The Last Wisps

An interactive storytelling website by Gabrielle Barrow

Overview

The Last Wisps is a short fictional interactive storytelling website. The narrative tackles themes of pollution and its effects on a fictional forest and its beings.

This site is targeted towards any and all audiences. It was made to be short and to the point.

Concept

The game includes five different character reactions, triggered when a user decides to click on one of "wisps."

Each character has a short dialogue detailing their experience with the forest succumbing to pollution and those who pollute and destroy it.

The story is **non-linear**, so therefore users can click on any character in any order.

The dialogues are short and meant to be a bit vague, for a quick experience where the user is somewhat left wondering what exactly happened, as well as enjoy the atmosphere and music.

Characters

User Interaction

As stated before, when a floating wisp is clicked, it triggers a character interaction:

When the first wisp is clicked, it triggers the site's main music to start playing. There are also sound effects for clicking the wisps as well.

* The song used for the website is called "Forgive Us Our Trespasses" by Ralph Lundsten. The wisp sound effect is a royalty free sound effect from the website freesound.org.

Technical Implementation

This website was created using:
- HTML
- CSS
- JavaScript
- GSAP

Some key features include:
- A list variable for the characters that was used to assign each character to a specific wisp later on with a for-each function
- Use of the JS library GSAP to create iterations for movement effects on each individual wisp
- Function created that changes the status of the ".hidden" feature, which displays none on the CSS file. After a character is clicked neither that character or wisp re-appears as the .hidden feature is removed

Challenges

Creating a function that made the wisps not reappear after they clicked ended up being the hardest. I used generative AI to assist in giving me a helping hand with that.

I could not figure out whether the best approach was giving the wisps a boolean value and switching them by assigning true and false to a feature that was visible and not visible was the move.

Ultimately, I ended up finding out it was much simpler: creating an if-then statement where if the wisp is clicked, a display of "none" is then added to its features, therefore making it non-visible after clicked for the rest of the time on the site.

```
closeBtn.addEventListener("click", () => {
    gsap.to(speechBubble, { opacity: 0, duration: 0.3, onComplete: () => {
        speechBubble.classList.add("hidden");

        // Hide character
        gsap.to(character, { x: "-500", opacity: 0, duration: 0.8, onComplete: () => {
            charaScene.classList.add("hidden");

            // Remove clicked wisp permanently
            const activeWisp = charaScene.dataset.activeWisp;
            if (activeWisp) {
                const clickedWisp = document.getElementById(activeWisp);
                if (clickedWisp) clickedWisp.style.display = "none";
            }

            // Bring remaining wisps back and restart wiggle
            gsap.to(wisps, {
                y: 0,
                duration: 1,
                stagger: 0.2,
                onComplete: startWiggle
            });
        }});
    }});
});
```

Thank you!