Additional Tasks

Week 4 - Algorithmic ecologies: (non) media–ted landscapes:

Sensor viz works:

1. **Analyse the SensorViz workflow**

https://hcie.csail.mit.edu/research/sensorviz/sensorviz.html

a. As a group, examine the sensor viz workflow (figure 12).

b. Identify key stages:

i. **3d modelling & visualisation**

SensorViz's 3D modeling and visualization capabilities are central to its design, facilitating an interactive and iterative prototyping process. The tool integrates with Rhino3D and utilizes the Grasshopper plugin to create a 3D editor where users can import and manipulate 3D models of their prototypes. Once a model is loaded, makers can position sensors within the 3D space, adjusting their placement and orientation to assess coverage and functionality.

A standout feature is the real-time visualization of sensor data. As users modify sensor positions or prototype geometry, the system updates the visualizations instantaneously, allowing for immediate feedback which in turn because of the dynamic interaction helps in understanding how changes affect sensor performance and coverage,thus reducing the need for physical trial and error.

Additionally, SensorViz provides a library of visualization primitives tailored to different sensor types, such as hall effect sensors, distance sensors, and wind sensors. These primitives represent the sensing volumes and data outputs, offering a clear and intuitive understanding of sensor behavior within the 3D model. This approach enhances the design process by enabling makers to foresee potential issues and make adjustments before physical construction.

ii. **sensor data collection & processing**

The SensorViz project is designed to interpret complex sensor networks through real-time visualization, offering a structured approach to sensor data collection and processing. In the context of the Bionic Mycelium Network, examining its workflow provides key insights into managing environmental inputs, refining sensor data, and triggering interactive responses within the mycelium-inspired system.

The sensor data collection process in SensorViz relies on real-time ingestion of environmental information such as temperature, humidity, motion, and air quality. In a similar way, the Bionic Fungi Network will use embedded sensors to detect changes in moisture, air movement, and user interactions with artificial plants and fungi. These raw signals are continuously gathered and transmitted to a central processing unit, ensuring that the system maintains a responsive and dynamic interaction loop.

Following collection, data preprocessing ensures accuracy and reliability before visualization or activation of system responses. SensorViz applies filtering techniques to remove noise and normalize incoming signals, which aligns with the need for refining data within the Bionic Fungi Network before triggering LED patterns or motorized movement in artificial fungi. Efficient preprocessing reduces inconsistencies and enables a more seamless translation of environmental stimuli into visual and kinetic outputs.

The event-handling framework within SensorViz allows real-time mapping of sensor input to dynamic visual representations. This workflow is particularly relevant to the Bionic Fungi Network, where different sensor readings will correspond to distinct network behaviors. For instance, a drop in humidity detected by moisture sensors could cause LED pathways to flicker, simulating a stress signal traveling through a fungal network. Similarly, user-initiated signals through a mobile application or physical interface could alter the system's state, reinforcing the interactive nature of the installation. The visualization layer in SensorViz transforms raw data into engaging, intuitive representations, a concept that can be directly applied to LED-based information flows in the Bionic Fungi Network, making mycelial communication tangible and immersive.

iii. ar/mixed reality integration

The integration of Augmented Reality (AR) in SensorViz bridges the gap between digital models and the physical world, offering a spatial context that is crucial for effective prototyping. Utilizing the Fologram plugin, SensorViz synchronizes the 3D editor with AR devices such as tablets, smartphones, or head-mounted displays like the HoloLens. Through this synchronization, users can project their digital prototypes into real-world environments, aligning them with actual physical spaces.

This AR overlay not only visualizes the prototype but also displays sensor data in situ, allowing makers to evaluate how sensors interact with their surroundings. For example, before constructing a physical prototype, users can assess how sensors detect environmental variables in the intended context, such as a wind chime's response to airflow in a specific location and this capability aids in refining sensor placement and orientation to optimize performance and the AR integration facilitates the assembly process such that, once the physical prototype is built, makers can use the AR system to guide the placement of sensors, ensuring alignment with the virtual model. This step-by-step guidance reduces errors and enhances the accuracy of sensor installation.

iv. database structure

The database architecture in SensorViz is optimized for handling continuous, high-frequency sensor data streams. Given the real-time nature of the Bionic Fungi Network, a similar structure will be necessary to efficiently manage the relationships between user interactions, environmental changes, and system responses.

A time-series database is likely employed in SensorViz to store sensor readings efficiently while allowing for time-based queries. This format enables pattern recognition over time, making it ideal for tracking environmental fluctuations and user interactions in the Bionic Fungi Network. Each data entry is indexed with a timestamp, ensuring that the system can analyze trends and adapt its responses accordingly. For example, if multiple users send distress signals to plants in different sections of the installation, the system could prioritize those areas by amplifying LED responses or triggering more dramatic movement patterns.

The data retrieval system in SensorViz relies on structured API-based queries, allowing for quick access to relevant information. This principle can be incorporated into the Bionic Fungi Network's backend, where a RESTful API would fetch real-time sensor readings and determine corresponding responses for each plant unit. An efficient query system ensures that user interactions result in immediate feedback, enhancing the immersive experience of the installation.

To optimize performance, SensorViz likely integrates edge processing, where some data refinement occurs at the sensor level before being sent to a central database. This minimizes server load and enhances responsiveness. Implementing a similar approach in the Bionic Fungi Network could involve embedding microcontrollers that preprocess sensor data locally, reducing latency and ensuring fluid transitions between inputs and outputs.

By adopting SensorViz's approach to data collection, processing, and storage, the Bionic Mycelium Network can establish a robust and efficient system where environmental inputs and user interactions seamlessly translate into dynamic visual and kinetic responses.

2. Start identifying your project's implementation needs

a. discusses how your project aligns with or differs from sensorviz.

Our project aligns with SensorViz in its emphasis on sensor-based interactivity, real-time data visualization, and non-screen-based experiences. Both of the projects aim to create immersive, human-centered installations where environmental data drives dynamic, visible changes that are tangible and intuitive. However, while SensorViz focuses on aiding makers in prototyping sensor placements and behaviors within 3D models using tools like Rhino3D and Grasshopper our project prioritizes biomimicry, ecological storytelling, and a philosophical exploration of communication in natural systems and focuses our route response in that direction.

SensorViz is primarily a design aid for sensor visualization during prototyping, whereas our installation simulates a biological network with embedded meaning and educational value. Additionally, SensorViz utilizes AR for assembly and model-sensor alignment, when our installation uses physical, robotic flowers and LED lighting to communicate system responses. Our project leans more toward artistic, conceptual, and bio-HCI exploration, whereas SensorViz is more utilitarian in aiding technical prototyping workflows.

Thematically, SensorViz does not directly explore emergent intelligence, ecological systems, or embodied cognition, which are central to our project. In short, SensorViz is a visualization and placement tool, while our installation is a living, reactive system that seeks to evoke empathy and cognitive engagement with the natural world.

b. what are your core data sources, visualisation methods, and interaction modes?

Core data sources for our project will primarily be environmental sensors (e.g., soil moisture, humidity, temperature, touch light sensors, airflow, etc.), potentially paired with user-triggered inputs via buttons, capacitive sensors, or mobile interfaces. These sensors are in turn meant to simulate the biological stimuli that a mycorrhizal network would process in a forest environment.

Visualization methods include LED light pathways embedded in visible "root" structures beneath robotic flowers, as well as motorized movements of the flowers to convey states like stress, hydration, or nutrient transfer. This offers an intuitive, non-verbal and non-screen-based method of demonstrating data flow and communication.

Interaction modes are tactile and spatial. Users will engage through direct manipulation of the environment, such as watering a sensor patch or adjusting light exposure, and receive feedback via dynamic flower behavior and illuminated pathways to provide a real-time cause-effect loop, reinforcing user engagement and understanding.

Overall, the project emphasizes biomimetic visualization rather than abstract data representation, aligning with constructivist learning theories and embodied cognition, where understanding emerges through physical interaction.

c. list potential tools for each stage:
i. what software could you use for 3d modelling and visualisation? (e.g., rhino3d, unity, processing)

1. Rhino3D + Grasshopper: Ideal for modeling organic forms like flowers and root systems; offers parametric design features.
2. Blender: For creating high-fidelity models and simulating light behaviors in preparation for visual prototyping.
3. Unity: If you wish to explore an extended AR component or simulate behaviors before physical prototyping.
4. Processing: For simpler, real-time visualizations and prototyping of data-driven behaviors if Unity is too resource-intensive.

ii. how will you collect and store sensor data? (e.g., raspberry pi, arduino, cloud database)

1. Arduino: For real-time interaction with sensors (moisture, touch, light), controlling motors and LEDs.
2. Raspberry Pi: For centralized control, data logging, and potential cloud interaction if remote monitoring is added.
3. Microcontrollers like ESP32: Provide Wi-Fi capability for networked communication between plant nodes.
4. Cloud database (e.g., Firebase, Google Sheets): Optional if you want to store historical interaction data or analyze patterns.

iii. will you need real-time interaction or offline processing?

Offline processing could be used for data analysis post-interaction, helping refine behaviors or understand user engagement trends, but is secondary to the real-time system.

d. note any technical constraints (e.g., hardware availability, skill level, time).

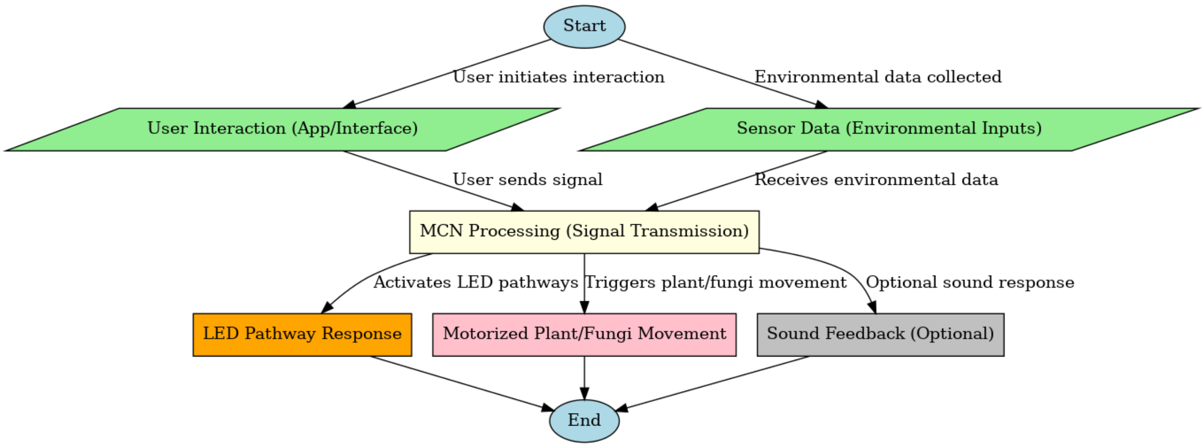
1. Hardware Availability: Limited access to high-quality actuators or specific sensor modules could constrain the complexity of flower movements or accuracy of data and the budget may limit how many plant nodes can be built and networked.
2. Skill Level: If the team lacks prior experience with microcontrollers, sensor integration, or robotics, initial learning curves may delay prototyping. Similarly, programming real-time interactions or tuning motors and LEDs to simulate lifelike responses could be technically demanding.
3. Time Constraints: The installation involves multiple subsystems, sensor integration, real-time control, mechanical design, and user experience where

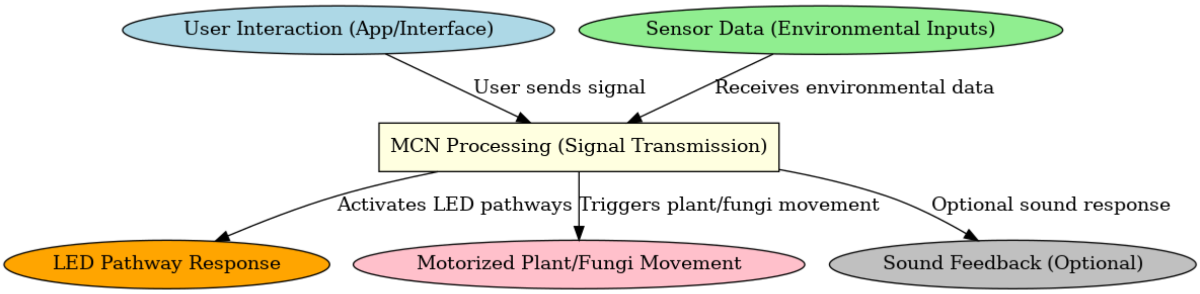
each requiring iterative development. Limited development time may restrict scope (e.g., fewer flowers, simpler movements, basic sensor types).

4. Environmental Constraints: If the installation is to be displayed in a dark room, ensuring reliable power supply and thermal management for sensors, LEDs, and motors becomes critical.
5. Mechanical Complexity: Designing robotic flowers with expressive motion while maintaining reliability over time may be difficult without access to CNC tools or 3D printing facilities.

3.Draft your system workflow
a. sketch a rough workflow diagram for your project, mapping out:
i. data flow from sensors to visualisation

The proposed system workflow begins with data inputs originating from two primary sources: user interactions via a digital interface (e.g., app or touchscreen) and environmental sensors capturing real-time changes such as light, moisture, or movement. These inputs converge in a central processing unit conceptually modelled after a Mycelium Communication Network (MCN) which functions as a biomimetic hub for interpreting and transmitting signals. From there, the processed data branches into multiple simultaneous output channels: visualizations via LED pathways, kinetic responses through motorized plant-like movements, and optional auditory feedback through sound elements. This multisensory output loop is designed to mirror organic, decentralized communication systems, with each component operating modularly yet cohesively. The system emphasizes a nonlinear, responsive structure where data flows fluidly from input to interpretation to output, supporting an immersive and biologically inspired user experience and from that these flowchart ideas have been proposed and designed



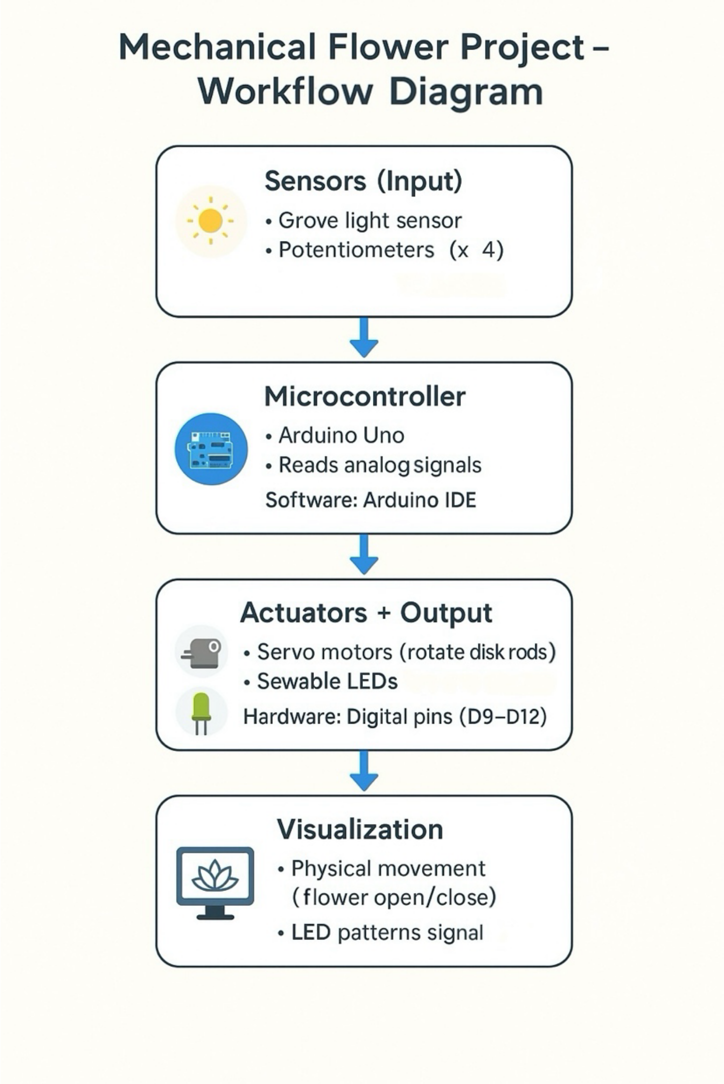


ii. software/hardware choices at each step

T he system begins by capturing environmental data using specific sensors tied to key plant-affecting variables: a DHT11/DHT22 sensor for moisture and temperature, a Grove light sensor for light levels, a pressure sensor for airflow and optionally, motion detectors for pollen/root migration with the ZE08-CH2O sensor to simulate chemical nutrient stress. These sensors interface with an Arduino Uno, chosen for its compatibility with analog and digital components and ease of integration with environmental sensors.

The Arduino transmits raw sensor data via serial communication to a Raspberry Pi 4, which acts as the central processing unit. Here, Python-based logic interprets the signals using libraries like PySerial (for communication), Guizero (for GUI rendering). The processing logic mimics fungal information relay, categorizing signals into stress/resilience modes that determine feedback pathways.

From the Raspberry Pi, output signals are routed to actuators and display modules. LED strips (NeoPixels) embedded in embroidered root-like textiles provide a visual representation of data flow. These are controlled using the Adafruit NeoPixel library through Arduino or Pi GPIO pins. Servo motors embedded in 3D-printed flower joints simulate plant movement in response to data changes, giving a kinetic output. A touchscreen would provide a user interface to manipulate resource values via physical potentiometers connected to the Pi's GPIOs, which adjust environmental parameters in real-time and the touchscreen's compatibility with the Pi or arduino is yet to be determined but will be experimented with



iii. potential gaps or open questions

How to include the nutrients and motion as part of the installation and is that even possible to represent? Why so? What can replace them?

4. Reflection & next steps

a. What key decisions still need to be made?

Finalizing the resources and their sensors and then starting to explore the Led protoyping and also working more on petal iteration and talks with specialist departments on campus

b. What are the biggest technical unknowns?

Ui, led and motor integration between the raspberry and arduino and flower 3d joints due to material constrictions and shortcomings and also what kind of screen to use to display the UI and the frame boarding for it

c. What resources or tutorials might help refine your plan?

Talk with sculpting department, pi and arduino forums and YouTube ‘serialing’ tutorials and more iterations of flower design pulling inspiration from Pinterest, artist research, inspiration projects like HCI, bionics and networks.