The Neural Grove: POC Development Plan

This document outlines the architecture and assets required to build a functional Proof of Concept (POC) for the first 10 lessons of "The Neural Grove."

1. Core Objective

The goal is to create a simple, interactive application that demonstrates the core user experience: minimalist visuals, direct manipulation, and narrative-driven learning. It should validate the central hypothesis that complex concepts can be taught intuitively without traditional UI elements.

2. Technology Stack Recommendation

For a rapid, cross-platform POC, a simple web-based framework is ideal.

- **Framework:** A lightweight JavaScript library like **p5.js** or a simple game engine like **Phaser**. These are excellent for handling the required graphics, audio, and user input (tap, drag, hold) in a mobile browser.
- **Deployment:** The POC can be hosted on a simple static web page using services like GitHub Pages or Netlify, making it easily shareable via a URL.

3. Asset Generation

The following assets are required. I will generate these for you.

- **Visual Assets (SVG format):** Simple, scalable line-art icons.
    1. `seed.svg`: The initial dormant seed.
    2. `seedling_root.svg`: The seed with a small root.
    3. `seedling_sprout.svg`: The seedling with a root and sprout.
    4. `seedling_leaf.svg`: The seedling with a leaf.
    5. `seedling_flower.svg`: The seedling with a flower.
    6. `light_yellow.svg`: The gentle yellow light source.
    7. `light_red.svg`: The harsh red light source.
    8. `droplet_water.svg`: The blue water droplet.
    9. `tree_of_wisdom.svg`: The icon for the skill tree.

- **Audio Asset Specifications (for a sound designer):**

1. `chime_positive.wav`: Soft, single, major-key chime. ~0.5s.
2. `chime_triumph.wav`: Bright, harmonious chord. ~1.5s.
3. `buzz_negative.wav`: Low-frequency, dissonant buzz. ~0.5s.
4. `hum_low.wav`: Gentle, sustained low hum.
5. `plink_water.wav`: Clear, high-pitched water droplet sound.
6. `unfurl_plant.wav`: Soft rustling/stretching sound.
7. `whoosh_ambient.wav`: A soft, airy whoosh sound.

4. Core Logic & Game Loop (Pseudocode)

This describes the flow for the main application file (`app.js`).

```
// --- SETUP ---
// Global Variables
LET current_lesson = 1
LET seedling_state = { has_root: false, has_sprout: false, leaves: 0, has_flower: f
LET assets = LOAD_ALL_ASSETS()
LET narrative_text = ""

// --- MAIN GAME LOOP (updates every frame) ---
FUNCTION gameLoop():
  CLEAR_SCREEN() // Black background
  DRAW_CURRENT_SCENE(current_lesson)
  DISPLAY_TEXT(narrative_text)

// --- SCENE MANAGER ---
FUNCTION DRAW_CURRENT_SCENE(lesson_number):
  SWITCH lesson_number:
    CASE 1: draw_lesson_1()
    CASE 2: draw_lesson_2()
    // ...and so on for all 10 lessons
    DEFAULT: draw_title_screen()

// --- EVENT HANDLERS ---
FUNCTION on_tap(x, y):
  // Logic for what happens when the user taps, specific to each lesson
  IF current_lesson == 1:
    // Create light at (x, y), turn seed towards it
    // Advance lesson after a delay
  ELSE IF current_lesson == 3:
    // Check if tap hit the water droplet
  // ... etc.

FUNCTION on_drag(start_x, start_y, end_x, end_y):
  // Logic for dragging elements, specific to each lesson
  IF current_lesson == 5:
    // Check if user is dragging a light source to the seedling

// --- EXAMPLE LESSON LOGIC ---
FUNCTION draw_lesson_1():
  DRAW_ASSET(assets.seed, center_screen)
  narrative_text = "In darkness, there is stillness..."

  IF user_has_tapped:
    // Turn the seed towards the tap location
    PLAY_SOUND(chime_positive)
    narrative_text = "And there... a response."
    WAIT(3 seconds) // Pause before advancing
    current_lesson = 2
```

This structure separates the game state, drawing loop, and user input, which is a robust pattern for this type of interactive application.

The development plan is now complete. It provides a clear roadmap, recommended technology, a list of required assets, and the core application logic in pseudocode. This document, titled "The Neural Grove: POC Development Plan," serves as a direct blueprint for a developer to build the Proof of Concept.

The next logical step is to generate the visual and audio assets outlined in this plan. Shall I proceed with creating these assets for you?