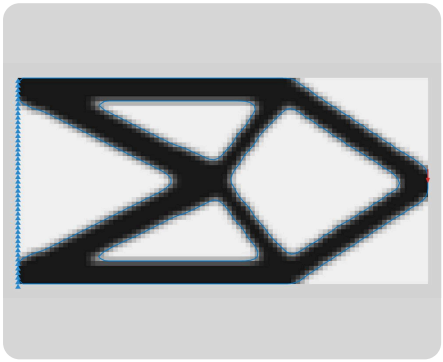
GRASSHOPPER PLUGIN · RHINO 8 FOR WINDOWS

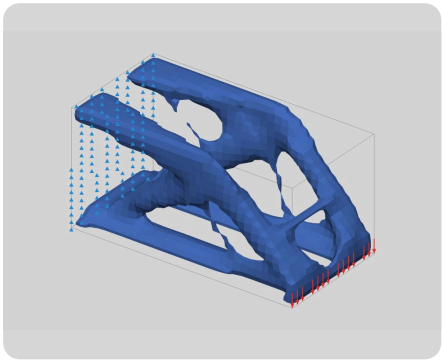
Moyu 1.0.0

Interactive topology optimisation, plus four viewport games disguised as architecture study tools

User guide · a research and teaching project on building a plugin quickly with an AI coding agent



2D pixel optimisation · 2D



3D voxel optimisation · 3D



VoxStack · Massing



Serpentine · Landscape



Rebound · Facade



Swift · Section

Contents

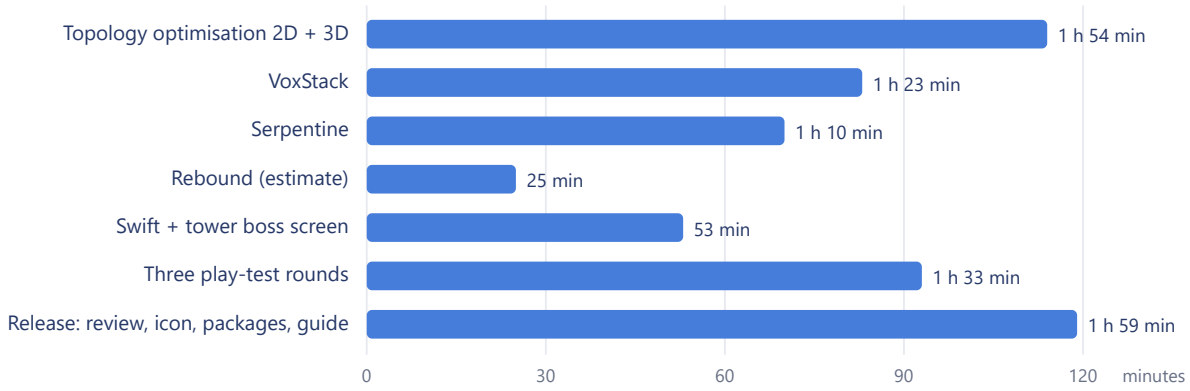
About this project 2
Install and panels 3
Example files 4
2D topology optimisation 5
3D topology optimisation 6
Playing the games 7
VoxStack 8
Serpentine 9
Rebound 10
Swift 11
Troubleshooting and uninstall 12

A research and teaching project. Almost all of the code, tests and documentation were written by Claude Code (Anthropic's coding agent); the author asked for each part, reviewed it and play-tested it. Optimisation results are for study and concept design, not a substitute for engineering analysis. The plugin never goes online and only writes high scores to `%APPDATA%\Moyu\`. MIT licence; source and development logs: https://github.com/ChenmingHe0126/moyu

Building a Rhino plugin quickly with an AI coding agent

Moyu is a research and teaching project: how quickly and how reliably can a coding agent build a complete Grasshopper plugin, with numerical solvers, custom viewport display, keyboard and mouse interaction, and tests? Almost all of the code, tests and documentation were written by **Claude Code**; the author asked for each part, reviewed it and play-tested it in Rhino; the icons were drawn with OpenAI Codex.

Roughly how long each part took



Claude Code's working time, summed turn by turn from the session timestamps. The author's reviewing and play-testing and about 34 minutes of Rhino crashes are not counted. About 9 h 17 min in total, over four days. Details: docs/dev-time-stats.md in the repository.

Every feature followed the same loop

1 Plan
Prototype screenshots in Rhino and a short illustrated plan for the author to approve.

2 Rules and tests
Solvers and game rules live in a core library that needs no Rhino, with unit tests (351 checks now).

3 Rhino side
Grasshopper components, display conduit, keyboard and mouse hooks, cameras.

4 Play-test
Tested in real Rhino with real keys and mouse; screenshots checked and fixed.

5 Record
Docs, time log, commit. Before release, a multi-agent code review.

What each part studies

Topology optimisation (numerics) · Can an agent get engineering numerics right, with tests: finite elements, SIMP with optimality criteria, a density filter, a banded Cholesky solver in 2D and multigrid conjugate gradients in 3D; and how a Timer makes the optimisation iterate live in Grasshopper while you edit it.

Four small games (interaction and the viewport) · How far the Rhino viewport goes from one Grasshopper component: drawing a whole scene and a HUD with a display conduit at a steady frame rate; taking the keyboard and mouse safely and handing them back to dialogs, Grasshopper and Rhino commands; driving the camera; redrawing the same data as a different drawing in one frame (the boss key).

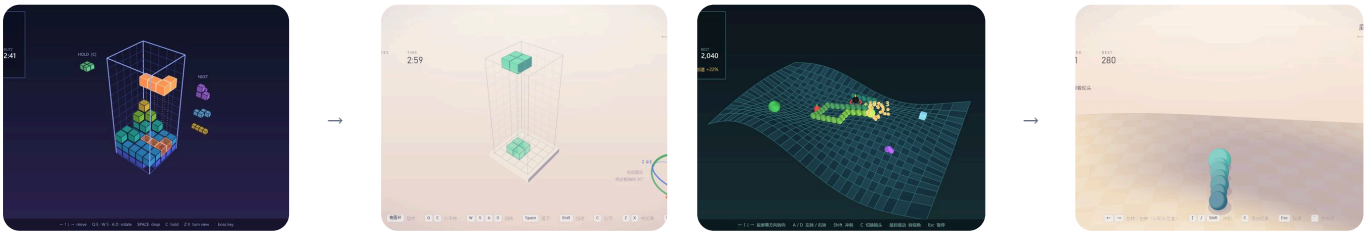
VoxStack · moving and turning in 3D while the view turns; a draggable gumball.

Serpentine · any surface as a playing field (UV grid, wrap, trims, slope); a first-person camera.

Rebound · real-time collision and mouse control on a plane; parametric brick bonds as levels.

Swift · one-key timing and generated levels, checked by an autopilot.

Prototype and result



Left: prototype screenshots that Claude Code built in Rhino for the author to approve. Right: after play-testing, the author asked for one minimal style, and both games were redrawn together (about 19 minutes).

Install

Needs Rhino 8 for Windows, in either runtime (.NET Core, the default, or .NET Framework). Not tested on Rhino for Mac.

A · Package Manager (recommended)

1. In Rhino run `_PackageManager`
2. Search for **moyu** → Install
3. Restart Rhino

B · The .yak file

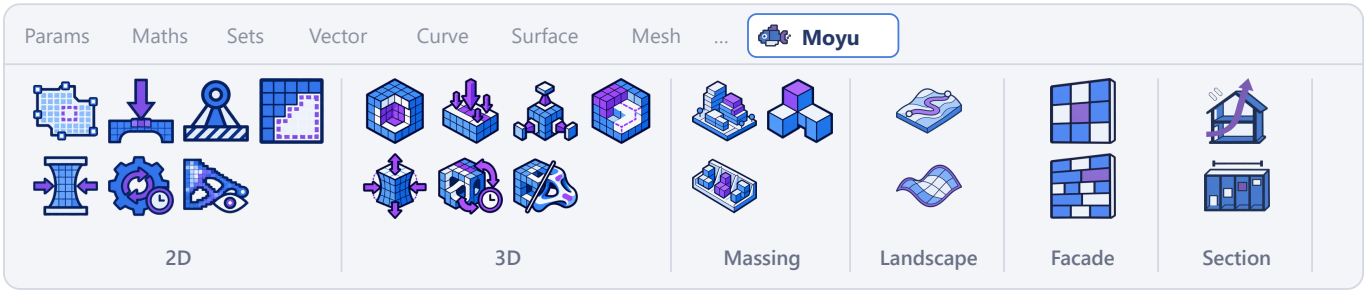
1. Drag the `moyu-1.0.0-rh8_0-win.yak` from food4rhino into a Rhino window
2. Restart Rhino

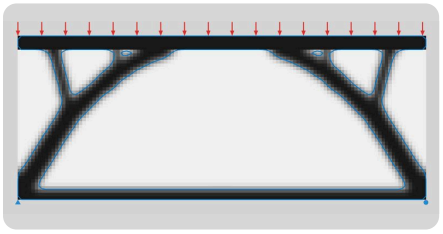
C · By hand

1. Copy **one** `Moyu.gha` to `%APPDATA%\Grasshopper\Libraries\Moyu\`
2. net7.0 for .NET Core, net48 for .NET Framework (`_SetDotNetRuntime` tells which)
3. If Windows blocks it: right-click › Properties › Unblock; restart Rhino

Six panels on the Moyu tab

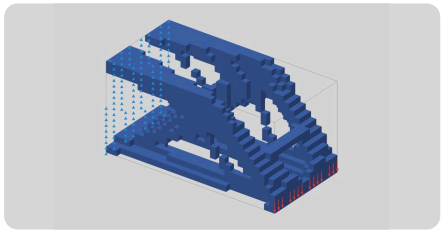
After the restart Grasshopper shows a Moyu tab. The four game panels are named and drawn like architecture study tools on purpose.





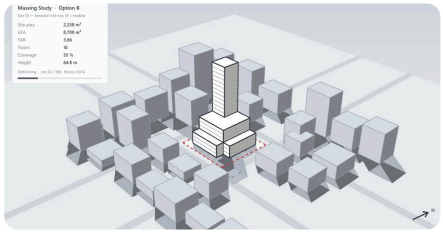
2D

Pixel topology optimisation in a plane: boundary, loads, supports, areas, material → solver → display (page 5).



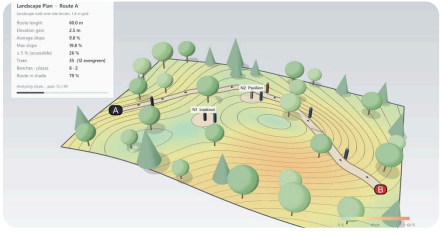
3D

Voxel topology optimisation of a closed solid, with conditions on points, curves or faces (page 6).



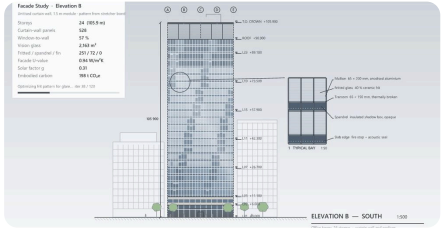
Massing

VoxStack: 3D falling blocks. Shown: its boss screen, a massing study (page 8).



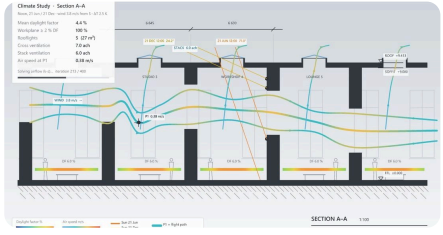
Landscape

Serpentine: a snake on any surface. Shown: its boss screen, a landscape plan (page 9).



Facade

Rebound: brick breaker on real bonds. Shown: its boss screen, a curtain-wall elevation (page 10).



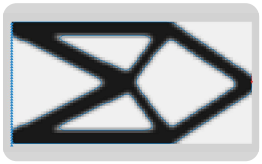
Section

Swift: one-key flight through a section. Shown: its boss screen, a daylight and ventilation section (page 11).

Start from an example

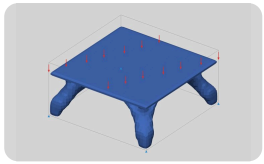
The examples folder of the download holds six ready-made .gh files. Open one in Grasshopper (File › Open, or drag it onto the canvas); the lines at the top of the canvas say what to do.

Where they are: after a Package Manager install, in `%APPDATA%\McNeel\Rhinoceros\packages\8.0\moyu\1.0.0\examples\` ; in the food4rhino archive, in `examples\` .



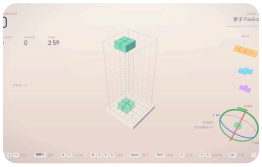
Topo2D.gh

Three 2D set-ups: a cantilever, a bridge elevation and an L-bracket with bolt holes. Each has its own Run toggle and Reset button; the Timer is wired.



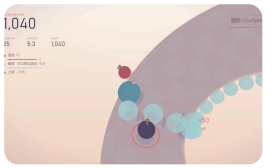
Topo3D.gh

Three 3D set-ups: a cantilever, a canopy on four corners and an L-bracket with holes.



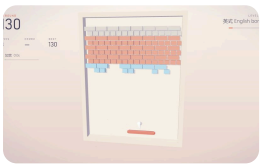
VoxStack.gh

A: just press ▶ Run. B: custom pieces from sketched boxes and a text pattern.



Serpentine.gh

A: the built-in terrain. B: other sites from Site Surface (torus, tube, saddle roof, sphere). C: your own surface.



Rebound.gh

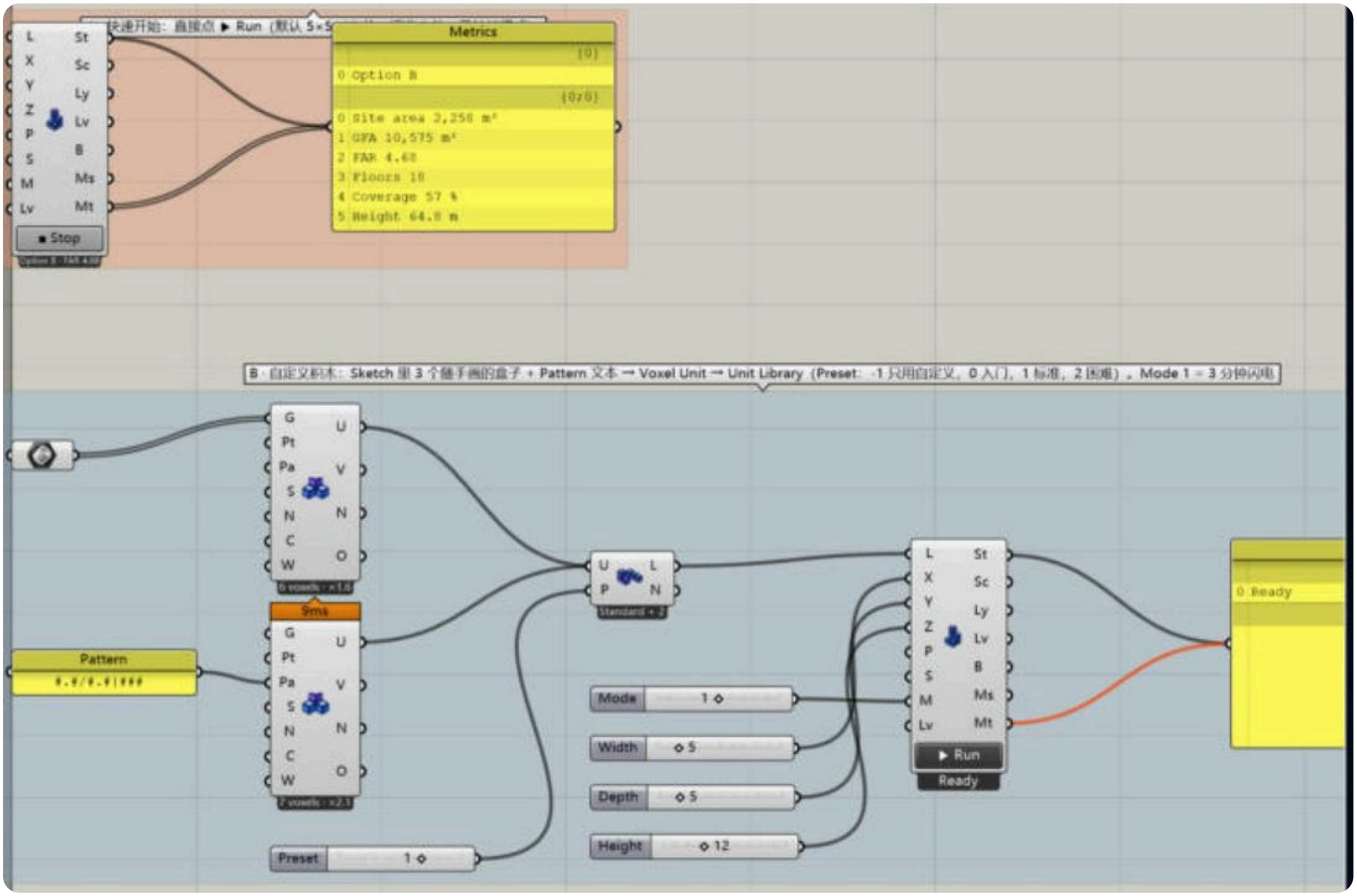
A: the six bonds in turn. B: pick a bond and a mode. C: a custom wall from closed rectangles.



Swift.gh

A: normal exposure, classic mode. B: tune exposure, opening height, bay width and seed.

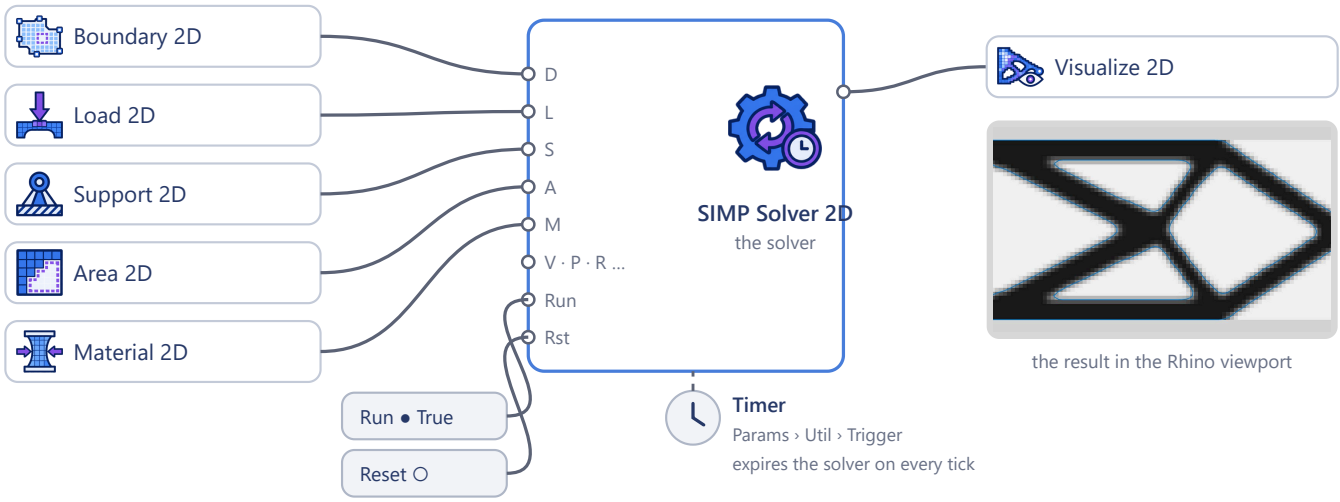
A real canvas: group B of VoxStack.gh



Two Voxel Units (left: sketched geometry; below: a text pattern) feed the Unit Library, which feeds Massing Study. The sliders set the well size and the mode; ▶ Run under the component starts the game.

2D pixel topology optimisation

Draw a design domain and its conditions in a plane; the solver moves material, step by step, to where the load path needs it (SIMP: minimum compliance under a volume limit). The typical wiring:



Boundary 2D
Closed planar curves → a square pixel grid; nested curves are holes. Works in the XY, XZ or YZ plane.

Load 2D
A force at points or along curves (a world vector), spread evenly along a curve.

Support 2D
Fixes points or curves in X and/or Y: both = pinned, one = roller.

Area 2D
Regions that stay solid (a slab) or stay empty (an opening).

Material 2D
Young's modulus E and Poisson's ratio ν (plane stress).

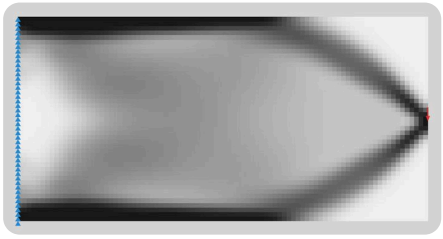
SIMP Solver 2D
While Run is true each tick advances Steps iterations, and the Timer keeps it going until it converges; Reset starts over. V volume fraction, P penalty, R filter radius.

Visualize 2D
Three display modes (density, solid, smooth), a threshold outline curve you can model with, load and support symbols.

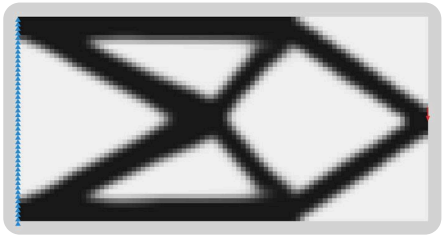
How it converges



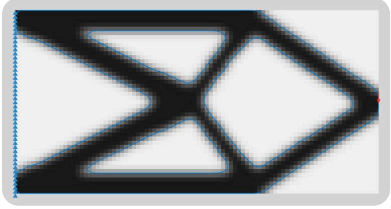
Step 0 · uniform density



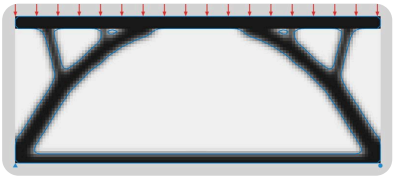
Step 5 · the load path appears



Step 20 · the members have formed



Smooth mode · with the outline curve

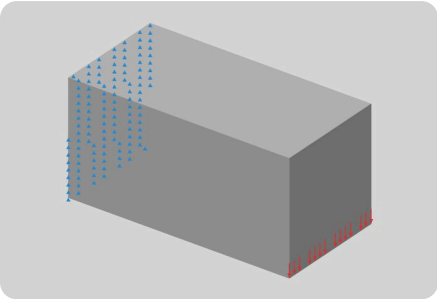


Bridge elevation · line load + solid deck

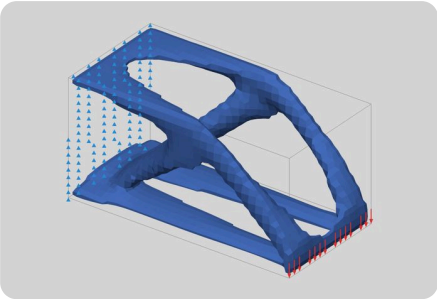
Edit while it runs. Drag a load or a support, or change a parameter, and the solver carries on from the current density; press Reset to start from scratch.

3D voxel topology optimisation

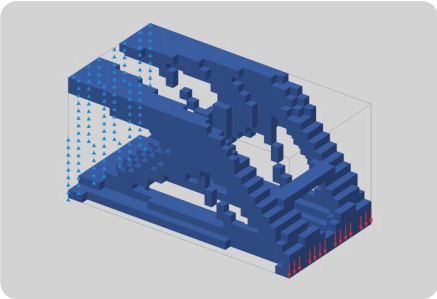
The same wiring with the 3D components: the domain is a closed solid, conditions sit on points, curves or faces, and the solver uses multigrid-preconditioned conjugate gradients.



Step 0 · left end fixed, line load at the bottom right



Step 10 · smooth isosurface (Mode 1)



Voxel blocks (Mode 0) · the same result

Boundary 3D

A closed Brep or mesh → cubic voxels. Resolution = voxels along the long side (default 32, at most 160); solids inside are holes.

Load 3D

A force on points, curves or faces, default (0, 0, −1), shared by the nodes it touches.

Support 3D

Fixes points, curves or faces in X, Y and/or Z.

Area 3D

Closed solids that stay solid or stay empty.

Material 3D

E and ν (isotropic 3D).

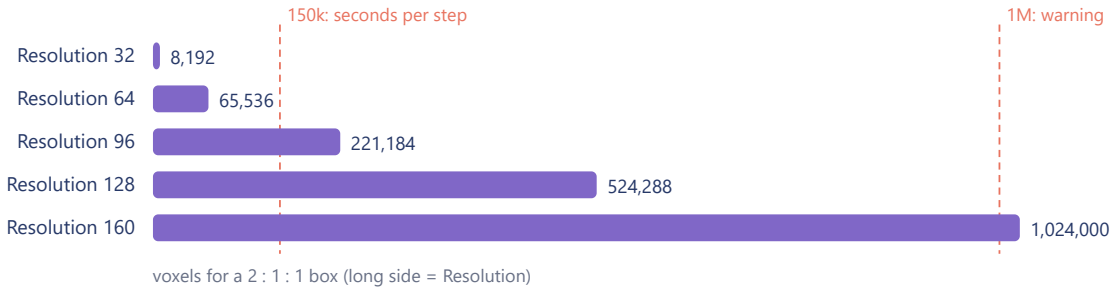
SIMP Solver 3D

Defaults: volume fraction 0.2, filter radius 1.5, 2 steps per tick.

Visualize 3D

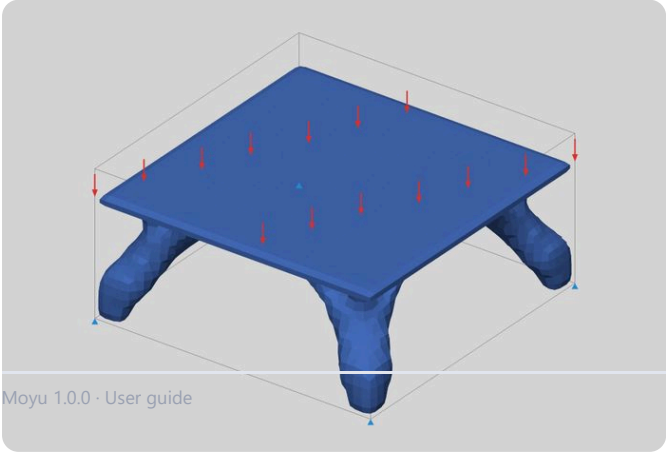
Mode 0 voxel blocks, Mode 1 smooth isosurface; also outputs a closed mesh and per-voxel densities.

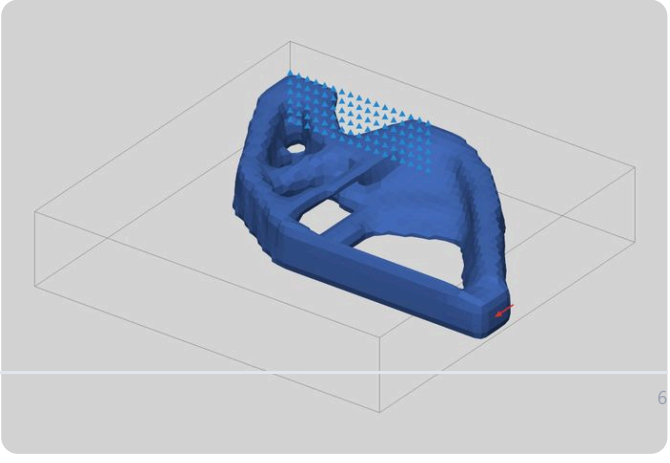
Choosing Resolution: the voxel count grows with its cube



Limits shared by 2D and 3D

- Steps is at most 50; each tick stops after about 1.5 s and leaves the rest to the next one, so Rhino stays responsive. Esc stops it sooner.
- Filter radius: 1–12 in 2D, 1–6 in 3D. NaN or infinity falls back to the default, with a warning.
- Status: **Waiting** = an input is missing; **Unstable** = a mechanism (add supports); **Error** = see the message.

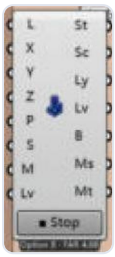




Four games, one set of controls

The games draw in the active Rhino viewport; the Grasshopper canvas only starts and sets them up. Every game starts slow and easy and ramps up; all of them are silent.



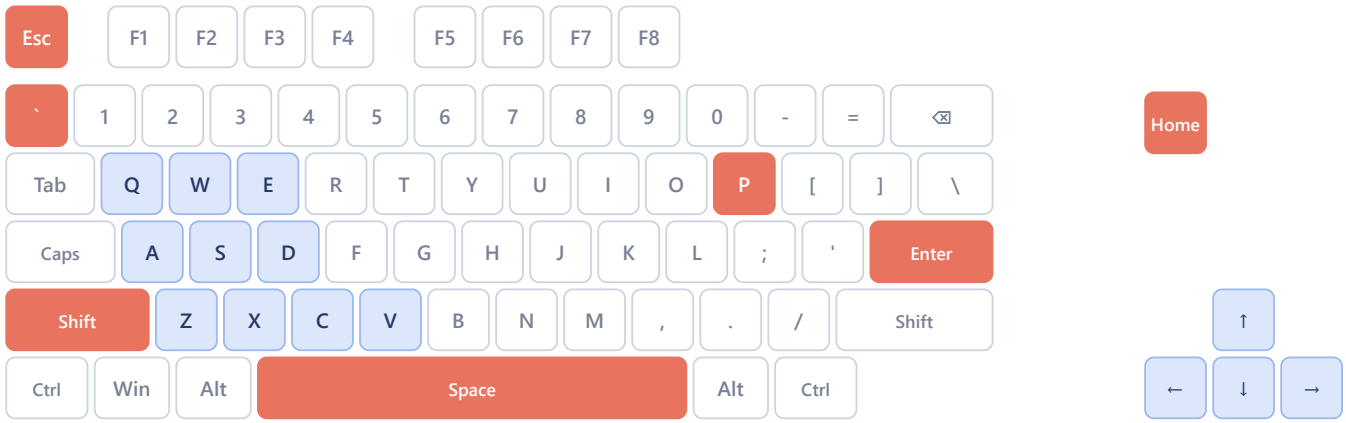


Start and stop

Press ▶ Run under a game component to start and ■ Stop to end. Run hands the focus to the Rhino window, so the keys work at once. A disabled component ignores Run.

The outputs (score, mass, route, metrics…) refresh only when you pause, switch to the boss screen or finish, so Grasshopper does not recompute while you play.

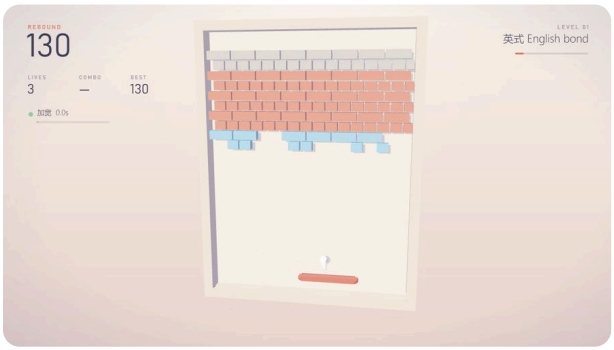
Keyboard



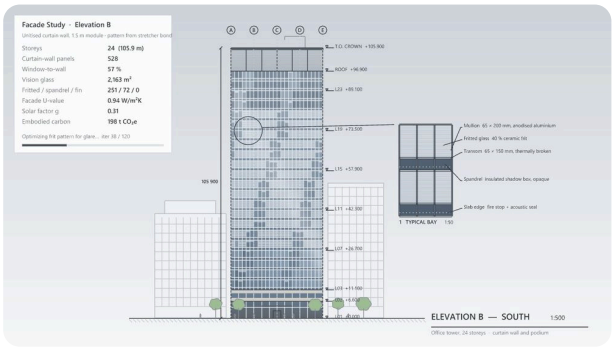
Coral: the same in every game. Light blue: used by some games (see each game's page).

Esc P	Pause. While paused, Esc resumes and Shift + Esc quits.
Enter Space	Play again on the results screen (Esc quits).
Home	Back to the game's default view.
`	The boss key: the key left of 1 (` on a US keyboard).
Left / right drag	Orbit the view; the middle button does nothing in the game viewport.

The boss key



Playing



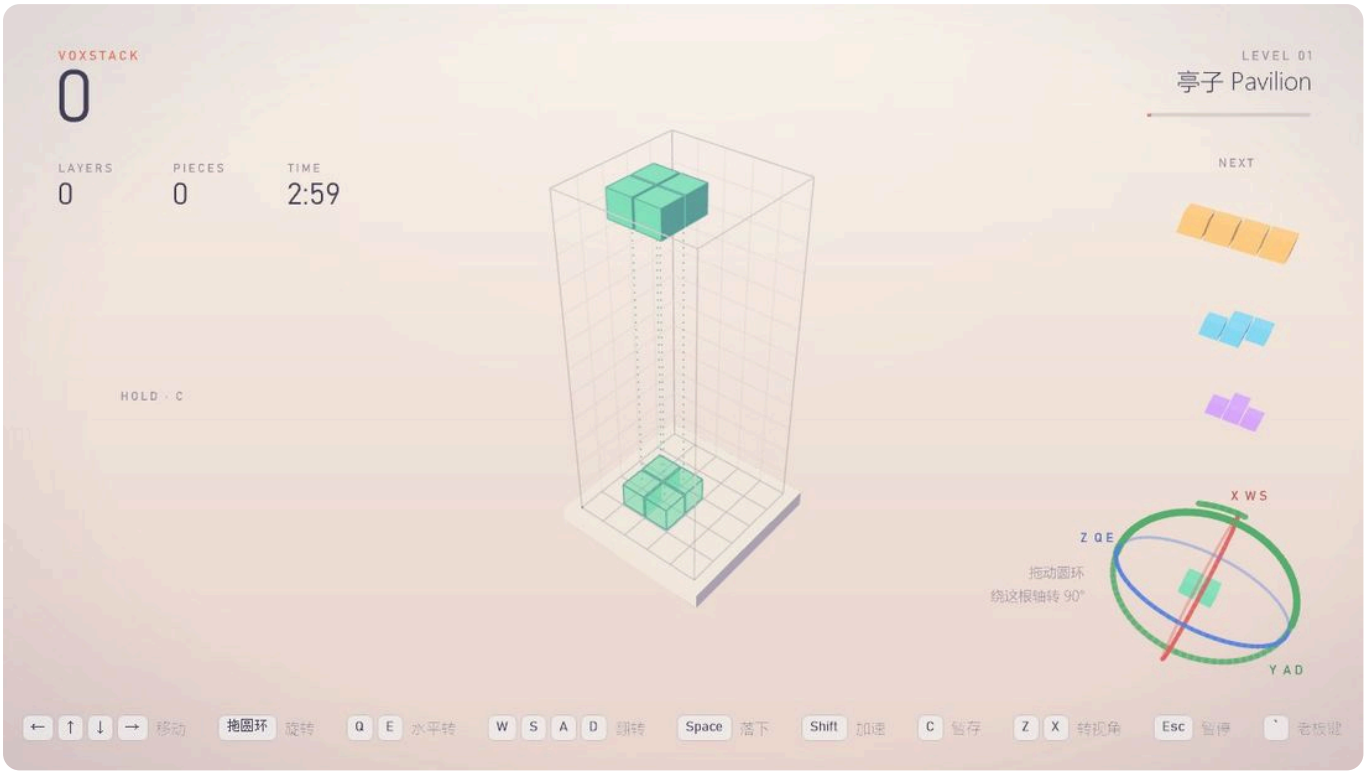
Boss screen: the same wall drawn as a tower elevation

One frame turns the game into a serious drawing whose numbers come from the game state; press again for a 3-2-1 countdown back. It also switches by itself when Rhino loses focus (Alt+Tab). In the boss screen the keyboard belongs to Rhino.

Who gets the keyboard. Keys go to the game only while the Rhino window (or the game's floating viewport) has the focus; Ctrl and Alt shortcuts always pass. When a dialog, the Grasshopper window or a Rhino command takes the keyboard, the game pauses (countdowns too); click the viewport and press Esc to carry on. An East Asian input method can stay on. High scores are kept in `%APPDATA%\Moyu\`.

VoxStack 3D falling blocks

Pieces fall into a 5 × 5 × 12 well; a layer of 25 cells clears when it is full. Turn the view as you like: the moves follow it.



A Rhino-style gumball sits in the bottom right: drag along a ring for about 30° and the piece turns 90° about that axis. The next pieces are on the right.

← ↑ ↓ →	Move (↑ is always away from you on screen)
Drag a ring	Turn 90° about that axis
Q E · W S · A D	Turn about the vertical · tip forward/back · roll left/right
Space · Shift	Hard drop · hold to fall faster
C	Hold the current piece
Z X · V	Turn the view 90° · top view



Massing Study

The game. Width / Depth / Height of the well (4–10 × 4–10 × 8–20), Size of a voxel (default 7.2 m), Mode 0 marathon / 1 three-minute blitz / 2 zen, Level to start at.



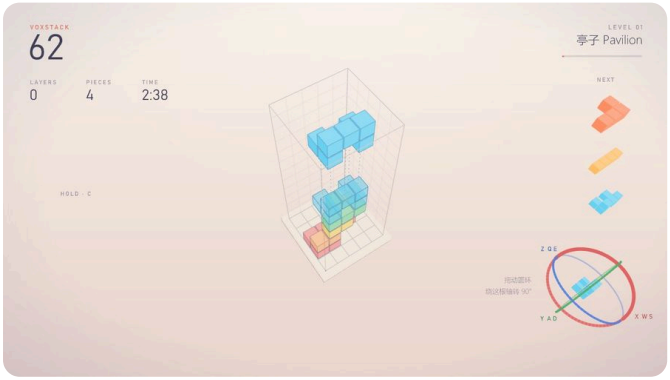
Voxel Unit

A custom piece from a sketched closed solid, voxel centre points, or a text pattern `###/#.#|#..` (/ between rows, | between layers). Up to 10 cells, ≤ 4 per side, face-connected.

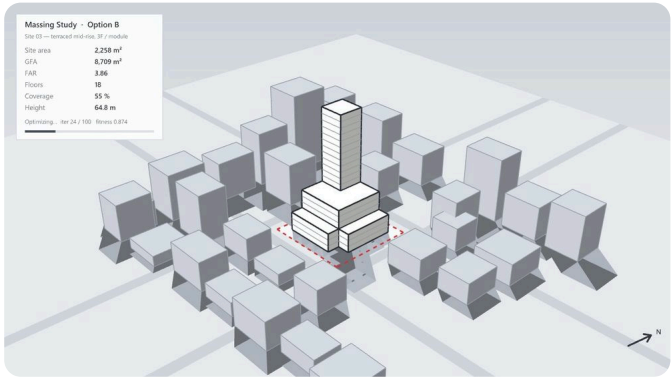


Unit Library

A built-in set (−1 custom only / 0 beginner / 1 standard eight / 2 hard) plus your custom pieces.



A custom piece · "Gate" is dealt from level 1. A cell takes 2.4 s to fall at first and 8 % less each level; the first two levels deal only flat built-in pieces.



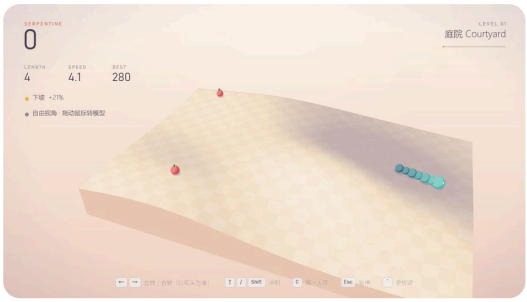
Boss screen: a massing study · One cell = a 7.2 m bay × 3 storeys, with the site line, neighbours, sun shadows and floor-area ratio.

Serpentine a snake on any surface

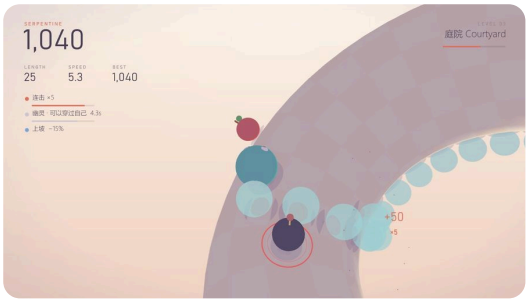
The snake moves cell by cell along the surface's UV grid. Closed directions (tube, torus, sphere) wrap around, open edges are walls and trimmed holes are obstacles; downhill is faster, uphill slower, so the surface is the level.



First person (default) · The camera rides behind the head; ahead is always up the screen.



Free view · press C to see the whole site



Torus · wraps both ways

← A · → D	Turn left / right relative to the head, whatever the camera does
Hold ↑ W Shift	Sprint ×1.8
C	First person ⇄ free view (dragging orbits only in free view)

Path Study

The game. Surface (the built-in terrain if empty; trimmed surfaces work), Cells along the long side (default 28), Mode 0 classic / 1 three-minute blitz / 2 zen (walls just stop you), Level, Slope effect on speed.

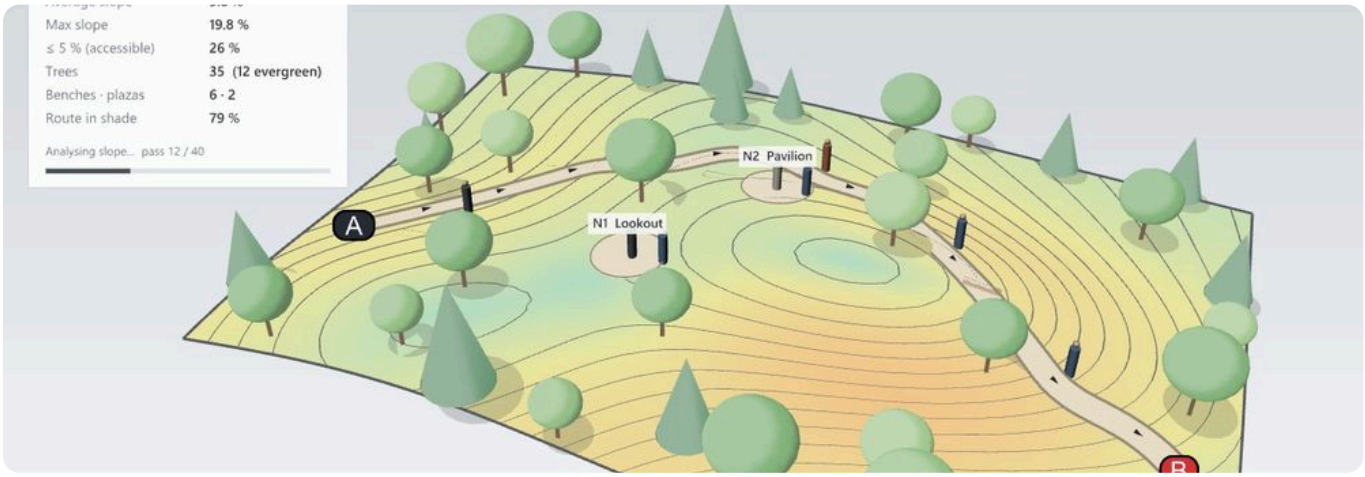
Items

- Apple +1
- Grape +3
- Melon +5, gone after 8 s
- Bomb −3, from level 3
- Ice: 6 s slower
- Ghost: 6 s through yourself

Eating again within 4 s builds a combo (up to ×5). Every five fruit is a level; five chapters: Courtyard, Meadow, Hills, Canyon, Ridge.

Site Surface

A site in one click: 0 rolling terrain, 1 torus, 2 tube, 3 saddle roof, 4 sphere; Size default 40 m, Seed varies the terrain.



Boss screen: a landscape plan · The surface becomes a slope analysis with contours and the snake a paved path, with trees, benches, people and small plazas; length, climb and the accessible share are measured from the snake.

Rebound brick breaker on a façade

The wall is laid in a real brick bond, one per level. Materials take different numbers of hits; broken bricks may drop power-ups, which wear off after a number of uses.



Active power-ups and their remaining uses are on the left; a new one is named above the paddle. Bricks hit in a row between paddle touches build a combo (up to ×8).

Materials

- Glass: 1 hit
- Brick: 2 hits
- Concrete: 3 hits
- Steel: unbreakable

Power-ups (by uses, not time)

- Wide · 8 catches
- Multi-ball
- Slow · 6 catches
- Sticky · 3 catches
- Pierce · 6 bricks
- +1 life
- Shrink (bad) · 5 catches

The first six drops are one of each good power-up.

← → · A D · mouse	Move the paddle
Space	Serve, release a stuck ball

Facade Study

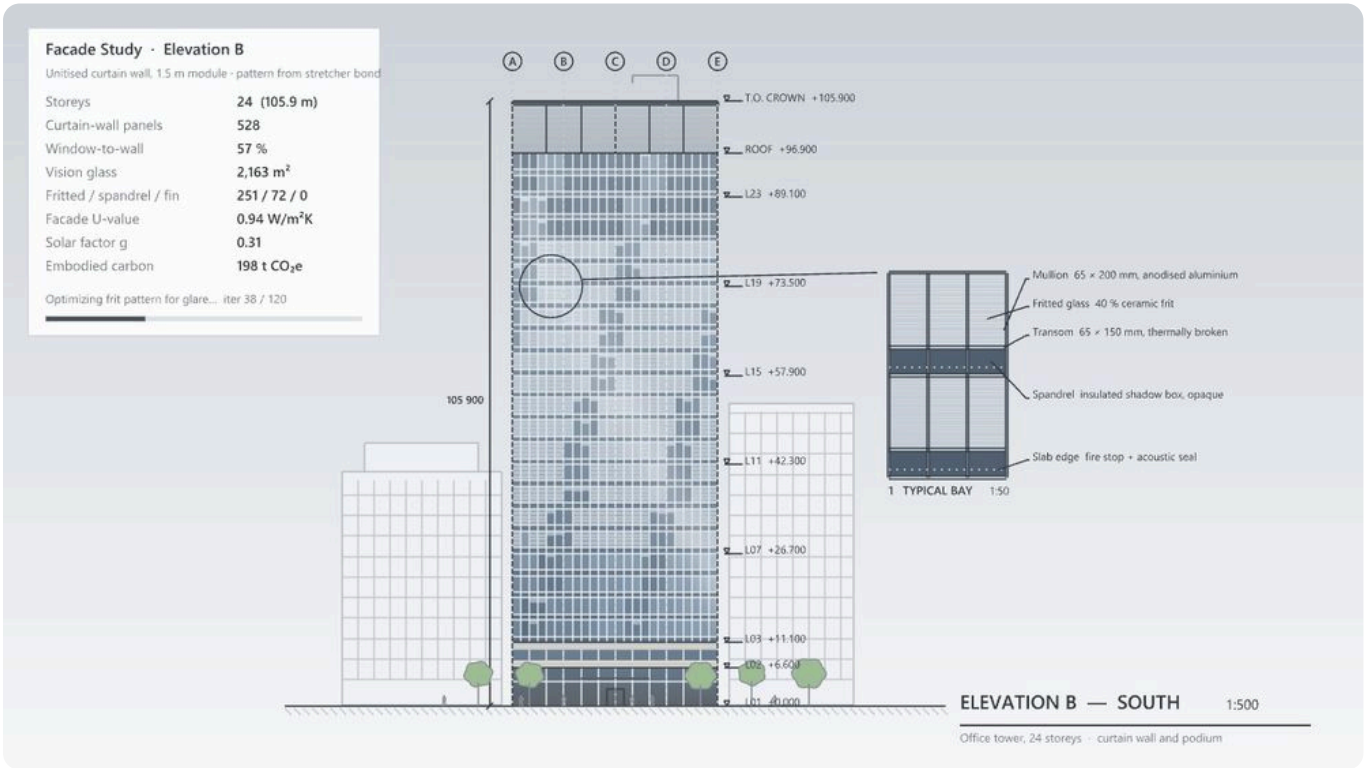
The game. Wall (the six bonds in turn if empty), Mode 0 classic with 3 lives / 1 three-minute blitz / 2 zen (no lives lost), Level, Plane (World XZ by default), Size of a module (default 2.4 m).

Brick Bond

Type −1 rotates, 0–5 picks a bond; Courses sets the rows. Or feed closed Curves as panels, sorted by size into glass, brick and concrete.



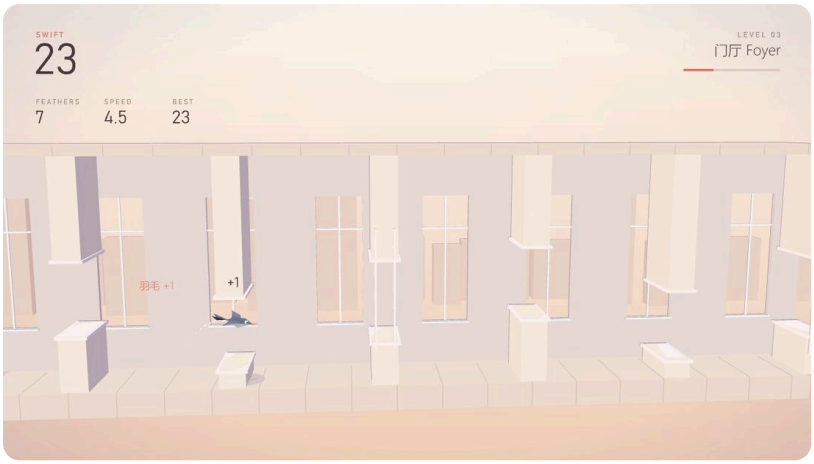
Six bonds · Stretcher, stack, English, Flemish, screen wall and arch (shown).



Boss screen: a curtain-wall elevation · Each panel takes the brick behind it: cleared bricks are clear glass, the rest fritted glass, spandrels or aluminium; window-to-wall ratio and U-value are computed from the wall.

Swift one-key flight through a section

A swift flies through an endless building section: each press is one wing beat up, and it sinks when you let go. Each wall's opening you pass is a point; touching a wall or the floor ends the flight.



Feathers float between the walls. The first openings are tall, central and slow; later ones get smaller, faster and more uneven, and from the 26th wall some have sliding sashes.

Space ↑ W	One wing beat (a left click too)
Right drag	Orbit the view

Every 10 points is a level and every 3 levels a chapter: Foyer, Cloister, Atrium, Lightwell, Dusk. Each difficulty was flown by an autopilot for 150 s on five seeds, so every level can be passed.



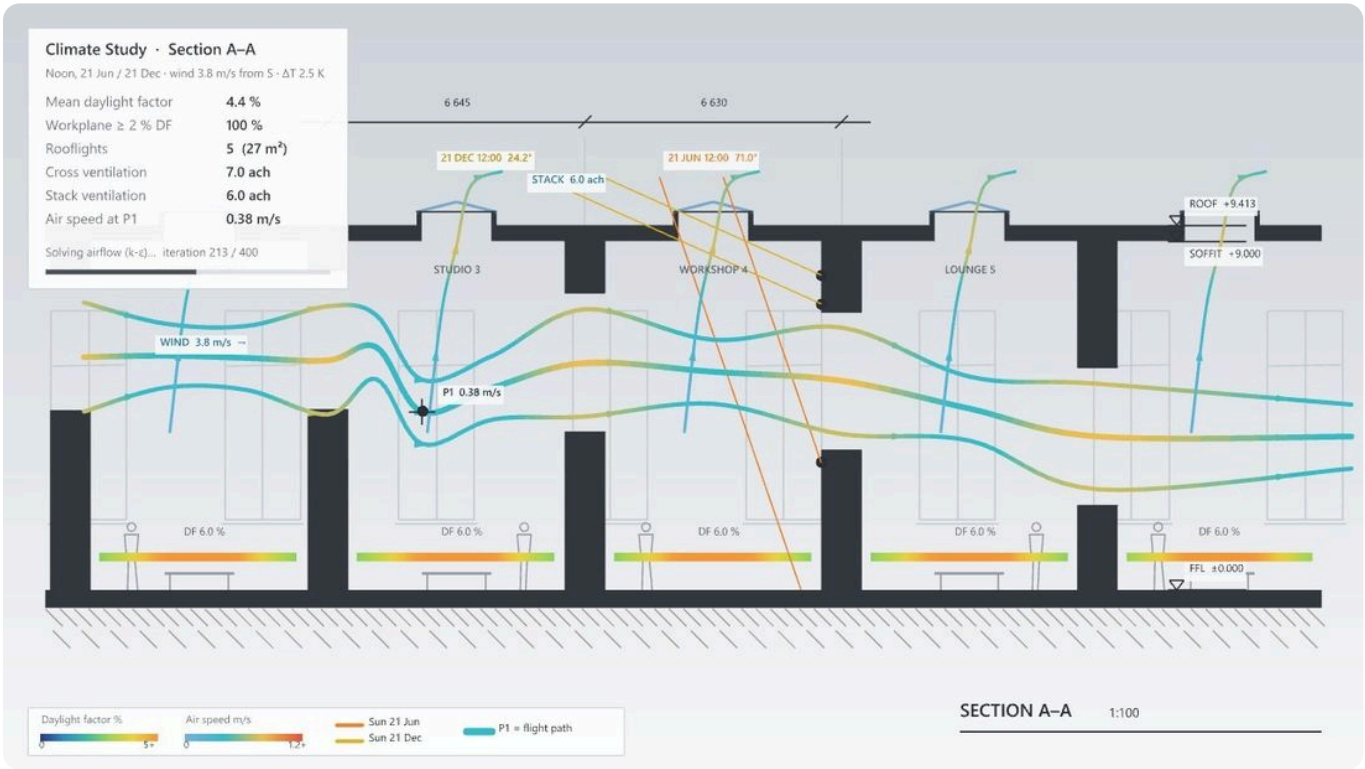
Results · Enter or Space flies again, Esc quits.

Airflow Study

The game. Grid (from Storey Grid; normal exposure and classic mode if empty), Plane (World XZ by default), Size of a unit (default 0.75 m). Outputs the flight path, the section outline and metrics.

Storey Grid

Exposure 0 sheltered / 1 normal / 2 exposed (faster and tighter), Mode 0 classic / 1 zen (bump through, bounce off the floor), Opening height 0.8–1.4, Bay width 0.8–1.5, Seed for a fixed layout (0 = new each run). Best scores are kept per setting.



Boss screen: a daylight and ventilation section · It is the game's own section: walls and openings where they were, plus rooflights, noon sun at both solstices, a daylight-factor band and airflow lines; the main streamline is the swift's own flight.

Troubleshooting

The game ignores my keys
The Rhino window does not have the focus. Click the game viewport, then press Esc.

▶ Run does nothing
The component is disabled (locked), or it shows a warning balloon that says why.

The solver says Unstable
The structure is a mechanism that can move or spin. Add supports, or fix them in both directions.

"Moyu data cannot be internalised"
Moyu values stored with right-click › Internalise cannot be restored. Reconnect the component that made them.

The game paused by itself
A dialog, the Grasshopper window or a Rhino command took the keyboard. Finish there, click the viewport, press Esc.

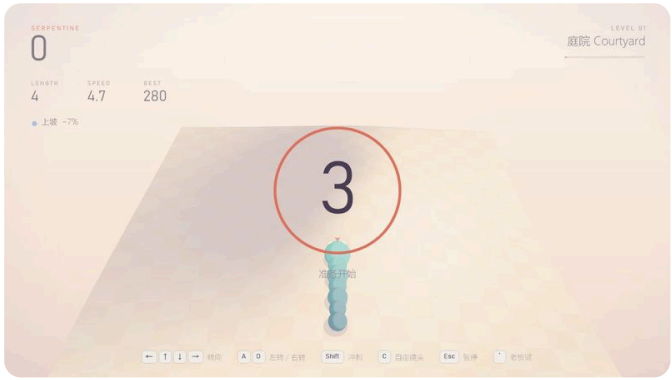
The solver says Waiting
An input is missing, usually the loads or the supports.

3D steps are slow
Too many voxels. Lower Resolution or the filter radius; above 150k voxels a step takes seconds, above 1M you get a warning (page 6).

"No Timer drives this solver"
There is no Timer in the document. Wire Params › Util › Trigger to the solver, or recompute by hand.



The pause screen · Also shown when another window takes the keyboard; click the viewport and press Esc.



The 3-2-1 countdown · Coming back from a pause or the boss screen counts down first.

Where Moyu keeps its files

Package Manager install
`%APPDATA%\McNeel\Rhinoceros\packages\8.0\moyu\1.0.0\`
net48\ and net7.0\ (the two builds), examples\, the two guide PDFs.

Copied by hand
`%APPDATA%\Grasshopper\Libraries\Moyu\`
The one Moyu.gha you copied.

High scores
`%APPDATA%\Moyu\`
One plain-text file per game, such as swift-scores.txt. The plugin writes nowhere else.

Uninstall

Package Manager
`_PackageManager` › Installed › moyu › Uninstall

Copied by hand
Delete `%APPDATA%\Grasshopper\Libraries\Moyu\`

High scores
To clear them, delete `%APPDATA%\Moyu\`

About. Moyu 1.0.0 · by Ming · MIT licence. The code, tests and documentation were written mostly by Claude Code; the icons were drawn with OpenAI Codex. Every component input, every key, the design notes and the development time log are at https://github.com/ChenmingHe0126/moyu ; please report problems as GitHub issues. Optimisation results are for study and concept design and come with no warranty.