
THE WOODEN IDOL

THE TURING CHAMBER

Research overview, verified results and current investigation

Marc Craig - artist and project originator
Status updated: 3 October 2026

This artist-led investigation began with the changing walls and collaborative practices of Leake Street, London. Observations about making, covering, retaining, completing and beginning again developed into SFVFS - Seed / Form / Void / Form / Seed - and then into twelve requirements called The Wooden Idol. Through sustained human-AI collaboration, selected mathematical readings of those requirements were formalised in Lean 4 and brought together in a computational construction: The Turing Chamber.

The central established milestone is a construction in which the twelve specified formal predicates coexist. The current investigation asks a stronger question: which computational abilities, if any, are required or generated by those predicates, rather than introduced separately by the construction?

This page provides an entry point for mathematicians, computer scientists, researchers in mathematical practice, and artists interested in the relationship between creative observation and formal verification.

CURRENT STATUS AT A GLANCE

- Public foundation: The Turing Chamber, Version 1.0.0, archived on Zenodo. The release records successful certification on 30 September 2026.
- Current work: Phase 5, investigating the reverse direction from constraints to computational primitives and operational behaviour.
- Latest bounded verification: 63 Lean source modules compiled successfully in a fresh recheck on 2 October 2026, without source edits or reuse of previously compiled project objects.
- Main boundary: satisfying all twelve does not force several specified forms of instruction execution, active transfer, fresh recording or use of retained history. The exact scope of each result matters.
- Open objective: a constructive bridge identifying which behaviour follows from which constraints, with every additional assumption stated.
- Review status: a separate AI-assisted audit and a fresh Lean recheck have been completed. These are not being presented as external academic peer review or confirmation of mathematical novelty.

PUBLIC CANONICAL ARCHIVE

<https://doi.org/10.5281/zenodo.23053338>

Archive record and source download:

<https://zenodo.org/records/23053338>

WHAT THE PUBLIC RELEASE ESTABLISHES

The public release is titled “THE TURING CHAMBER - Canonical All-Twelve Formal Machine and Lean 4 Certificate”. It contains the source archive `turing-chamber-canonical-ef99ece.zip`.

The principal certificate is:

```
WoodenIdolTuringChamberFinal.wooden_idol_turing_chamber_all_twelve
```

The archived result establishes that the specified B1-B12 predicates hold simultaneously on one final coupled system, together with the productive proof-system condition. The construction draws on a helix-machine and finite-control development. Its computational results and its B-predicate certificates are connected by explicit formal constructions; their precise relationship is part of the continuing investigation.

B5 is supplied through an explicitly attached arithmetic proof sector using $I\Sigma_1$ derivability and Foundation’s formalisation of Gödel’s second incompleteness theorem. The release does not claim that the Chamber’s ordinary transition dynamics themselves generate Gödel II.

The coexistence result is a substantive answer to the formal existence challenge. It does not, on its own, prove that the twelve conditions generate the computational core or determine its instruction semantics.

Release identity:

Version:	1.0.0
DOI:	10.5281/zenodo.23053338
Source commit:	ef99ece128ebb492cd2b842949d51b292db9f612
Lean:	4.35.0-rc2
Mathlib:	065356127b1dc0016f66b7283ce0ce2c4055aa55
Foundation:	e72cfe981aa65166f37fa4e2584f4806bc48d72f

These identifiers refer to the public release. They do not identify the later Phase 5 development as a new public release.

THE FOUNDATIONAL PROGRAMME: PHASES 1-4

The following describes the purpose and attained outcome of the foundational stages. The final coexistence certificate is not being treated as evidence that every stronger explanatory test proposed along the way has passed.

Phase 1 - Fix the mathematical readings

The first task was to settle which precise predicates were being investigated. Artistic language, experiential principles and mathematical definitions had to be distinguished so that a proof could be checked against a fixed target. The canonical source supplies those exact formal targets. Earlier descriptions remain part of the project's history, but do not silently replace the frozen definitions.

Phase 2 - Put the requirements on a shared system

The next task was to express the requirements through a common state-and-transition framework, rather than twelve unrelated examples. The attained construction provides a coupled system on which the predicates can be assessed together. A stronger question - whether a particular geometric or structural component is indispensable to computation - needs its own dependence proof and is not settled merely by sharing one system.

Phase 3 - Build explicit connections and test their limits

The programme introduced named links between the machine construction and individual formal requirements. Positive constructions were accompanied by controls and boundary tests where available. Each connection has its own scope: for example, the existence of a coherent transport operation is different from proving that the working transition actually executes it. The later audit preserves this distinction rather than treating every structural connection as an operational bridge.

Phase 4 - Establish coexistence on one final object

The public canonical release is the milestone of this stage: the twelve formal predicates are certified together on the final coupled construction. The separate proof-theoretic component is made explicit. This establishes compatibility and provides a fixed object that others can inspect. The reverse-generative question remains a further obligation.

READING B1-B12: THREE DISTINCT LAYERS

The project uses three related vocabularies:

1. The original artistic challenge: prose requirements developed from the creative investigation.
2. The experiential principles: Selection, Forgetting, Arrival, Answer, Plan, Done, Detach, Awareness, Release, Unknown, Acceptance and Infinity/Finite.
3. The formal predicates: exact Lean definitions in an identified source version.

A Lean theorem establishes its formal statement under its declared assumptions. A further interpretation argument is needed to connect that statement to an artistic or experiential

description. Shared B-numbers do not make those descriptions mathematically interchangeable.

For example, the original website wording names B6 “Transformation Without Displacement”; the audited formal B6 concerns earned completion and covered episode requirements; the experiential name is “Done”. These are different readings within the development, not three identical theorem statements. The relationship between them is open to examination.

The following is a concise guide to the frozen System-level predicates used in the Phase 5 audit. It is not a replacement for their definitions and supporting types:

B1 - No table in the specified interpreter captures all reachability. This is not, by itself, a claim about every possible algorithm or decoder.

B2 - An injective orbit has a nonconstant declared observation and arbitrarily late returns of that observation.

B3 - An inhabited, invariant, backward-stable source is recognised and usable but unnamed in the selected internal language.

B4 - The declared relation is characterised by the meeting of forward orbits.

B5 - The proof predicate is consistent and does not prove the designated consistency statement; the proof-layer assumptions and their instantiation must be specified.

B6 - An earned completion exists, and doing-to-done transitions have covered episode requirements. Fresh recording and later use are separate questions.

B7 - The state space is inhabited, the proof predicate is consistent, and a declared output-reference operation is fixed-point-free. This operation is not automatically the working step.

B8 - Coherent cross-site transport is available with retained origin and a non-creation condition on coherence. Availability does not assert execution by the working step.

B9 - Each working step changes state, preserves body identity and preserves the declared relation.

B10 - The working step is injective, with additional conditions on denotation, revealed observations and continuing possibilities.

B11 - A nontrivial invariant exists that is unnamed in the selected language. Language-relative unnamedness is not automatically general noncomputability.

B12 - Completed states have linked continuation seeds with required retention; continuation agrees with realised linked reachability. Availability is distinct from activation.

PHASE 5: FROM SATISFACTION TO GENERATION

Phase 5 asks what the conditions contribute to the machine's behaviour.

The compatibility direction is:

Construct a particular system, then prove that it satisfies the twelve predicates.

The stronger generative direction is:

Start from a specified subset of the predicates, within a declared class of systems, and derive or construct a computational primitive with the required operational behaviour.

Candidate behaviours include a successor process, instruction-sensitive execution, active transfer between sites, and the later use of retained information. Each requires its own theorem and a control showing what happens when a relevant condition or connection is removed. Added assumptions must be named, rather than absorbed into the meaning of a B-predicate.

Not every desired implication survives. Establishing a counterexample is a completed result about that implication; it helps identify what a constructive approach would still need.

WHAT THE PHASE 5 INVESTIGATION HAS ESTABLISHED

Constraints can hold while a specified instruction is not executed

An audited family satisfies the frozen twelve for arbitrary supplied engine updates, with proof-system productivity established separately. A native dispatcher passes a specified one-step register-increment interface; an idle update fails it. Consequently the twelve do not universally force that interface. The result is about the named encoding, transition and readout; it does not exclude arbitrary external decoding or every alternative computational interpretation.

Available transport need not become an actual exchange

B8 supplies a genuine structural transport property. In an All12 control family, however, every working step preserves the encounter site, so the specified cross-site transfer is not realised by repeated working steps. A further execution condition can connect transport to dynamics, but that condition is additional structure, not something already derived from B8.

Retention need not be fresh recording

The designated System.history field can remain the constant list [0,1] in an All12 control, with the encounter identifier already present before completion. Thus the twelve do not universally require a new canonical history entry or any change to that history field.

A growing log need not influence later register output

A separate engine-snapshot log grows along the dynamic branch of the audited family. Nevertheless, changing the past log while keeping the specified present view fixed cannot change the future register trace. This holds for every supplied engine update in that family. Log length still affects the declared canonical observation; the result does not say that the log has no effect on anything.

Some retained structure has a demonstrable role

Erasing the old log preserves the present-engine/register computation, but the resulting native HALT transition can merge distinct states and therefore lose injectivity. B10 requires injectivity. This explains a role for retained structure in the representation, while leaving the instruction dispatcher itself supplied separately.

There are positive consequences and a precise growth obstruction

B10 preserves distinctions between complete states at equal finite horizons. B2 gives arbitrarily late returns of a declared observation without a return to the identical complete state. Frozen B3 rules out a natural-number rank that strictly increases at every working step for every state. This last result does not prohibit growth on selected encounters or branches, or measures in other orders.

External input can operate a supplied executor

A separately constructed ignition/dialogue model proves that, after an accepted input supplies a program and starting registers, its subsequent native computation agrees with the specified register-machine run. This is an implementation result. The dispatcher, acceptance rules, scheduling and communication mechanism are supplied; that dialogue wrapper is not itself globally certified All12. Two roles within one global state are not a proof of independently scheduled distributed interaction.

THE CURRENT OPEN TARGET: WHEN THE PAST CHANGES THE NEXT ACTION

The next defined target is reachable delayed feedback under a common input. No completed theorem for this target is claimed here.

Starting from the same initial system, two accepted input histories should lead to states with the same declared present view but different retained records. Apply one identical subsequent input sequence to both states. The target is to prove different later register outputs, using a fixed decoder that reads the present view rather than inspecting the hidden record itself.

The receiving and record-reading rules must be concrete. A control with the record-reading connection removed should fail to produce the distinction. The input-labelled transition must also be connected honestly to the frozen unary working step: an idle input must agree with that step, and preservation under non-idle inputs must be stated separately.

Such a construction would demonstrate operational use of retained information. If its reading rules are supplied, it remains an implementation theorem. A stronger generative claim would require showing which B-conditions force or make those rules indispensable within the chosen class of systems.

The Leake Street analogy is a wall whose earlier layers influence the next mark. A retained layer can affect the next action through the surface itself; deliberate recollection is not the only possible mechanism. The analogy motivates a question. It is not evidence that the current formal predicates already answer it.

VERIFICATION, EVIDENCE AND LIMITS

The 2 October 2026 fresh recheck compiled all 63 source modules in the scoped audit: 61 unchanged inherited files and two residue-audit additions. It used Lean 4.35.0-rc2 and the pinned Mathlib revision given above. The 39 canonical abstract-closure sources were compared directly with the supplied canonical snapshot and were byte-identical. Sources, commands, hashes, exit codes and logs were preserved.

The two residue modules contain 24 printed theorem-axiom audits. These use only subsets of propext, Classical.choice and Quot.sound; two require no axioms. No source repairs were needed for this recheck.

Across the inherited material, 20 printed audits contain 18 distinct additional native_decide-generated axiom names, including an inherited abstract All12 aggregate. Those existing native-evaluation dependencies are disclosed separately. Successful compilation is not being described as standard-foundational-axioms-only verification of every inherited theorem.

The heavy Foundation/ $\mathbb{I}\Sigma_1$ arithmetic adapter was not rebuilt in this scoped recheck. Its certification belongs to the public release's recorded verification, not to the later 63-module result. The new audit does not independently recertify every file in the larger development.

The current results do not establish completion of the original Phase 5 generative ambition, autonomous collaboration between independently scheduled Idols, or a formal theory of lived experience. No conclusion about the Riemann hypothesis, P versus NP, consciousness or general physical laws is claimed. Mathematical novelty remains a question for literature comparison and specialist assessment.

ACCESS AND RESEARCH ENQUIRIES

The pre-Phase-5 canonical construction and its source are publicly available through Zenodo. Current Phase 5 development is maintained in a private GitHub repository. This page reports its scope and status without publishing the development repository.

Researchers can contact Marc Craig to discuss access to a bounded review package containing the relevant source closure, frozen definitions, theorem statements, dependency information, axiom audits and successful build logs. A review package can be shared separately without making the repository public. The public summary is an introduction to that evidence, not a substitute for inspecting it.

Feedback is particularly welcome on:

- Whether the chosen formal predicates adequately capture the intended mathematical readings.
- Existing results or frameworks that already explain the proved consequences and obstructions.
- A precise model class and minimal additional assumptions for a constructive primitive bridge.
- The distinction between structural retention, operational feedback and interaction.
- The artistic and methodological significance of the human-AI formalisation process.

The project originated with Marc Craig's artistic practice and observations. AI systems contributed interpretation, proposals, criticism, coding and proof development. Lean checks formal statements against definitions and assumptions; it does not certify the artistic interpretation, empirical truth, originality or significance of the project.

For research enquiries, please use the contact form below, naming the aspect of the project you would like to discuss.

Canonical source and certificate: <https://doi.org/10.5281/zenodo.2305338>

Project: <https://itvoids.com>

Artist: <https://marccraig.co.uk>