-id=4%3A3&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=4%3A3

YouTube Video Demo of Website

View video demo of desktop version here: <https://youtu.be/uzHvuPhWWjY>

Part 1: Project Description

My final project is an extension of and companion to my BFA Senior Thesis art project from this past year, titled *Treehugger* ([see overview of the project here](#)). Its purpose is to serve as a hub for viewing the outputs of this multimedia project, which has both physical and digital artifacts associated with it.

On the website is an interactive, rotating 3D scene representing the site of the original dead tree that inspired the community-sourced, paper-based activities of the project. The menu allows users to view the original installation performance video, as well as an artist biography and statement with some images. These overlays are set against the 3D scene to convey the feeling of being at the original site of the tree.

In theme with the project, the digital simulation-like nature of the web companion draws attention to the un-reality of the web and hopefully reminds the viewer of how the “real thing”, nature, should not fall to the wayside in our worldly priorities. This extra layer of reflection, on top of the many past layers that manifested during the original run of the project, should make the experience of visiting this web companion engaging in a new way even to people who visited the original exhibitions. Consequently, this website is targeted

towards both the audience that participated in my original thesis project, as well as new visitors who would not have otherwise been able to discover the project in-person.

Part 2: Website Interactions

3D Graphic Environment	Interaction Type	How to Reproduce It
	View the texture of the tree as it auto-rotates	Web/Mobile: Visit the main screen (the slow rotation is automatic)
	Rotate the tree trunk	Web: Click anywhere on the 3D environment and drag to the left or right. Mobile: Tap and hold anywhere on the 3D environment and drag to the left or right. You can let go to watch it spin freely before it slows down (like spinning a globe in real life).
	Zoom in and out on the surface of the tree	Web: Scroll mouse up and down or pinch in and out on Mac trackpad Mobile: Pinch in and out anywhere in the browser window
Main Menu	Interaction Type	How to Reproduce It
	View Main Menu (“View Performance” and “About the Artist”)	Web/Mobile: Visit the main screen (the menu displays by default)
HTML Overlays	Interaction Type	How to Reproduce It
	View performance video	Web/Mobile: Click/tap the “View Performance” menu item. Play the embedded YouTube video.
	Read artist statement and biography	Web/Mobile: Click/tap the “About the Artist” menu item. Scroll anywhere within the text area to read the entire thing.
	Close the “View Performance” or “About the Artist” overlays	Web/Mobile: Click/tap on the “x” icon in the top right corner of the screen. Or click/tap anywhere on the overlay to

		exit back to the main menu.
--	--	---

Part 3: External Tools

Name of Tool	Three.js (Javascript Library)
Why I chose to use it	I have a small amount of experience in 3D modeling using Cinema 4D as of last year, and have spent some time thinking about how I would incorporate this medium into the Treehugger project. I felt this would be a fun and appropriate next step to take the project further, and experiment with how I could translate a physical site into the web. It was recommended for me to use Three.js by a classmate to better convey the physical origins of my project in a web space, and my TA gave me pointers about how to do it that got me excited about intersecting these recent interests I was already fostering.
How I used it	I used the 'three.js' build file to load in the scene-building objects of WebGL and placed them on the webpage, with textures that would emulate the original physical environment this was inspired by. For example, the trunk of the tree is a cylinder that uses a panoramic image I captured of the original tree surface.
What it adds to my website	It adds another layer of interest to the theme of my project, which is all about considering humans' relationship to nature and how this differs from our relationship to technology. It's something cool to look at while someone views my artwork and thinks about its message.

Part 4: Iterations and Scoping

I originally had my site display everything in a pannable, "2-dimensional" screen, but because this lost the delight of interacting with a 3-dimensional space that earlier versions of my thesis project have had, I decided to try out three.js as a way to re-incorporate this 3-dimensional aspect again. Due to the project timeline, I was unable to incorporate the 35 individual submissions directly on the site like I originally wanted to, as this would require a lot of skill integrating three.js with React that I did not have the skills or experience to carry out. What I have come up with at the end allowed me to learn a lot about three.js as well as create a simple but

delightful hub that summarizes my work and compiles multiple channels for its components into one place online.

Part 5: Implementation Challenges

My biggest challenge with this project was scoping it to something manageable, despite my excitement about learning different technologies. I was unable to incorporate a lot of cool integrations with Three.js because they were based on React, and I had decided to experiment with Three.js first so it was difficult to translate my code into the React framework post-hoc. It was also difficult for me to translate my existing knowledge about 3D modeling into the web format, because as the objects I wanted to create became more complex (I originally wanted to attach individual planes to the “tree trunk” cylinder and host a clickable image in each one), the steps I would have to take to learn, implement, and debug these 3D objects became exponentially more overwhelming. Despite these challenges with understanding the system, I got a good enough grasp of it to remain excited about how powerful three.js and React can be, especially when used together.

WAVE Accessibility Check

See below for screenshots of the WAVE tool results. (Note that the website is also tab-focusable in a logical order, except that the controls for the 3D environment are not keyboard-accessible.)

The screenshot shows the WAVE web accessibility evaluation tool interface. On the left, the 'Summary' section displays the following data:

Type	Count
Errors	0
Contrast Errors	0
Alerts	5
Features	5
Structural Elements	10
ARIA	0

A note at the bottom of this section says: "Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility." Below this is a "View details" button.

The right side of the screenshot shows a portion of the Treehugger website. At the top, there are navigation icons for globe, user profile, and search. The main content area features a large image of a wooden pier over water, with the word "Treehugger" overlaid in a large, white, sans-serif font. Below the image are three buttons: "View Performance" with a blue arrow icon, and "About the Artist" with an orange arrow icon.

The following apply to the entire page:

WAVE
web accessibility evaluation tool

powered by WebAIM

Styles: OFF ON

Details

Summary Details Reference Structure Contrast

- 5 Alerts
 - 4 X Tabindex
 - 1 X YouTube video
- 5 Features
 - 4 X Alternative text
 - 1 X Language
- 10 Structural Elements
 - 1 X Heading level 1
 - 2 X Heading level 2
 - 2 X Heading level 3
 - 2 X Unordered list
 - 1 X Inline frame
 - 1 X Navigation

The page content shows a large image of a wooden wall on the right and a building across a body of water on the left. A prominent h1 heading "Treehugger" is centered. Below it are two buttons: "View Performance" and "About the Artist". A code editor icon with "h2" and "Code" is visible at the bottom.

The following apply to the entire page:

WAVE
web accessibility evaluation tool

powered by WebAIM

Styles: OFF ON

Structure

Summary Details Reference Structure Contrast

Main

- Navigation
- Treehugger

- Drag to spin
- ↔
- Treehugger Performance
- About the Artist

The page content shows a large image of a wooden wall on the right and a building across a body of water on the left. A prominent h1 heading "Treehugger" is centered. Below it are two buttons: "View Performance" and "About the Artist". A code editor icon with "h2" and "Code" is visible at the bottom.