# Visual Lean: An accessible interface for Lean 4 proofwriting


Autumn Mapes

rm2264@cam.ac.uk
Department of Applied Mathematics and Theoretical Physics
The University of Cambridge, UK



**Abstract**

Interactive theorem provers (ITPs) are programming languages built to represent and prove mathematical theorems. ITPs are normally presented in a technical manner, requiring familiarity with the conventions of both formal logic and computer programming. In order to foster the adoption of ITPs outside of pure mathematics, we propose Visual Lean, an interactive drag-and-drop editor for writing Lean proofs compatible with any existing Lean game. Though the application is geared towards getting users familiar with the basics of formal proof writing, we hope to discover design principles that could benefit even advanced Lean workflows.


## 1 Introduction

Interactive theorem provers (ITPs) are becoming increasingly popular within pure mathematics, leading to a wealth of results in centralized repositories like Lean's `mathlib` [2]. Recent efforts like SciLean have begun to explore the formalization of results in applied mathematics with the aim of building practical problem-solving tools [3]. As ITPs find an audience beyond mathematical logic, onboarding and usability become a significant roadblock. Already, the Lean community has benefited from its close integration with Visual Studio Code, providing detailed proof state information in real time. Powerful tactics like `simp` and `linarith` reduce the tedium of algebraic manipulation. As an experiment towards an even higher level of abstraction, we have built a simplified interface for writing Lean proofs called Visual Lean.

## 2 The Visual Lean Prototype

Visual Lean has an entirely drag-and-drop interface: no keyboard is necessary. Visual Lean is built upon the public Lean4Game repository and thus is compatible with any existing Lean game [1]. As a demonstration, we have provided a visual counterpart of a Lean game called the Natural Numbers Game: users are guided through proofs of simple results in Peano arithmetic. This demonstration is currently playable online in any web browser or device[1]. All implementation code is hosted publicly on Github[2][3][4].

The interface is split into three major modes:

- **Combining Mode** 1
  Users can manipulate hypotheses to resolve the 'goal' statement fixed on the right side. A few examples of basic Lean tactics that have a visual counterpart in Combining Mode:
  - `rfl`: Clicking on any goal of the form $x = x$ resolves the goal through reflection.

[1] https://leangame.autumnofautumn.com/
[2] https://github.com/ryyanmapes/lean4game
[3] https://github.com/ryyanmapes/NNG4
[4] https://github.com/ryyanmapes/VisualTest

  - `exact`: If you have a hypothesis of the form $P$ and a goal of the form $P$, dragging the hypothesis onto the goal solves the proof with `exact`.
  - `apply`: Dragging an implication hypothesis $P \to Q$ onto a goal $Q$, `apply` is used to rewrite the goal as $P$. Likewise, dragging a hypothesis $P$ onto a hypothesis $P \to Q$ creates a hypothesis $Q$.
  - `constructor`: Clicking a goal of the form $P \land Q$ splits the proof state into two proof branches: one where the goal is $P$, and one where the goal is $Q$. Proof branches are displayed visually and can be freely navigated between.

- **Transformation Mode**
  By double-clicking expressions of equality like $x = y$, users can make known substitutions via generated `rw` expressions. This makes rewriting specific parts of large expressions much simpler in Visual Lean compared to writing `conv`-style expressions manually.

- **Construction Mode**
  Clicking goals like $\exists a : A,\ P(a)$ and hypotheses like $\forall a : A,\ P(a)$, the user is prompted to construct an arbitrary expression for $a$.

This prototype was programmed with the help of assistance from Claude Code 4.6 and Codex 5.4. The design of the application is entirely our own.

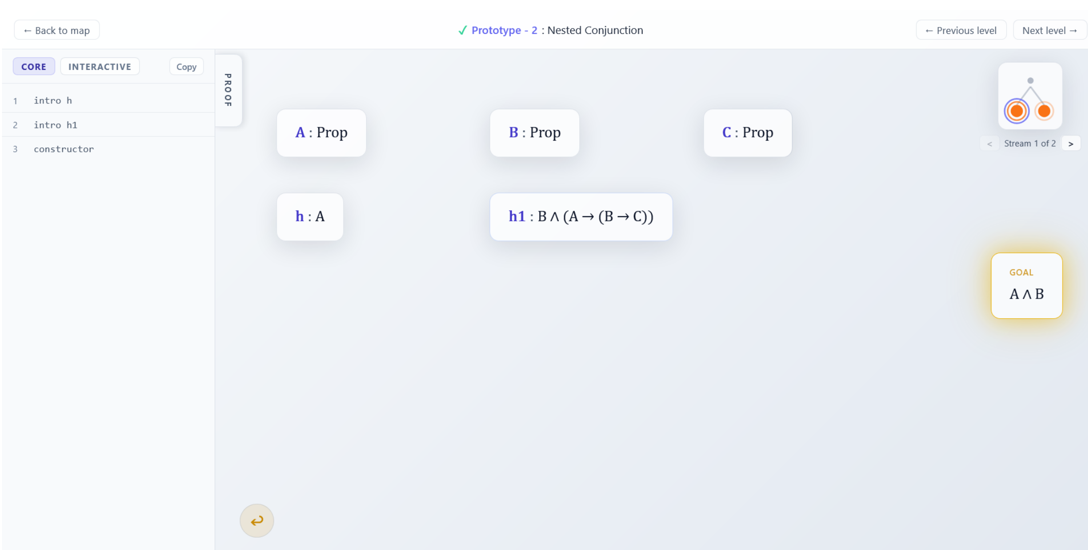


Figure 1: Midway through a proof of $A \to B \land (A \to B \to C) \to (A \land B \land C)$ in the Combining Mode of Visual Lean.

# 3 Acknowledgments

This work was conducted under the advisement of Timothy Gowers, Matthew Colbrook, and Miles Cranmer under the Faculty of Mathematics at the University of Cambridge.

# References

[1] Alexander Bentkamp, Jon Eugster, and Marcus Zibrowius. Lean game server. https://adam.math.hhu.de/.

[2] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, pages 625–635, Cham, 2021. Springer International Publishing.

[3] Tomáš Skřivan. Scientific computing in Lean. https://lecopivo.github.io/scientific-computing-lean/, 2026.