Philip Beesley Studio Inc.
www.philipbeesley.com
Living Architecture Systems Group
www.lasg.ca

ISBN 978-1-988366-75-3
Riverside Architectural Press

Aether

Spatial Behavior Software Suite

MATT GORBET, KEVAN CRESS & PHILIP BEESLEY
LIVING ARCHITECTURE SYSTEMS GROUP

Aether

Spatial Behavior Software Suite

Matt Gorbet, Philip Beesley & Kevan Cress
Living Architecture Systems Group

Publisher: Riverside Architectural Press | www.riversidearchitecturalpress.ca

Title: Aether | Spatial Behavior Software Suite
Names: Gorbet, Matt, 1973-author. | Cress, Kevan, 1995-author. | Beesley, Philip, 1956-editor. | Living Architecture Systems Group, issuing body.

Description: This document describes the inner workings of Aether, a comprehensive browser-based software suite enabling the composing of behaviours of various technologies within a Living Architecture Testbed

Identifiers: ISBN 978-1-988366-75-3

Design and Production by Living Architecture Systems Group

Publication: Feb 2026
Riverside Architectural Press
7 Melville Street
Cambridge, Ontario, N1S 2H4, Canada

All images authored by Philip Beesley Studio Inc. / Living Architecture Systems Group

Errors or omissions would be corrected in subsequent editions.
This book is set in Garamond and Zurich BT.

Canada Council for the Arts | Conseil des arts du Canada
Social Sciences and Humanities Research Council of Canada | Conseil de recherches en sciences humaines du Canada

Contents

Aether
Living Architecture Software Suite

Living Architecture installations behave in expressive ways that can be controlled using simple software interfaces. **Aether: Living Architecture Software Suite** can support many different kinds of Living Architecture constructions. *Aether* can be used for applications including small kits explored by young users, sculptures and environments composed by artists, and test-bed environments studied by scientists.

Like diverse species within an ecosystem, objects and devices can be connected by using *Aether* within a physical installation. Physical devices connected with *Aether* software infrastructure enable Living Architecture installations to behave in lifelike ways. These coupled arrangements can be controlled and configured using graphical interfaces. The software system can create a virtual ecosystem populated with invisible influence fields, virtual beings and digital twins of the physical installation. Spatialized soundscapes can be generated to further animate the space. These digital and physical worlds can exchange information, communicating and mutually influencing each other.

When sharing space with a Living Architecture installation, kits designed for exploration and other external systems can join this digital ecosystem. Interfaces can be tailored to users with different experience levels. These can allow for easy adjustment of parameters that control behaviour. Deep connections can also be found that control underlying algorithms, allowing users to design new kinds of expressions and experiences.

Open-source licensing is being developed to support collaborative development of this software. Living Architecture Systems Group studio members can work with new collaborators to support access and further development.

Software Suite Features

A **browser-based software suite** enables control, configuration and composition of behaviour within installations. These constructions can include temporary experiments and kit constructions, sculptures, and permanent testbeds.

Simple interfaces accommodate a broad range of users. Schoolchildren, collaborating artists, and research scientists can explore, experiment, and add to evolving installations.

Ability to pair the testbed's expressive behaviour with **external devices** and systems like robotics, lighting, games and VR

Instructor-led exploratory workshops can make learning and using the system easy and fulfilling, addressing educational goals in Science, Technology, Engineering, Art and Mathematics (STEAM).

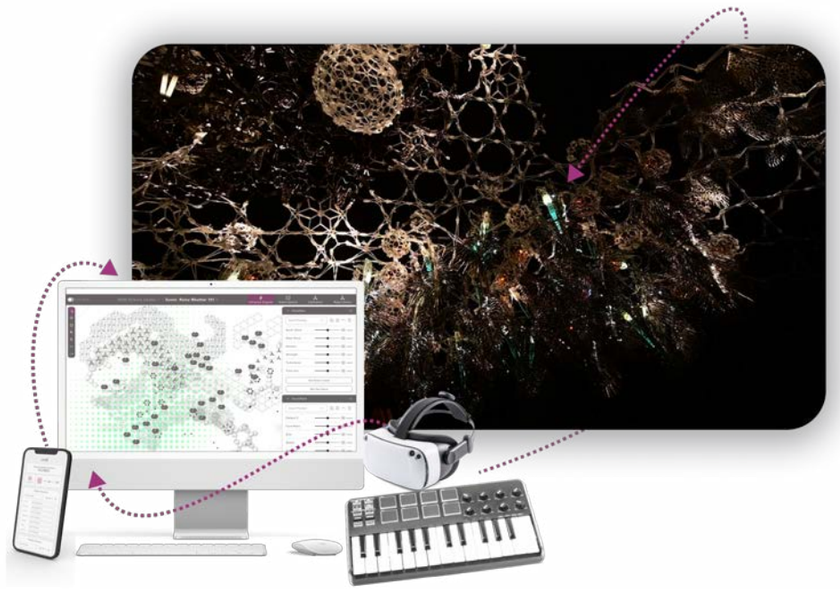
The *Aether* Software Suite can be made available under an Open-Source License. Contact the Living Architecture Systems Group for detailed licensing information.

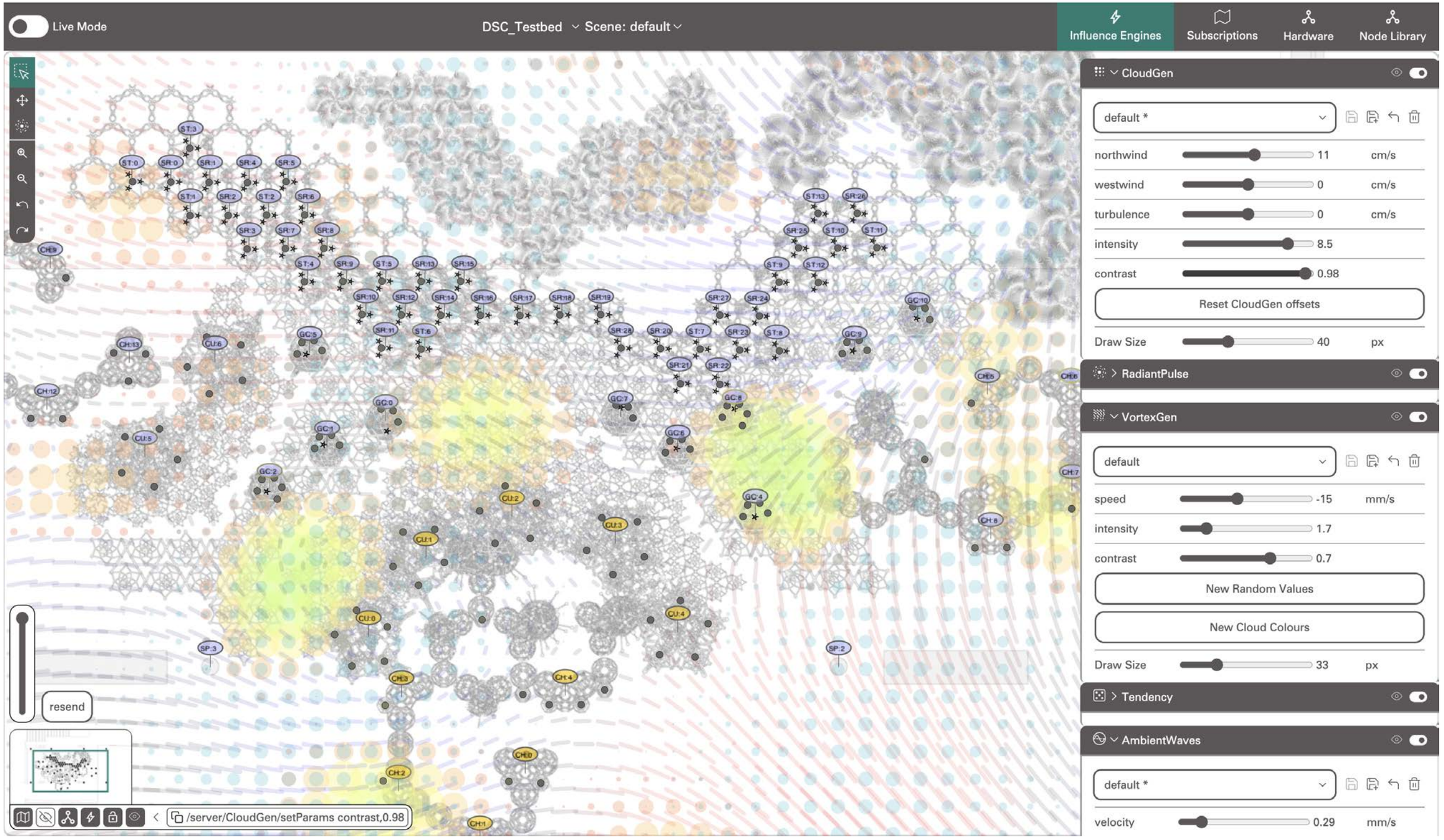
Below

Users will be able to access Aether on desktop or mobile while composing the testbed's behaviours and interfacing with various external devices

Following Page

Aether's web interface for tuning behaviours, virtual clouds of influence drift through the testbed affecting a multitude of actuators




Live Mode
DSC_Testbed
Scene: default
Influence Engines
Subscriptions
Hardware
Node Library
CloudGen
default *
northwind 11 cm/s
westwind 0 cm/s
turbulence 0 cm/s
intensity 8.5
contrast 0.98
Reset CloudGen offsets
Draw Size 40 px
RadiantPulse
VortexGen
default
speed -15 mm/s
intensity 1.7
contrast 0.7
New Random Values
New Cloud Colours
Draw Size 33 px
Tendency
AmbientWaves
default *
velocity 0.29 mm/s
resend
/server/CloudGen/setParams contrast,0.98

Behaviour Systems

The software, hardware and interfaces that surround a Living Architecture testbed enable engagement with the interactive behaviour systems that animate these works of Living Architecture. This testbed infrastructure supports a Spatialized Digital Milieu; a virtual ecosystem populated with influence fields, virtual beings and digital twins of the physical testbed. The digital and physical worlds exchange information, communicating and mutually influencing each other.

When sharing space with the Living Architecture testbed, Integrated Kit Assemblies and external systems can join this digital ecosystem. Interfaces tailored to users with different experience levels allow for simple adjustment of parameters in the digital milieu, or deeper connection with its underlying algorithms.

Left

Living Architecture Behaviour systems create dynamic responsive environments

Below

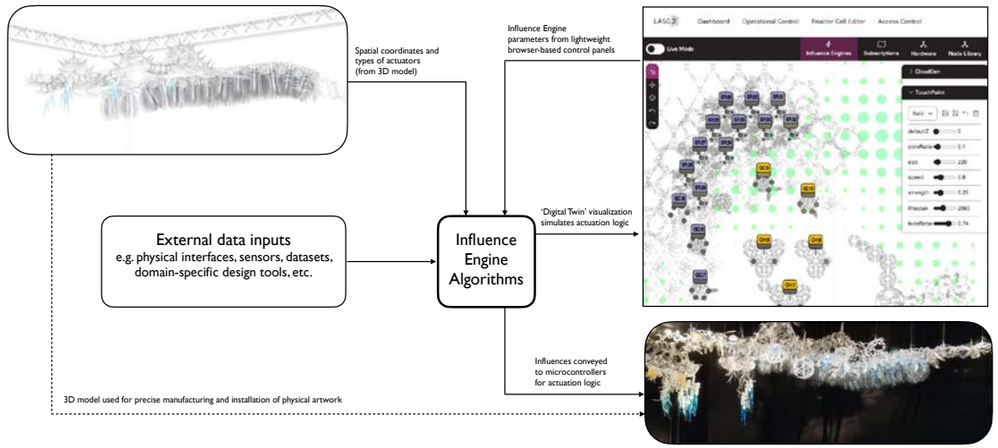
Composers and artists work together to create dynamic environments

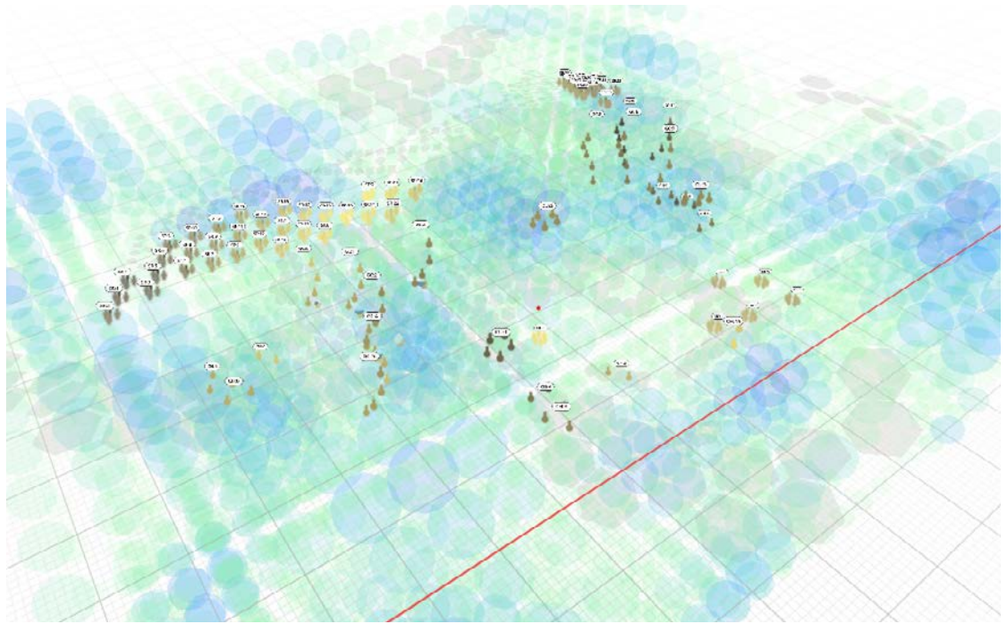
Global Behaviour: Influence Engines

At the heart of *Aether*'s global behaviour system are the concepts of *Influence* and *Attunement*. Layered software algorithms called *Influence Engine*s generate 'influences' on the various spatially distributed parts of the sculpture. For example, a simple *Influence Engine* might create waveforms that move through the sculpture by determining the amplitude of a wave at various physical locations in the space. A different *Influence Engine* might use particle dynamics to calculate the trajectory of virtual objects which influence elements within the sculpture.

Below

A Diagram of the Influence Engine System





Above

A 3D visualization of the "CloudGen" Influence Engine

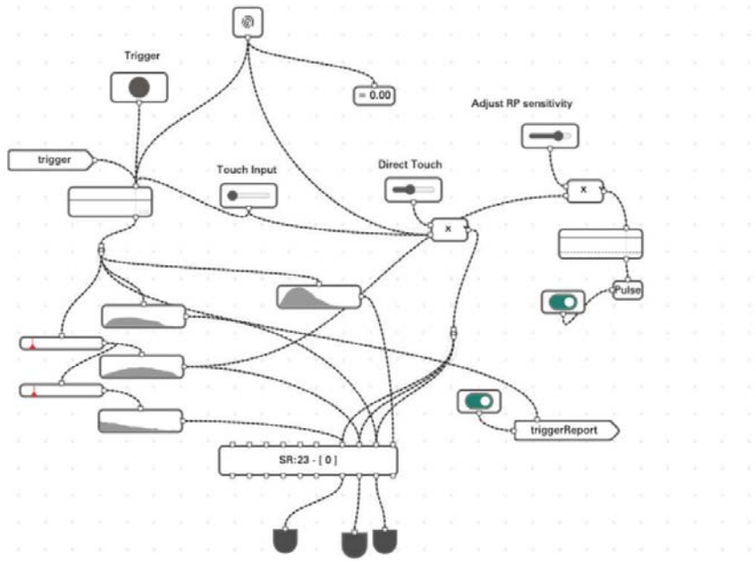
When a computational element within the testbed is *Attuned* to a given influence, it becomes sensitive to this influence. Much as the leaves of a tree move in response to breezes that blow, the algorithms within *Aether*'s *Influence Engine*s do not directly determine what each part of an installation should do. Instead, parametric controls permit the 'attuning' of elements, subscribing them to the 'influence' of *Influence Engine*s.

Elements can respond in many ways to influences. For example, an actuator might glow in the presence of a virtual particle. Alternately, that influence might dampen an already existing glow. These examples are inspired by the behaviours of independent natural lifeforms in a changing ecosystem environment. *Aether* supports control of the local logic of whether and how to respond to certain influences, how strong their impact becomes, and how that can become visible. Like the natural group behaviour of fireflies, crickets and ants, coordinated behaviour might occur when many elements within an installation are similar. The potential also exists for parts of the sculpture to respond independently as influences shift. Each *Influence Engine* runs simultaneously. Parameters for each *Influence Engine* can be adjusted using web-browser-based controls. The *attunements* of sculpture elements and their responses to the influences can also be adjusted via software tools in a browser.

Internal Behaviour: Logic Reactor

Complementing global *Influence Engine*s, *Aether*'s *Logic Reactor* system has been developed by PBSI/LASG in order to provide individual devices containing actuators and sensors with rich internal logic. Devices can process inputs from sensors, *Influence Engine*s, and messages from other devices, adjusting their behaviour and playing back sequences of stored values as time-based electrical signals. This logic is made visible as a *Logic Graphs* and *Reaction Sequence* outputs, configured through a novel visual programming editor in a web browser. Changes to the graph are uploaded wirelessly and in realtime to hardware devices.

Devices can be configured to drive different types of actuators using the *Logic Reactor Editor*. Editing sequenced profiles can change the output behaviour. For example when triggered, a connected LED light may be configured to follow a sine wave profile so that it rises and falls in light intensity. Alternately, if that profile expresses a series of pulses, the light could blink on and off. Logic for triggering different sequences at varying speeds and under varying conditions can be explored using this simple system. Complex behaviour including feedback loops and simple state machines can also be explored.



Below

A web browser interface for designing internal Logic Graphs

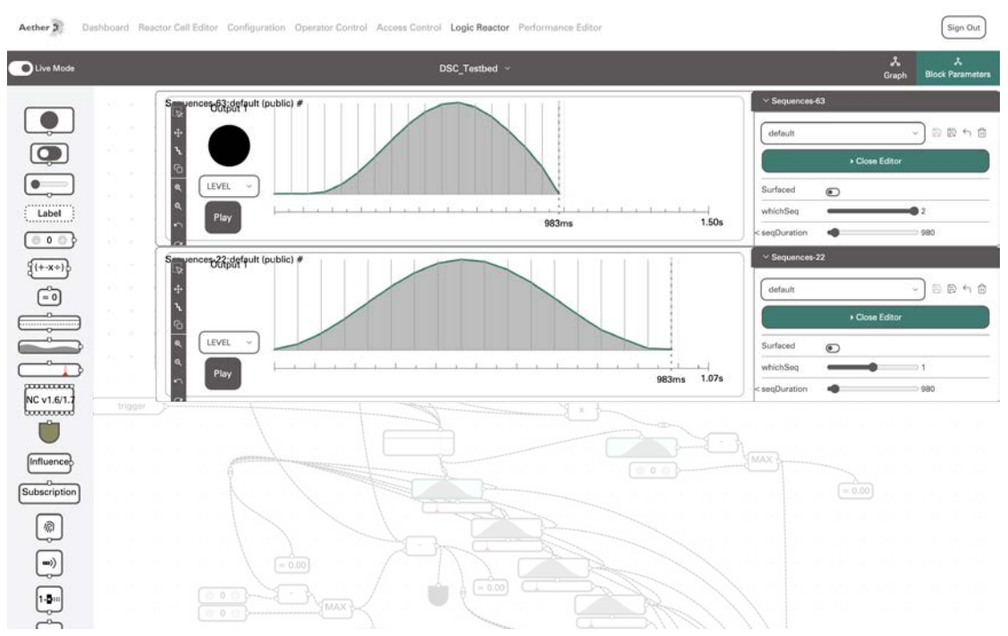
Logic Graphs can receive input as stimulus in several ways:

- From virtual environmental stimuli, controlled by *Influence Engine* algorithms
- From direct or broadcast messages sent to the device over the network
- From attached physical sensors such as sound sensors, infrared proximity sensors, radar-based time-of-flight sensors, and capacitive touch sensors. These can all be used to detect interaction, triggering behaviour changes within specific components of the testbed.

Aether's *Logic Graphs* can be created, saved, and experimented with through the use of its real-time web-based software interface. Complex behaviours can combine internal responses to the dynamic *Influence Engine*s and the external interactions of individuals and other devices with the testbed. *Logic Graphs* within *Aether* offer a combination of intuitive visual programming and realtime embedded firmware logic control. This feature provides a powerful way to experiment with low-level behaviour code within the devices that make up LASG sculpture and kit systems.

Below

An integrated interface for designing sequences for playback



Software Paradigms

Living Architecture Systems Group installations engage educational and research exploration embedded within architectural spaces. *Aether* software has been developed to support research and creation, offering time and space dimensions that are matched to a range of architecture and sculpture installations. Software properties have also been developed with features that can support collaboration, experimentation and education.

Properties for Architectural Scales and Timeframes

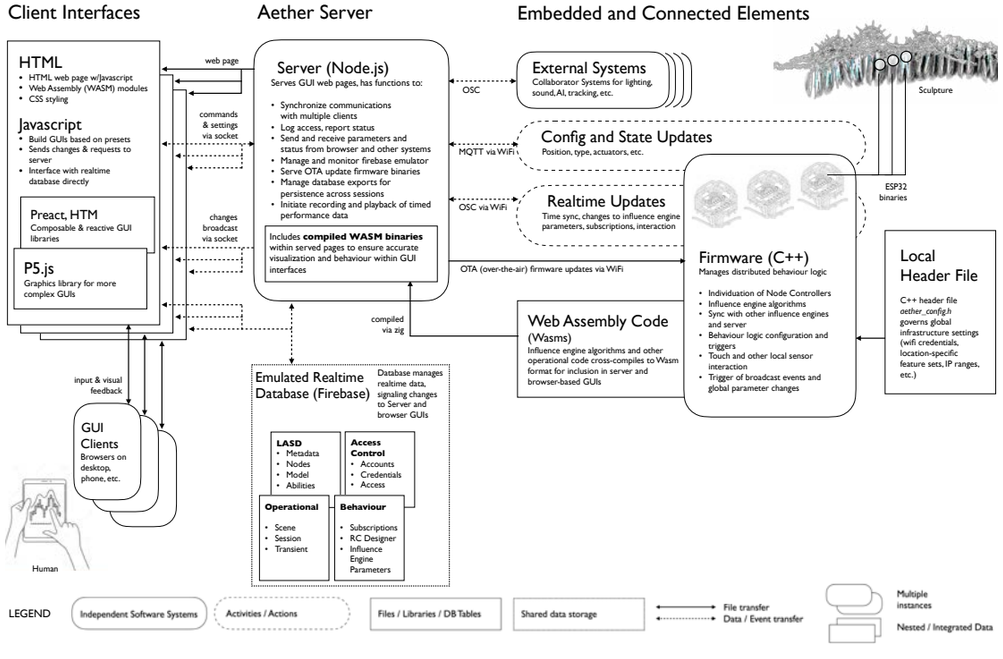
- Persistence & Resilience (shared state and fault-tolerance)
- Spatial Coordination (physical anchoring and awareness)
- Scalability (distributed processing)

Properties for Collaboration, Experimentation and Education

- Extensibility & Flexibility (layered behaviour systems)
- Interoperability (shared protocols and APIs)
- Ecosystem of Tools (3rd Party control)

Below

A diagram of the systems that form the core of the Living Architecture Digital Ecosystem

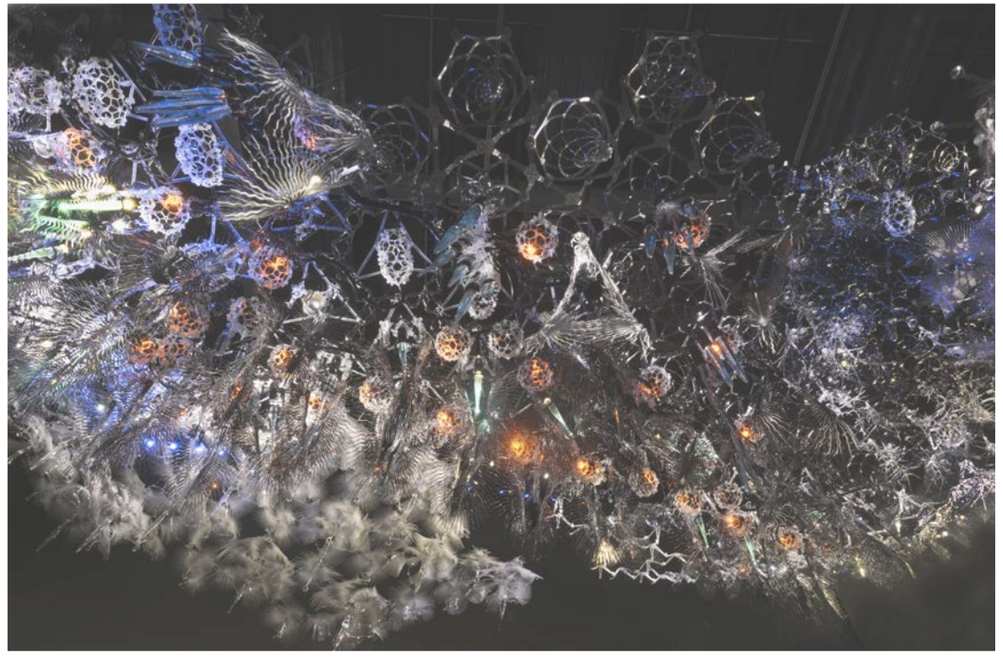


Persistence & Resilience

While many experimental settings for dynamic, interactive installations are not expected to be permanent, long-term consistency and reliability is required for architecture-scale installations. *Aether*'s Spatialized Digital Milieu infrastructure is designed to run persistently within the physical environment, accommodating and affecting active objects within the spaces of installations. In order to achieve persistent, reliable performance, *Aether* has been developed with features that prioritize resilience. *Aether*'s low-level systems monitor independent code modules and data integrity.

Below

Dozens of distributed devices create a resilient canopy

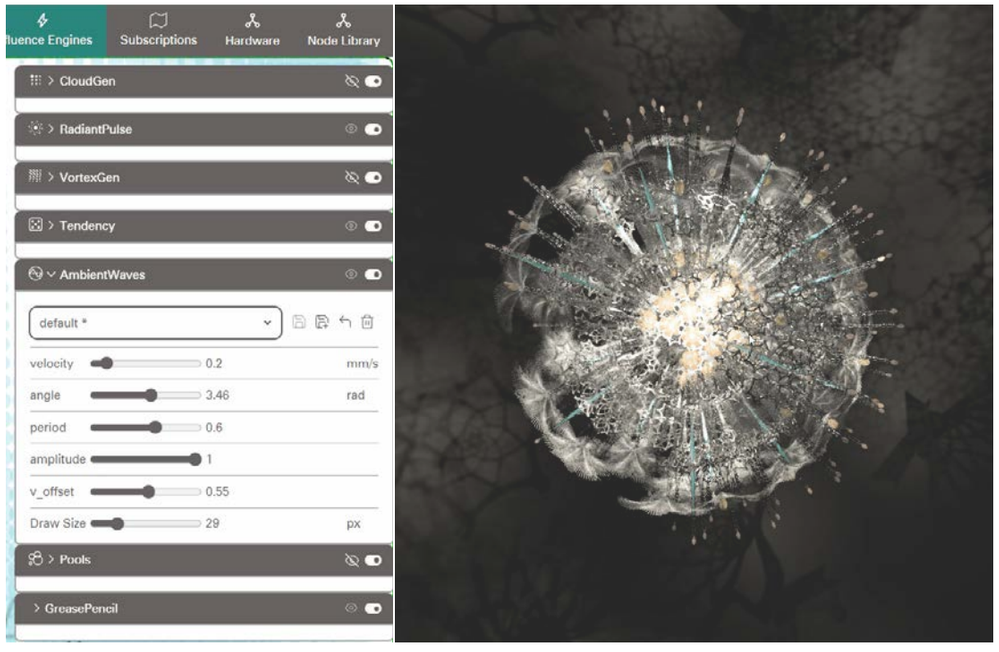


Spatial Coordination

For active elements to be a part of the environment, the underlying data architecture needs to provide an understanding of the dimensional relationships of objects to the space, to one another and to people. Elements within the LASG software systems use their position in physical space to directly affect their behaviour. Like a digital breeze, *Influence Engine* algorithms use this spatial positioning to trigger coordinated behaviours within the space.

Below

Generated behaviours can be visualized in 3D space using the Aether Simulator

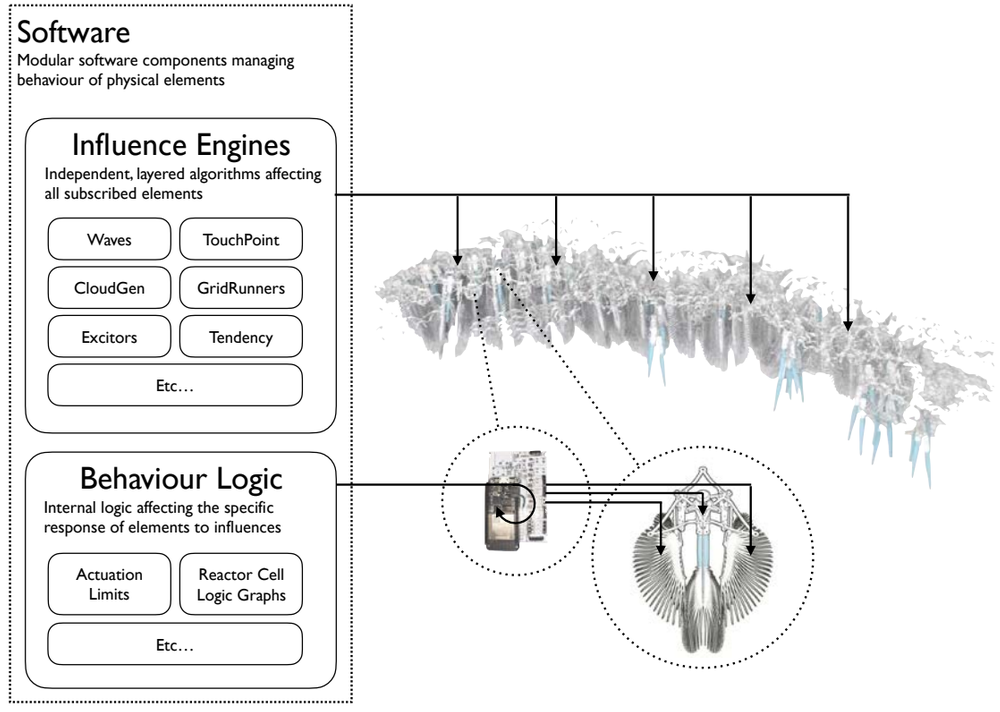


Scalability

Distributed behaviour among many hundreds or thousands of elements at an architectural scale requires strategies for moving beyond centralized processing and the need to choreograph every element of such an extensive system. LASG systems implement a distributed, behaviour system, giving internal behaviour and agency to the many devices within the system, with only minor and low-bandwidth central control.

Below

A web interface for Reactor Cell Logic Editor using visual coding with logic graphs

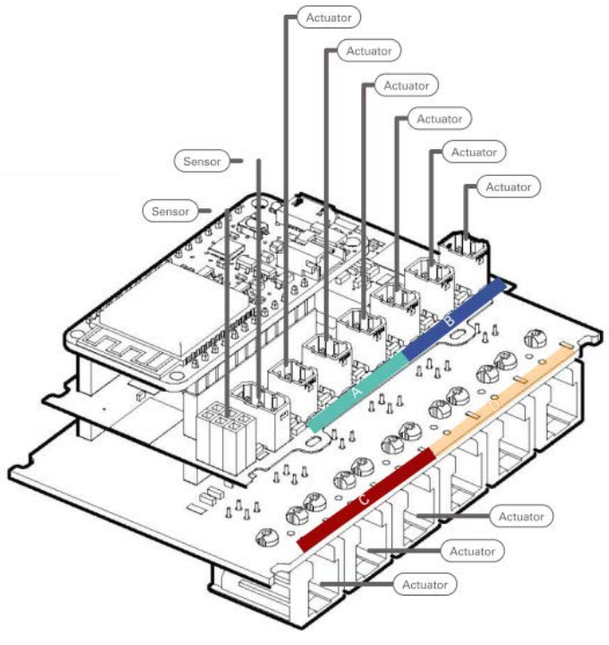


Extensibility & Flexibility

Just as new furnishings can be added to an architectural space, responsive environments need to be able to be extended with new characteristics, influences and behaviours that were not imagined by the original creator. *Aether* is designed to permit new layers of behaviour and complexity to be safely layered within installations, while the integrity of existing systems is preserved.

Below

The logic editor interface allows designers to reconfigure the actuators in a testbed, without the need to recompile firmware

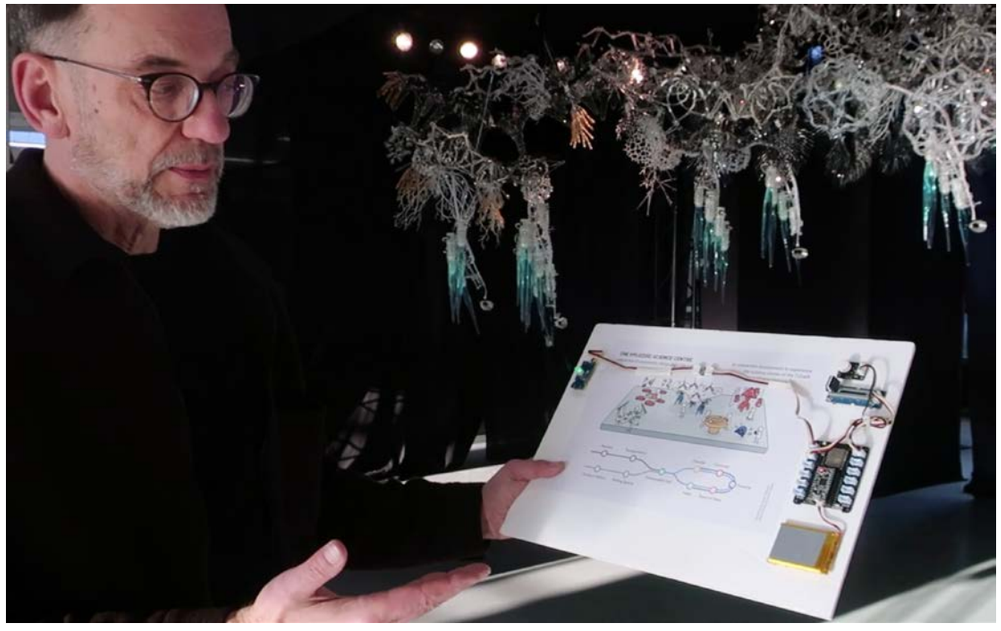


Interoperability

For broad adoption and resilience, systems acting within the space must be compatible with the software infrastructure, regardless of their brand, hardware or diverse technological capabilities. *Aether* software systems anticipate new kinds of devices as elements within testbeds. APIs are positioned for engagement with the existing milieu and devices. This resilience can help guard against the obsolescence that tends to occur within technology operating during long architectural timeframes.

Below

The LASG software system allows for communication with other hardware and devices.



Ecosystem of Tools

Aether has been designed to accommodate the possibility that behaviour within installations may be authored by many different people with diverse creative backgrounds.The web-based interfaces of *Aether* are designed to enable creators with differing levels of technical knowledge to explore the potential of the testbeds. *Aether* can also be configured using familiar software packages and technology tools from domains such as dance, music, theater, robotics or gaming. Bespoke interface devices for specific explorations can also be accommodated.

Top Right

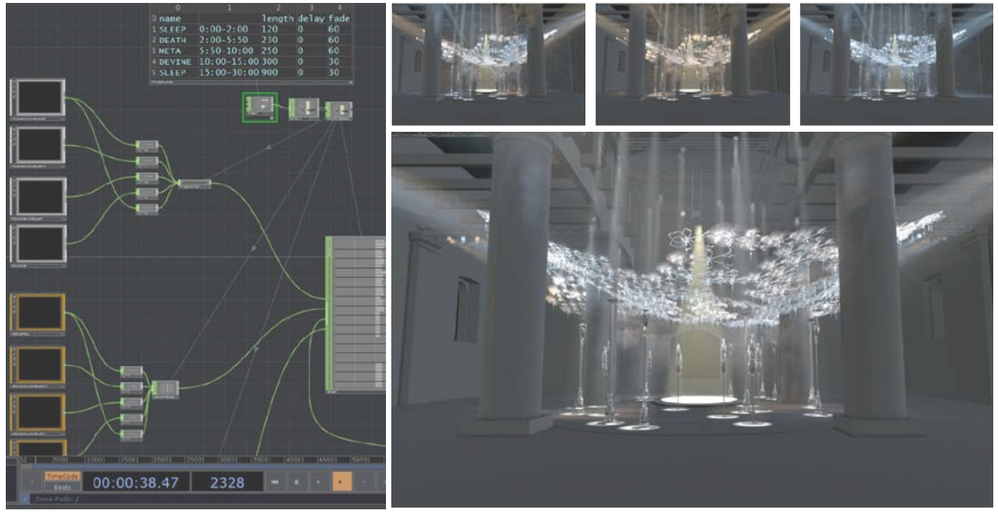
"Living Shadows," a novel shadow-based augmented reality system uses a game engine to interface with the LASG software infrastructure. Read more about Living Shadows: https://livingarchitecturesystems.com/publications/

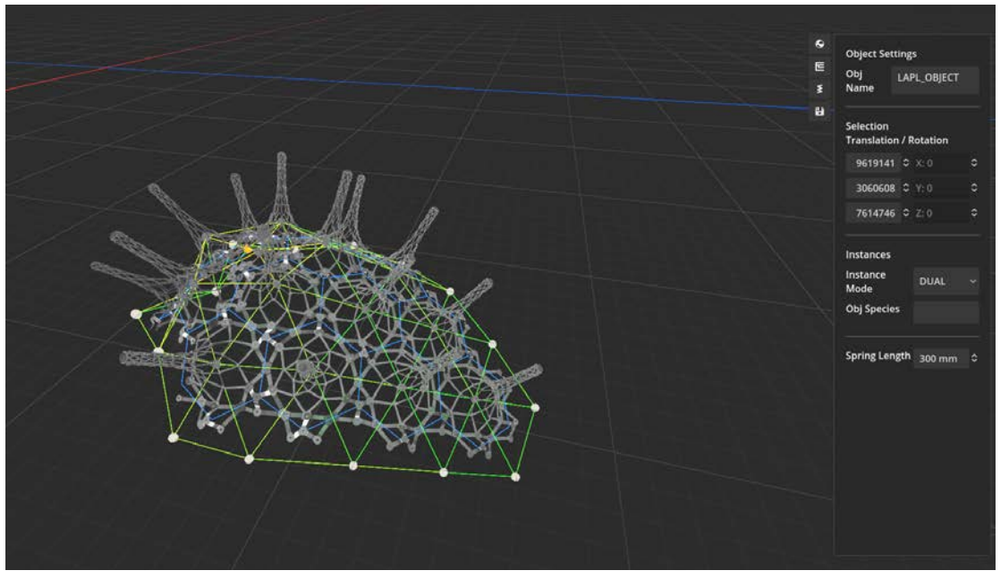
Bottom Right

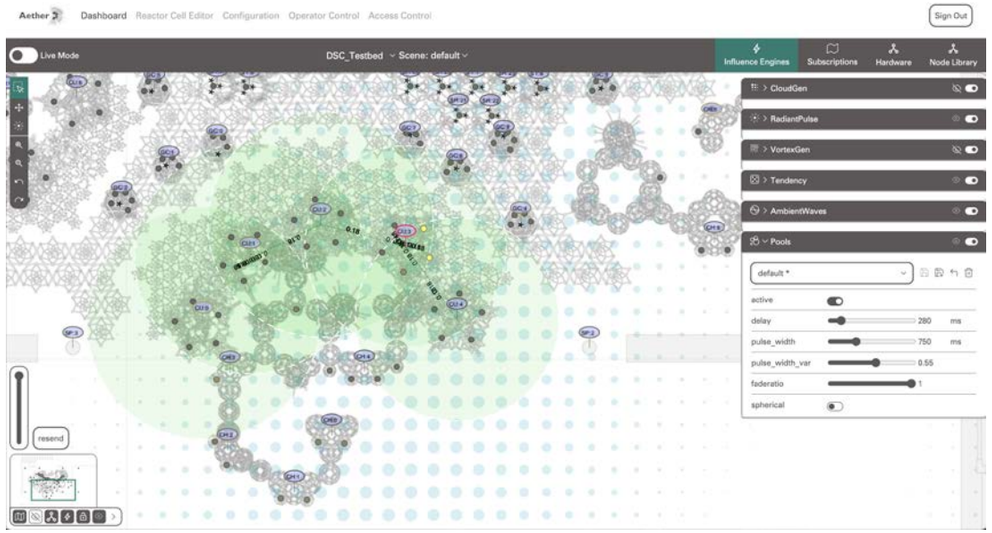
The "Living Architecture Pattern Language" System, a bespoke design tool that interfaces with the LASG software infrastructure

Below

Touch designer & Blender communicating with the LASG software infrastructure to create and pre-visualize dynamic animated lighting







Composing an Ecosystem: Performance Editor

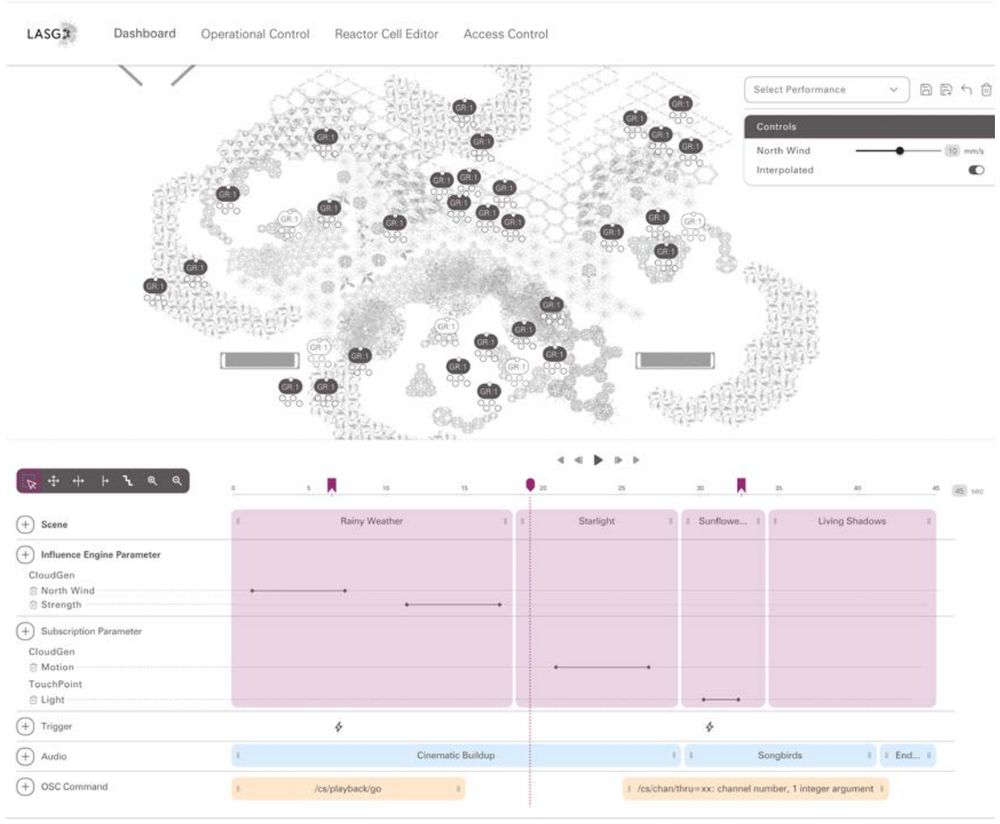
Aether has been designed to accommodate the possibility that behaviour within installations may be authored by many different people with diverse creative backgrounds.The web-based interfaces of *Aether* are designed to enable creators with differing levels of technical knowledge to explore the potential of the testbeds. *Aether* can also be configured using familiar software packages and technology tools from domains such as dance, music, theater, robotics or gaming. Bespoke interface devices for specific explorations can also be accommodated.

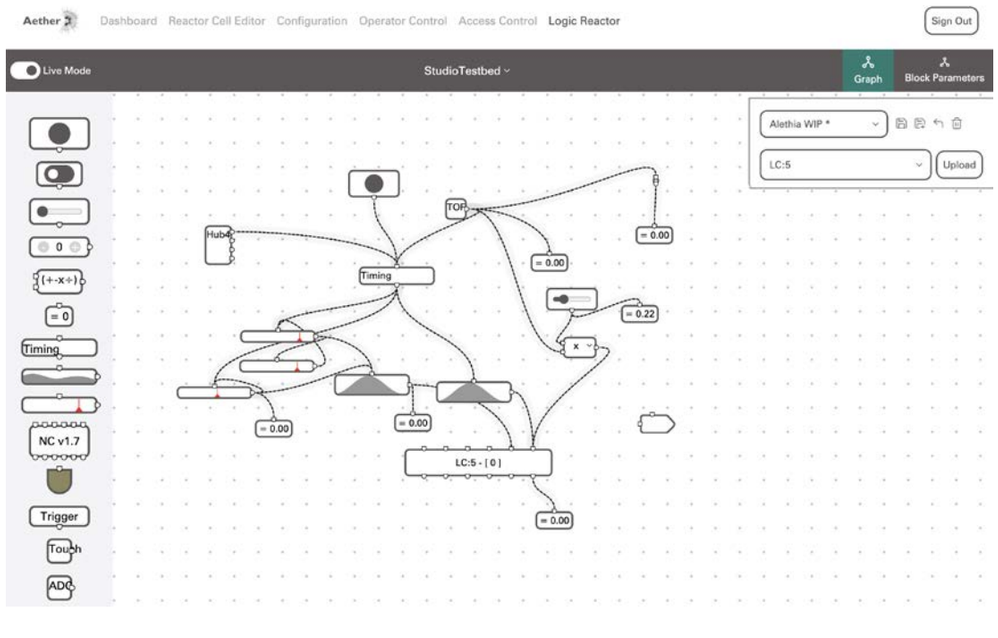
Above

Reactor Cell Chains can propagate across devices, creating complex spatialized chain reactions.

Right

A Mock-up of the Performance Editor





STEAM and Behaviour Design Exploration

Future development plans of *Aether* include *Logic Reactor Editor* features that can be used to explore and design internal behaviour logic for individual responsive entities. Foundational behaviours can be combined in order to produce emergent effects within large ecosystems. A 'block coding' style would be deployed, supporting high-level code exploration in STEAM-based education contexts.

Above

A web interface for Reactor Cell Logic Editor using visual coding with logic graphs

Facing Top

Students explore emergent effects using coding to design the behaviour of Reactor Cells

Facing Bottom

Simple coding paradigms reduce the barrier of entry especially in workshop scenarios

How to Obtain Aether

Proposals for access and development of *Aether* are welcome. If you are interested in using *Aether* software, or in collaborating in further development of this suite of tools, contact info@lasg.ca.