# Generative Grove

## Design Documentation

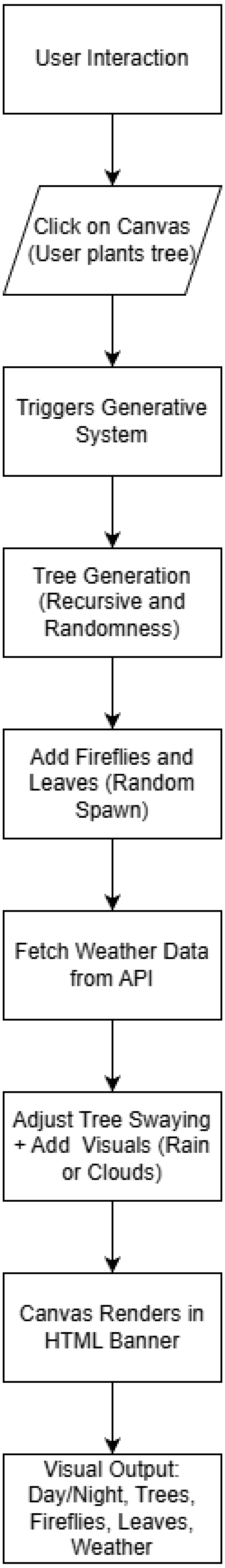
### Project Overview

Generative Grove is an interactive generative artwork designed as a banner for a built landing page. It uses p5.js to create a visually engaging and responsive experience that changes over time. The primary goal of this is to explore how techniques, user interaction, and real-time data can work together to create an interactive visual system. The canvas was embedded within the layout, integrated seamlessly within the page design and structure.

### Project Plan

Before starting the development of Generative Grove, I created a plan outlining the system's flow and structure. This plan shows the key parts needed for the generative artwork and how they would interact within the landing page.

The main elements included a responsive HTML and CSS landing page structure with a dedicated banner to hold the generative system. Within this container, I planned to use a p5.js canvas to render the generative visuals, ensuring that it would integrate cleanly into the webpage layout without disrupting other content.


User Interaction
Click on Canvas (User plants tree)
Triggers Generative System
Tree Generation (Recursive and Randomness)
Add Fireflies and Leaves (Random Spawn)
Fetch Weather Data from API
Adjust Tree Swaying + Add  Visuals (Rain or Clouds)
Canvas Renders in HTML Banner
Visual Output: Day/Night, Trees, Fireflies, Leaves, Weather

## Visual Elements

The generative artwork is created from several key elements that were used: a shifting background gradient, interactive trees that are created by the user, flickering fireflies flying, falling leaf particles, and weather responsive elements. The background gradient gradually changes over time, simulating the transition between day and night using a sine function and colour interpolation techniques.

Trees are the main interactive feature, users can click on the canvas to plant a tree, each of which grows recursively with randomised angles, thickness, and branch patterns. The recursive algorithm, combined with randomisation, produces fractal-like structures, resulting in organic and unpredictable growth for each tree.

The branches sway gently in response to Perlin noise, which creates a natural, smooth variation in movement. The system also uses real time weather data from the OpenWeatherMap API to use windspeed to influence the strength of the sway of the trees and can also alter some visual overlays based on the current weather condition, such as adding clouds.

Additional visual elements include the fireflies flickering and drifting across the canvas. These are not interactable but add a sense of atmosphere to the banner. Their brightness uses sine functions to simulate a natural glow. Leaves are generated from time to time during tree growth, falling with randomised speeds and fading over time to remove clutter. These all work together to create a more atmospheric and immersive visual experience that feels both dynamic and responsive to inputs.

## Techniques Used

The main generative techniques used in this assignment are randomness and noise. Randomness is used a lot in the tree generation, determining branch angles, lengths, and the probability of branching, as well as the behaviour of fireflies and falling leaves. Perlin noise was used to create the smooth and continuous movement of the branches, giving the trees a realistic swaying motion.

Supporting these techniques is recursion, which is fundamental to tree generation algorithms. Each branch is created by calling the branch function recursively, which results in complex fractal patterns. Parametric design principles are also used in the system, with variables such as windFactor, depth, and branchSpread influencing how the trees grow and respond to the environment.

Together, all these techniques create a system that is both organic and responsive, mostly driven by algorithms but shaped by the user input on the trees, and the environmental weather data.

## User Interaction and Data Integration

User interactions play a central role in this generative art’s design. By clicking on the canvas, the user can plant new trees, directly influencing the visual output of the banner. Each tree grows within its own set of random and recursive patterns, making sure that every user interaction results in a unique outcome, making each user's experience unique. This all fosters a sense of connection to the art, as it allows users to shape the evolving digital landscape of the banner.

In addition to user input, the system uses real time weather data through the OpenWeatherMap API. The wind speed affects the sway of the tree branches, while the weather condition such as Clouds or Rain introduces visual changes that reflect the environment's current state. This combination of user agency and real world data enriches the generative art and the experience, grounding it both in interactivity and contextual relevance.

### Design Choices

The design of the generative art piece is intended to show a sense of natural beauty, peace, tranquillity, and organic growth. The colour palette was carefully chosen to reflect natural colours, shifting from deep blues during the night to brighter tones during the day. The trees are created in browns and greens, while the fireflies' glow softly in shades of yellow, to add warmth and a subtle atmosphere to the scene. The use of gradients, subtle movements, and the layered visuals create an immersive atmosphere that evolves over time. The interaction of user input, randomness, recursions, and real time weather data, create a dynamic and unpredictable experience, ensuring that no two user sessions are the exact same.

## Reflection

### Debugging and Refinement

Throughout the development of Generative Grove, I ran into some challenges that required debugging, refinement, and optimisation. One of the biggest challenges was the behaviour of the trees themselves. During early coding and testing, the tree generation would sometimes result in the branches and leaves glitching out, behaving unpredictably, glitching colours, and not working properly. This was mainly due to some other part of the swaying code which had given me this issue when trying things out. To address this, I removed this swaying code, which used to sway the trees with mouse movement, leaving only weather data for swaying.

Another big issue I had was integrating the generative art into the webpage banner. Initially, I tried to use an embedded version of the p5.js sketch by copying the embed code from the p5.js online editor. While this approach worked to integrate it, I was not happy with the appearance as it introduced unwanted white borders around the canvas that didn't match the aesthetic of the landing page. The embed felt disconnected from the design and didn't allow for the level of customization I wanted.

To improve this, I decided to download the p5.js files and host them locally. However, this introduced new challenges: when I first tried to run the downloaded files, the sketch didn't work properly. There were issues with the p5.js version I used, and my code wasn't loading or rendering as expected. After troubleshooting, I realized that the best approach was to manually create a dedicated sketch.js file, paste my code there directly, and link it properly in the HTML. This way, I had full control over the code structure and could ensure the canvas integrated seamlessly on the landing page. I also replaced the p5.js file with a more up-to-date version, which helped resolve the issues. This process required trial and error, but it allowed me to embed the canvas cleanly and without unwanted borders, fully aligned with the page's layout.

Other challenges included issues where some trees failed to generate any leaves, likely due to the random chance in the algorithm. While most trees produce branches and leaves, some end up completely bare, which I intend to fix in future versions by introducing a minimum threshold for leaf generation. Additionally, an unintended behaviour remains where users can click anywhere within the canvas area, not just the visual tree area, to plant trees. This means that clicking outside the intended visual space still spawns trees, which I plan to address in a future update by adding proper click detection boundaries.

Overall, this debugging and refinement process has been good for improving the project. From resolving integration issues to stabilizing the recursive system and identifying areas for further improvement, these experiences have shaped the project into a more seamless and visually engaging interactive system.

## Critical Reflection

This project aligns with emerging trends in digital media, where interactive and dynamic content has become increasingly central to engaging audiences. Generative art offers a way to create living, breathing digital experiences that feel fresh, responsive, and deeply connected.

By integrating algorithmic processes with user interaction and real-time data streams, this project demonstrates how generative systems can create unique, changing visuals that are shaped both by their code and by the people interacting with them. This approach reflects changes in digital design and interactive media, where content is changing from being no longer fixed, but instead generated in response to users, environments, and any external data.

There are many potential applications of generative art in our sector of interactive digital media. Systems like Generative Grove could be adapted for interactive website design, immersive installations in galleries or public spaces at a better project stage, and dynamic visual components in AI-generated content platforms.

They offer a way to create engaging, non-repetitive visuals that encourage exploration and interaction. Generative art also has potential in commercial settings, such as creating responsive branding materials, data-driven visualizations, or interactive storytelling experiences that adapt in real time, which I intend to test with in the future.

Generative art can be an important role in shaping the future of creative industries. As technology continues to evolve, the ability to create systems that generate content dynamically opens new possibilities for designers, artists, and developers. *Generative Grove* contributes to this by demonstrating on a base level how randomness, recursion, and real-world data can combine to produce a visually rich and interactive digital environment. By bridging the gap between code, art, and user experience, projects like this showcase how generative art can transform digital creations into livelier systems that invite participation, creativity, and reflection.