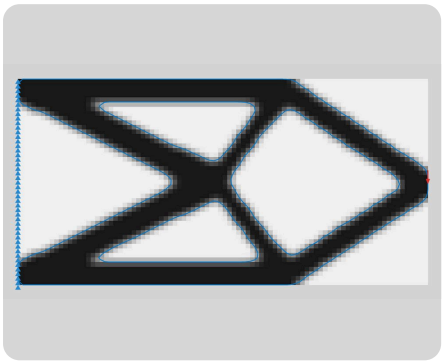
GRASSHOPPER PLUGIN • RHINO 8 FOR WINDOWS

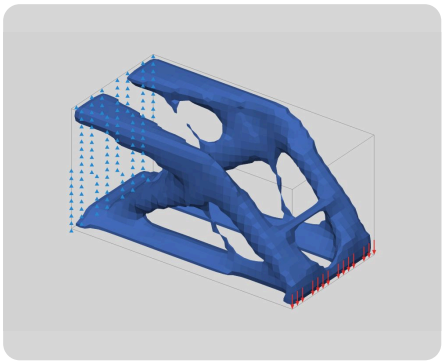
Moyu 1.0.0

交互式拓扑优化，加上四个伪装成建筑研究工具的视口小游戏

用户手册 · 一个用 AI 编程代理快速开发插件的研究与教学项目



2D 像素拓扑优化 · 2D



3D 体素拓扑优化 · 3D



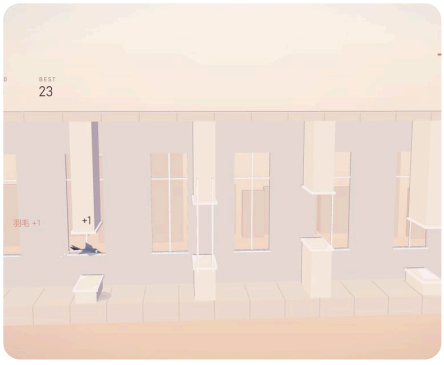
VoxStack 叠楼 · Massing



Serpentine 蜿蜒 · Landscape



Rebound 回弹 · Facade



Swift 雨燕 · Section

目录

关于这个项目 2
安装与面板 3
示例文件 4
2D 拓扑优化 5
3D 拓扑优化 6
游戏的共同操作 7
VoxStack 叠楼 8
Serpentine 蜿蜒 9
Rebound 回弹 10
Swift 雨燕 11
常见问题与卸载 12

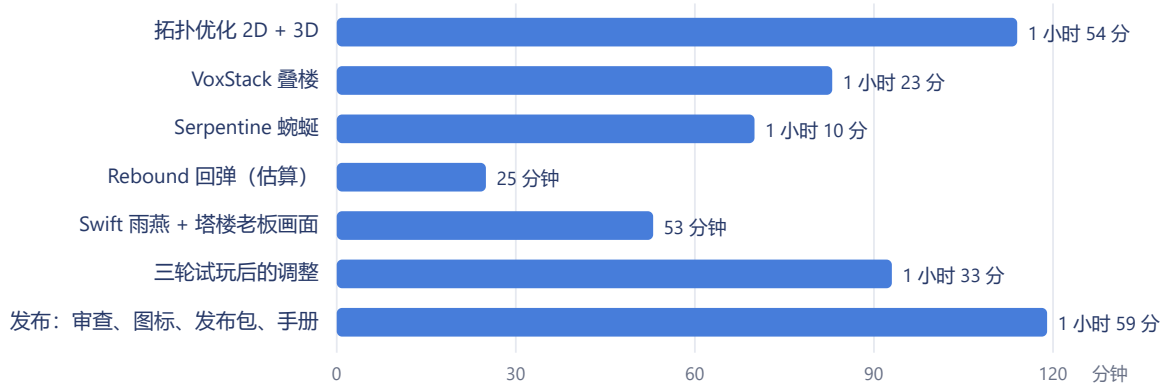
研究与教学项目。代码、测试和文档几乎都由 Claude Code（Anthropic 的编程代理）编写，作者提需求、看结果、试玩。拓扑优化的结果只用于学习和概念设计，不能替代工程分析。插件不联网，只在 `%APPDATA%\Moyu\` 写最高分。MIT 许可，源码与开发记录：https://github.com/ChenmingHe0126/moyu

01 关于这个项目

用 AI 编程代理快速做一个 Rhino 插件

Moyu 是一个研究和教学项目：看一个编程代理能多快、多可靠地做出一个完整的 Grasshopper 插件，包括数值求解器、自定义视口显示、键盘鼠标交互和测试。几乎所有代码、测试和文档都由 **Claude Code** 编写；作者提需求、看结果、在 Rhino 里试玩；图标由 OpenAI Codex 生成。

每个部分大概用了多久



Claude Code 的工作时间，按会话记录的时间戳逐轮相加。作者看结果、试玩的时间和 Rhino 本身的故障（约 34 分钟）不算。共约 9 小时 17 分，分布在四天里。明细：仓库里的 docs/dev-time-stats.md。

每个功能都按同一套流程做

1 方案
在 Rhino 里做原型截图，写一页图文方案给作者确认。

2 规则与测试
求解器和游戏规则写成不依赖 Rhino 的核心库，配单元测试（现在 351 项）。

3 Rhino 端
Grasshopper 组件、显示管线、键盘鼠标钩子、镜头。

4 实机试玩
在真 Rhino 里用真实按键和鼠标测，截图看画面再改。

5 记录
文档、用时统计、提交。发布前再做一次多代理代码审查。

每个部分研究什么

拓扑优化（数值） · 代理能不能写对带测试的工程数值代码：有限元、SIMP + 最优准则法、密度过滤、2D 带状 Cholesky、3D 多重网格共轭梯度；以及怎样用 Timer 让优化在 Grasshopper 里实时迭代、边算边改。

四个小游戏（交互与视口） · 从一个 Grasshopper 组件出发，Rhino 视口能做到什么：用显示管线稳定地画整个场景和 HUD；安全接管键盘鼠标，并在对话框、Grasshopper 和 Rhino 命令需要时让出去；控制镜头；一帧之内把同一份数据换成另一种图（老板键）。

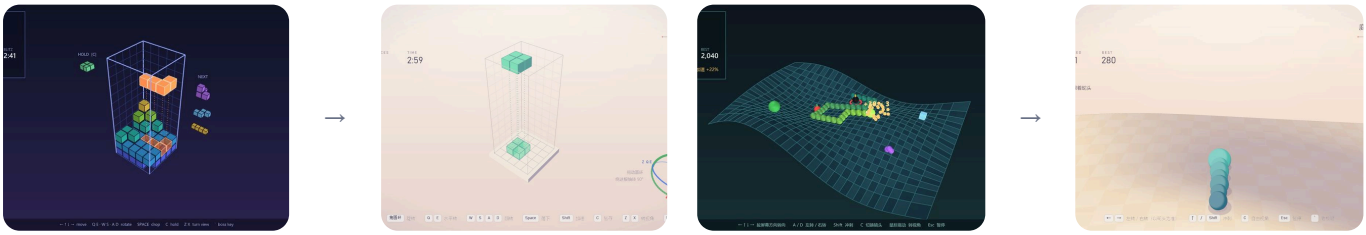
VoxStack · 视角在转时的三维移动与旋转；可拖动的 gumball。

Serpentine · 任意曲面变成场地（UV 网格、闭合、修剪、坡度）；第一人称镜头。

Rebound · 平面上的实时碰撞和鼠标控制；参数化砌法当关卡。

Swift · 单键节奏、程序生成关卡，用自动驾驶验证每关能过。

方案原型和成品



左：方案阶段 Claude Code 在 Rhino 里拼出来给作者确认的原型截图。右：作者试玩后要求统一改成极简风格，两个游戏的画面一起重做（约 19 分钟）。

安装

需要 Windows 版 Rhino 8。两种运行时（默认的 .NET Core 和 .NET Framework）都能用；没有在 Mac 版上测试过。

A · 包管理器（推荐）

1. 在 Rhino 里运行 `_PackageManager`
2. 搜索 **moyu** → Install
3. 重启 Rhino

B · .yak 文件

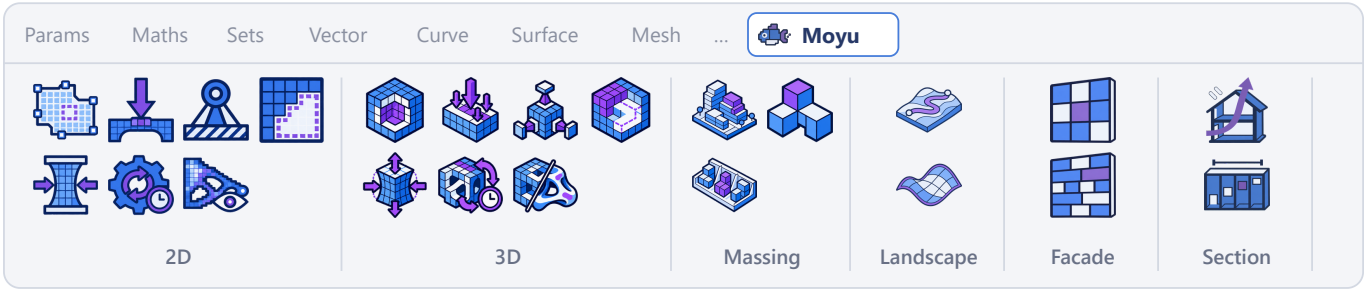
1. 把 food4rhino 下载的 `moyu-1.0.0-rh8_0-win.yak` 拖进 Rhino 窗口
2. 重启 Rhino

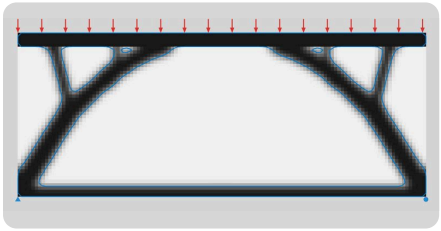
C · 手动复制

1. 把 **一个** `Moyu.gha` 复制到 `%APPDATA%\Grasshopper\Libraries\Moyu\`
2. .NET Core 用 net7.0，.NET Framework 用 net48（`_SetDotNetRuntime` 可查）
3. 被 Windows 阻止时：右键 › 属性 › 解除锁定，重启 Rhino

Moyu 页签上的六个面板

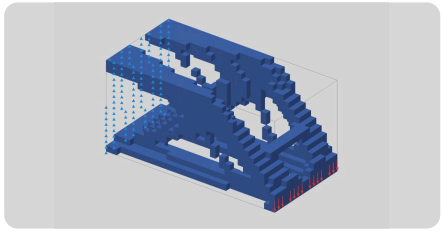
重启后，Grasshopper 工具栏上多出一个 Moyu 页签。四个游戏面板故意起了建筑研究的名字，图标也画成研究工具。





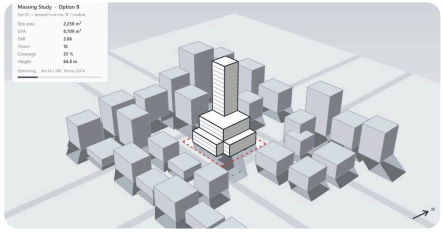
2D

平面像素拓扑优化：边界、荷载、支座、区域、材料 → 求解器 → 显示（第 5 页）。



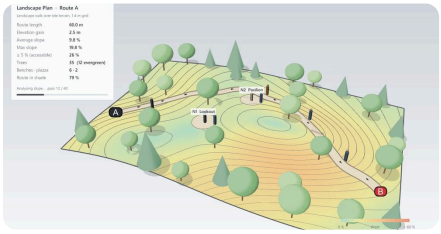
3D

封闭实体的体素拓扑优化，条件可以加在点、线、面上（第 6 页）。



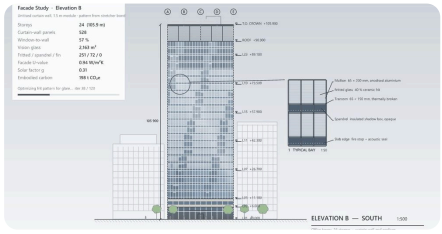
Massing

VoxStack 叠楼：三维落块。图是它的老板画面，一张体块研究（第 8 页）。



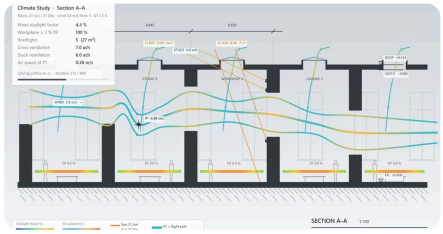
Landscape

Serpentine 蜿蜒：曲面上的贪吃蛇。图是它的老板画面，一张景观平面（第 9 页）。



Facade

Rebound 回弹：按真实砌法排的打砖块。图是它的老板画面，塔楼幕墙立面（第 10 页）。



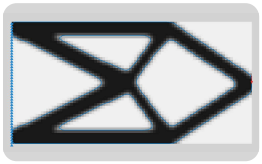
Section

Swift 雨燕：剖面里单键飞行。图是它的老板画面，采光通风剖面（第 11 页）。

从示例开始

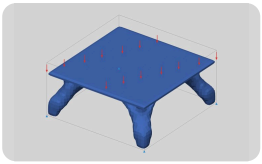
发布包的 examples 文件夹里有 6 个现成的 .gh 文件。在 Grasshopper 里 File › Open 打开，或者直接拖到画布上；画布顶部有两三行中英文说明。

文件在哪：包管理器装的在 `%APPDATA%\McNeel\Rhinoceros\packages\8.0\moyu\1.0.0\examples\`；food4rhino 压缩包里的在 `examples\`。



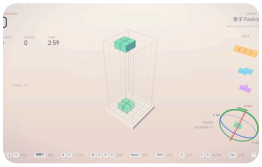
Topo2D.gh

三组 2D：悬臂梁、立面上的桥、带螺栓孔的 L 形支架。每组有自己的 Run 开关和 Reset 按钮，Timer 已接好。



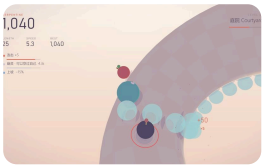
Topo3D.gh

三组 3D：悬臂梁、四角支撑的雨棚、带孔的 L 形支架。



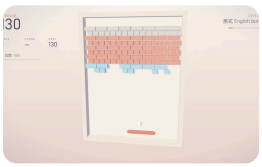
VoxStack.gh

A 直接点 ▶ Run；B 用随手画的盒子和文本图案做自定义积木。



Serpentine.gh

A 内置起伏地形；B 用 Site Surface 换场地（环面、圆管、马鞍屋面、球面）；C 接你自己的曲面。



Rebound.gh

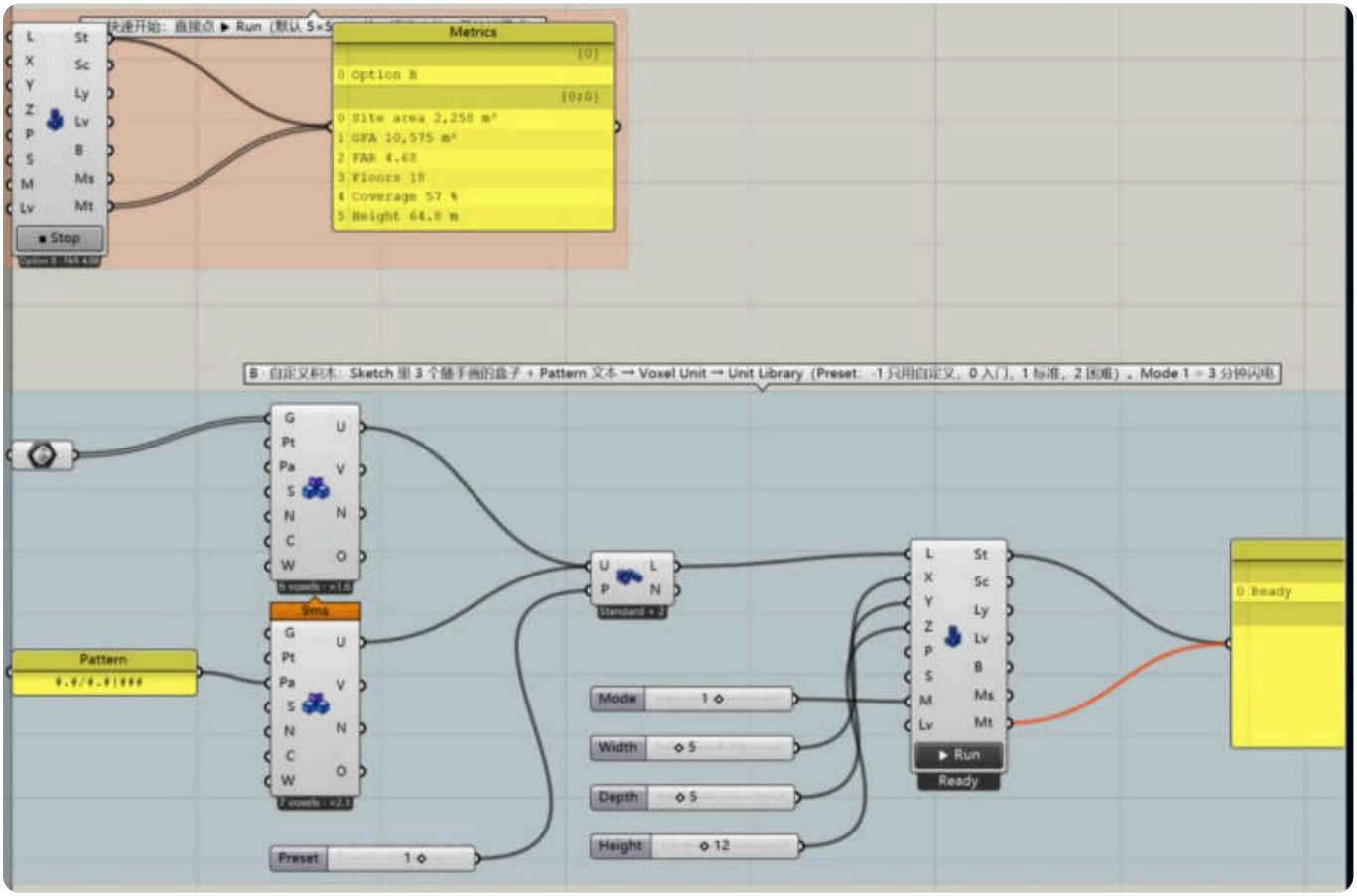
A 六种砌法逐关轮换；B 指定砌法和模式；C 用一组闭合矩形做自定义墙。



Swift.gh

A 普通风压、经典模式；B 调风压、开口高度、开间和种子。

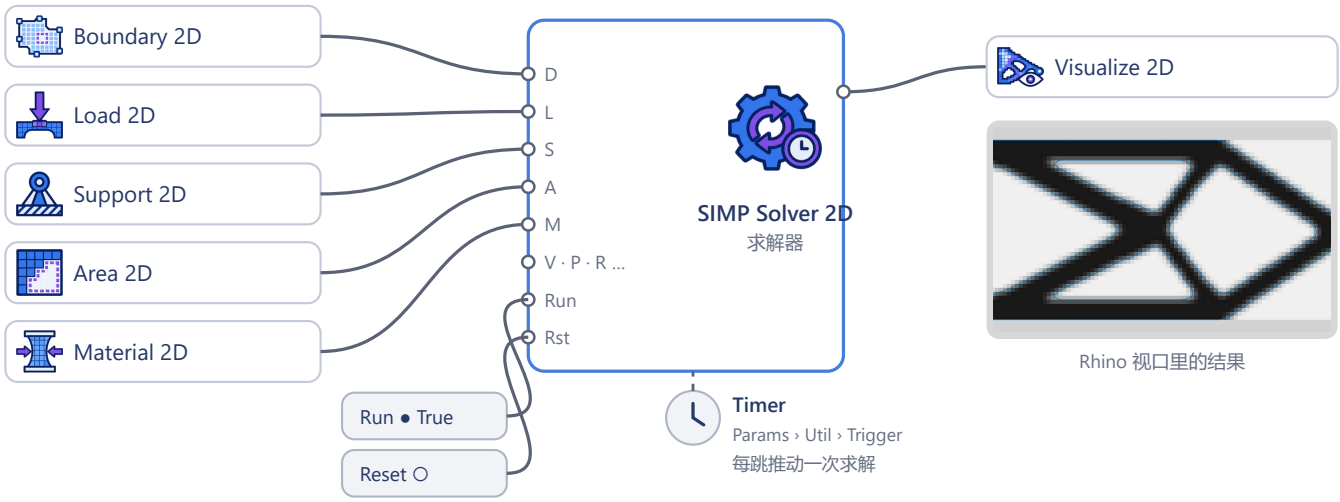
真实画布：VoxStack.gh 的 B 组



两个 Voxel Unit（左：随手画的几何；下：文本图案）接进 Unit Library，再接到 Massing Study。滑块是井的尺寸和模式；组件下方的 ▶ Run 开始游戏。

2D 像素拓扑优化

在一个平面里画出设计域和条件，求解器一步步把材料挪到传力需要的地方（SIMP：体积受限下的最小柔度）。下面是典型连线。



Boundary 2D

闭合平面曲线 → 方形像素网格；套在里面的曲线是孔。自动识别 XY / XZ / YZ 平面。

Load 2D

点或曲线上的力（世界坐标向量），曲线上平均分到经过的节点。

Support 2D

点或曲线约束，分别锁 X / Y：两向 = 铰支，单向 = 滚动支座。

Area 2D

始终实心（如楼板）或始终空心（如洞口）的区域。

Material 2D

杨氏模量 E、泊松比 ν（平面应力）。

SIMP Solver 2D

Run 为真时每跳推进 Steps 步，Timer 一直推到收敛；Reset 从均匀密度重来。V 体积分数、P 惩罚、R 过滤半径。

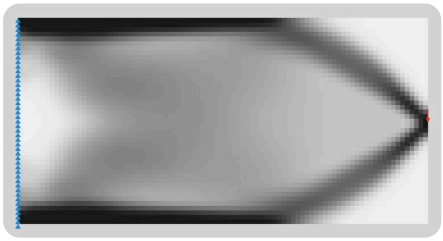
Visualize 2D

三种显示（密度 / 实心 / 平滑）、阈值轮廓曲线（可以直接拿去建模）、荷载与支座符号。

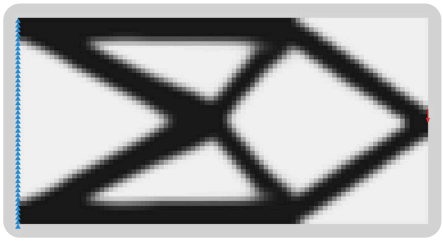
迭代过程



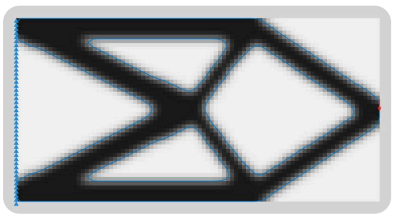
第 0 步 · 均匀密度



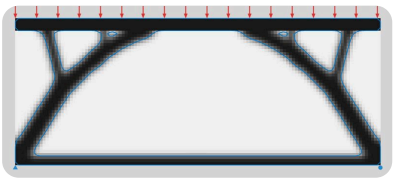
第 5 步 · 传力路径开始出现



第 20 步 · 杆件基本成形



平滑模式 · 和轮廓曲线一起出图

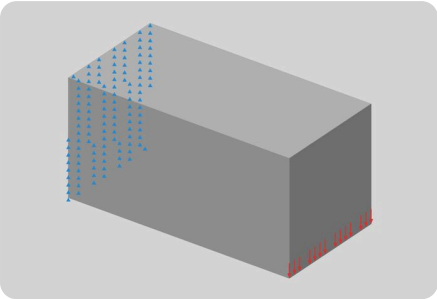


立面上的桥 · 线荷载 + 实心桥面

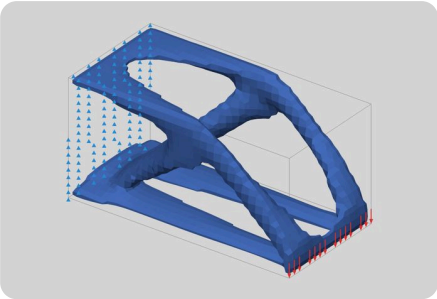
边算边改。运行中拖动荷载、支座或改参数，求解器保留当前密度继续收敛，不用从头再来；想重新开始按 Reset。

3D 体素拓扑优化

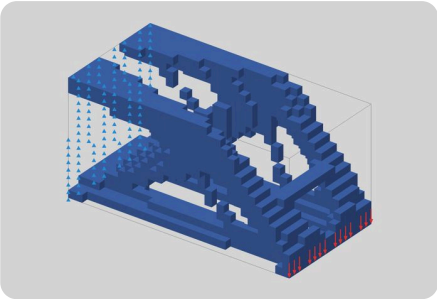
连线和 2D 一样，组件名把 2D 换成 3D：设计域是封闭实体，条件可以加在点、曲线或面上，求解器用多重网格预条件共轭梯度。



第 0 步 · 左端固定，右下边线荷载



第 10 步 · 平滑等值面（Mode 1）



体素块（Mode 0） · 同一个结果

Boundary 3D

封闭 Brep / 网格 → 立方体素。Resolution = 长边体素数（默认 32，最高 160）；套在里面的实体是孔。

Load 3D

点、线、面上的力，默认 (0, 0, −1)，总力平均分到接触的节点。

Support 3D

点、线、面上的约束，分别锁 X / Y / Z。

Area 3D

封闭实体围出的始终实心 / 始终空心区域。

Material 3D

E、ν（三维各向同性）。

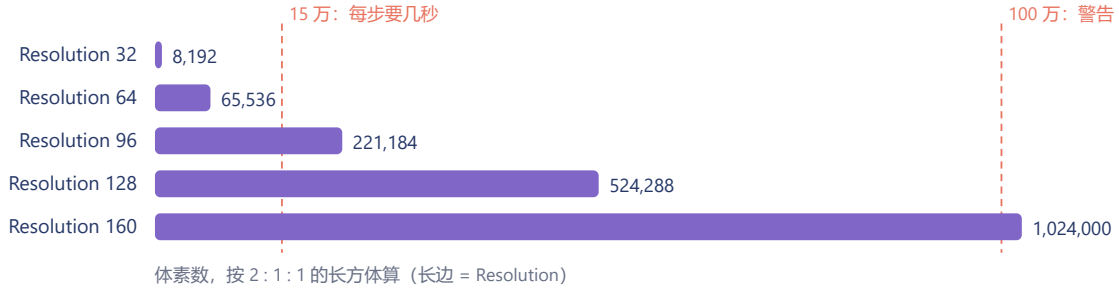
SIMP Solver 3D

默认体积分数 0.2、过滤半径 1.5、每跳 2 步。

Visualize 3D

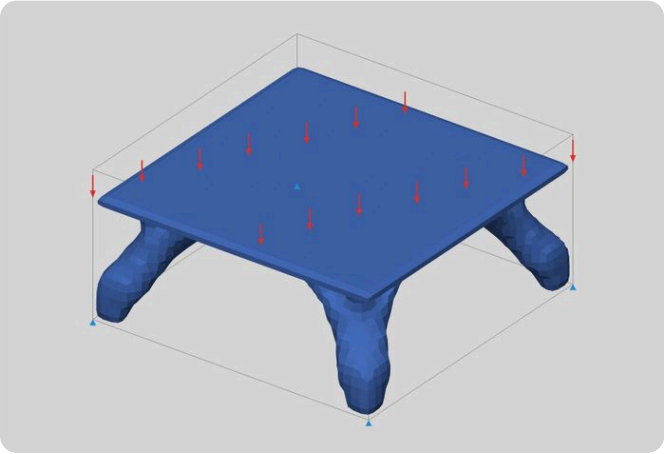
Mode 0 体素块 / Mode 1 光滑等值面；另输出闭合网格和逐体素密度。

选 Resolution：体素数随它的三次方增长

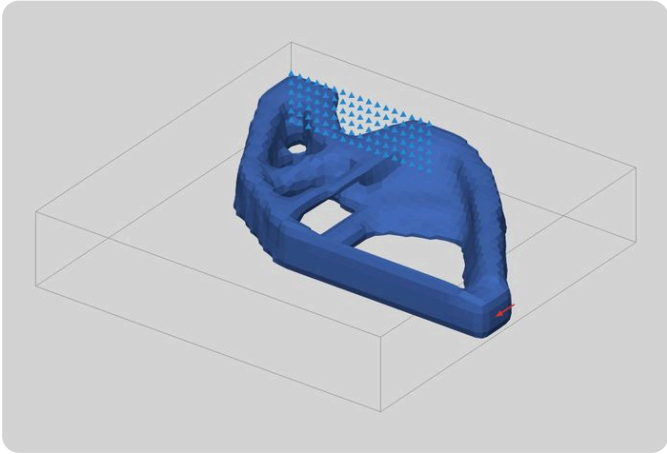


2D 和 3D 通用的上限

- Steps 最多 50；每跳算满约 1.5 秒就停在当前步，剩下的留给下一跳，Rhino 不会卡住。按 Esc 停得更早。
- 过滤半径：2D 1–12，3D 1–6。收到 NaN 或无穷大时按默认值算并警告。
- 状态：**Waiting** 缺输入；**Unstable** 结构是机构（加支座）；**Error** 原因在消息里。



雨棚 · 顶面面荷载、四角铰支、实心屋面板、空心通道。

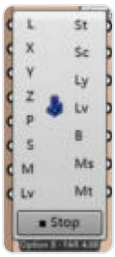


带孔的 L 形支架 · 嵌套圆柱是孔，顶面固定，端部水平荷载。

四个游戏，一套操作

游戏画在当前的 Rhino 视口里，Grasshopper 画布只用来开始和设置。每个游戏开局都很慢、很简单，逐渐变难；默认静音。



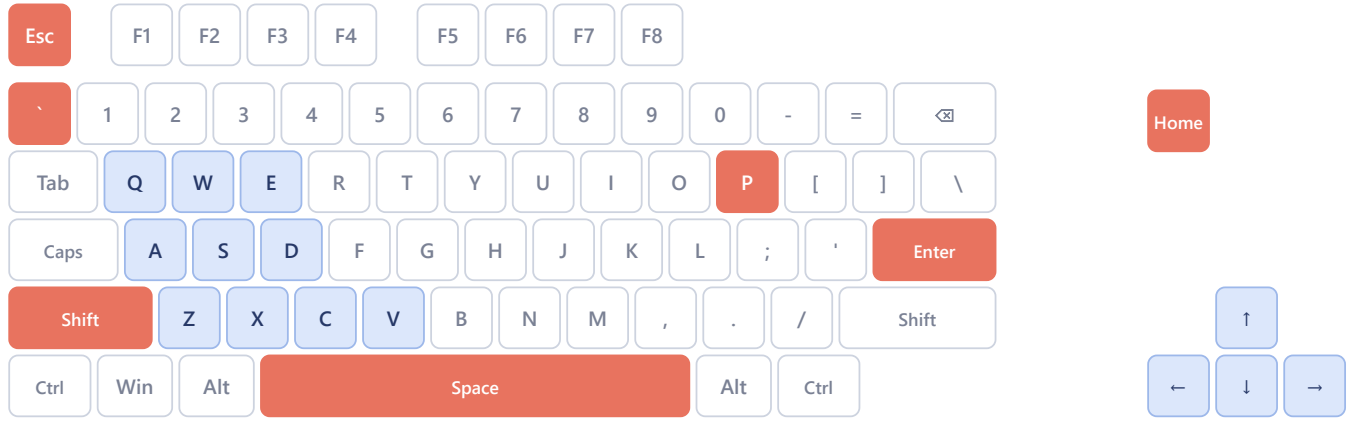


开始和结束

点游戏组件下方的 ▶ Run 开始，再点 ■ Stop 结束。点 Run 会把焦点交给 Rhino 主窗口，一开局按键就有效。组件被禁用时 Run 不起作用。

组件输出（分数、体块、路线、指标……）只在暂停、老板模式和结束时刷新，游戏进行中 Grasshopper 不会重算。

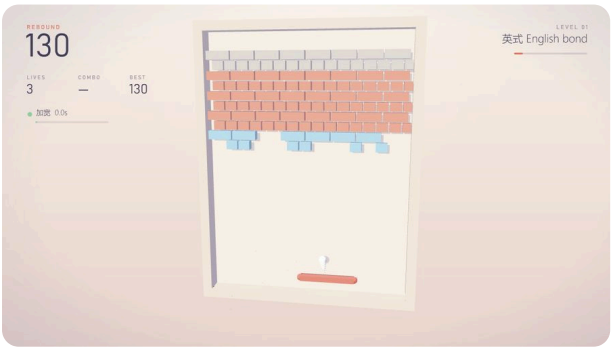
键盘



珊瑚色：每个游戏都一样；浅蓝：部分游戏用（见各游戏页）。

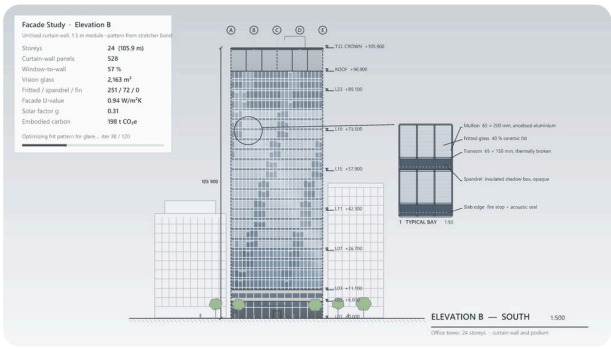
Esc P	暂停。暂停时 Esc 继续，Shift + Esc 退出。
Enter Space	结算画面上再来一局（Esc 退出）。
Home	镜头回到游戏的默认视角。
`	老板键：1 左边的那个键（美式键盘上是 `）。
鼠标左键 / 右键拖动	转视角；中键在游戏视口里不起作用。

老板键



游戏中

`



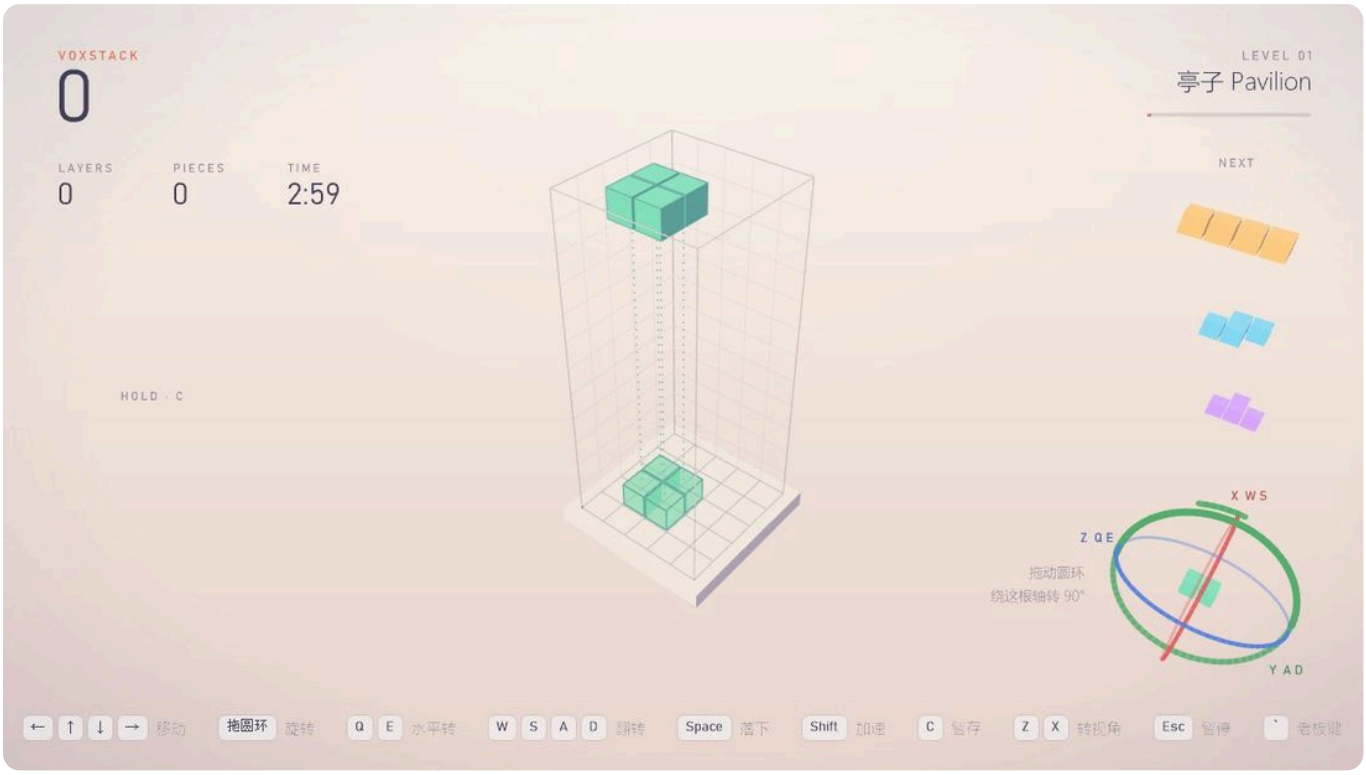
老板模式：同一面墙画成塔楼幕墙立面

一帧之内切成一张正经的设计图，指标按游戏里的数据实算；再按一次 3-2-1 回来。Rhino 失去焦点（Alt+Tab 到别的程序）时自动切过去。老板模式里键盘全部还给 Rhino。

键盘让给谁。只有焦点在 Rhino 主窗口（或游戏所在的浮动视口）时按键才归游戏；Ctrl / Alt 组合键任何时候都放行。对话框、Grasshopper 窗口或 Rhino 命令拿走键盘时，游戏自动暂停（倒计时也停住）；点回视口、按 Esc 继续。中文输入法开着也能玩。最高分存在 `%APPDATA%\Moyu\`。

VoxStack 叠楼 · 三维落块

积木落进一口 5 × 5 × 12 的井，某一层 25 格铺满就消掉。视角可以随便转，移动方向跟着视角走。



右下角固定着一个和 Rhino 一样的 gumball：按住一个圆环顺着它拖约 30°，积木就绕这根轴转 90°。右上是接下来的积木。

← ↑ ↓ →	水平移动（↑ 永远是屏幕上远离你的方向）
拖 gumball 圆环	绕那根轴转 90°
Q E · W S · A D	绕竖轴转 · 前后翻 · 左右翻
Space · Shift	直接落下 · 按住加速
C	暂存当前积木
Z X · V	视角转 90° · 俯视



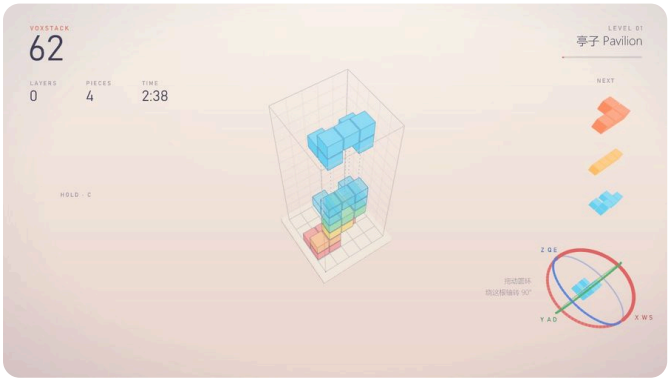
Massing Study
游戏本体。Width / Depth / Height 井尺寸（4–10 × 4–10 × 8–20），Size 体素边长（默认 7.2 m），Mode 0 马拉松 / 1 三分钟闪电 / 2 禅，Level 起始等级。



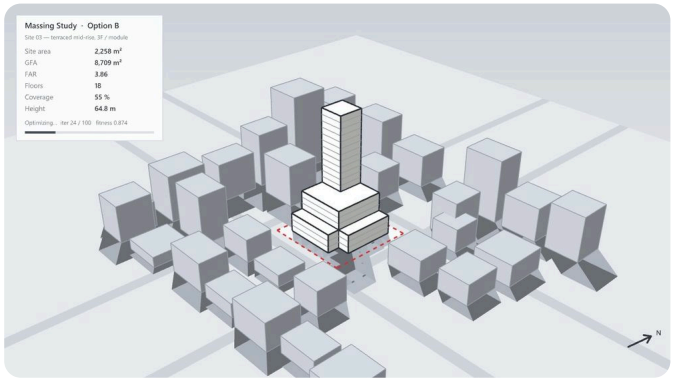
Voxel Unit
自定义积木：随手画的封闭体、体素中心点，或文本图案 `###/#.#|#..`（/ 分行，| 分层）。最多 10 格，每边 ≤ 4，必须面连通。



Unit Library
内置积木集（−1 只用自定义 / 0 入门 / 1 标准 8 块 / 2 困难）加上自定义积木。



自定义积木 · "Gate 门楼"第 1 级就会发。开局 2.4 秒落一格，每级快 8 %；前两级只发扁平的内置积木。



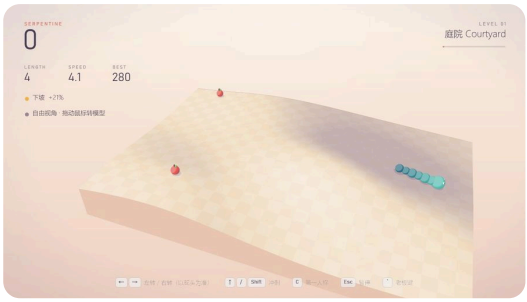
老板模式：体块研究 · 一格 = 7.2 m 柱跨 × 3 层，带场地红线、周边体块、日照阴影和容积率等指标。

Serpentine 蜿蜒 · 曲面上的贪吃蛇

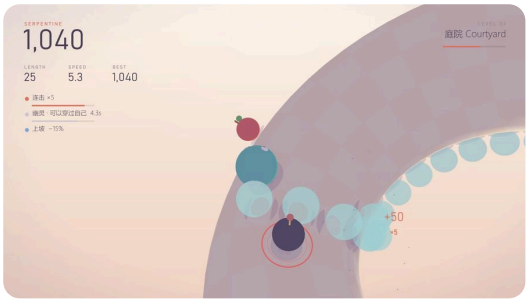
蛇沿曲面的 UV 网格一格一格走。曲面在某个方向闭合（圆管、环面、球）就能绕过去，开放的边是墙，修剪的洞是障碍；下坡快、上坡慢，曲面的形状就是关卡。



第一人称（默认） · 镜头锁在蛇头后上方，前方永远在屏幕上方。



自由视角 · 按 C 切换，看整个场地



环面 · 上下左右都能穿过去

← A · → D	以蛇头为准左转 / 右转（不管镜头朝哪）
按住 ↑ W Shift	冲刺 ×1.8
C	第一人称 ⇄ 自由视角（只有自由视角能拖动转模型）



Path Study
游戏本体。Surface（不接就用内置起伏地形，修剪面也行），Cells 长边格数（默认 28），Mode 0 经典 / 1 三分钟闪电 / 2 禅（撞墙停住），Level，Slope 坡度对速度的影响。

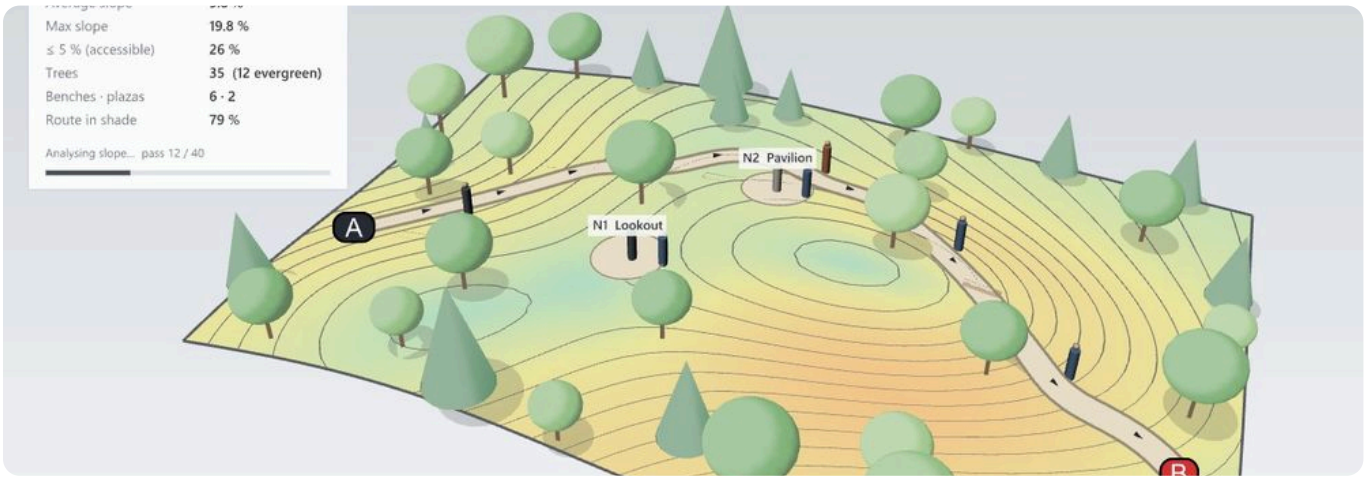
道具

- 苹果 +1 段
- 葡萄 +3 段
- 西瓜 +5 段，8 秒后消失
- 炸弹 −3 段，第 3 级起
- 冰块：6 秒减速
- 幽灵：6 秒穿过自己

4 秒内连吃有连击倍率（最高 ×5）。每吃 5 个水果升一级，五章配色：庭院、草坡、丘陵、峡谷、山脊。



Site Surface
一键生成场地：0 起伏地形、1 环面、2 圆管、3 马鞍屋面、4 球面；Size 默认 40 m，Seed 改变地形。



老板模式：景观设计 · 曲面画成坡度分析和等高线，蛇身变成铺装步道，配树、长椅、人和节点广场；路线长度、爬升、无障碍比例按蛇身实算。

Rebound 回弹 · 立面打砖块

砖墙按真实砌法排，一关一种。不同材料要打的次数不同；打碎的砖有机会掉道具，增益按次数用完就失效。



左边是正在生效的增益和剩余次数，挡板上方写着刚拿到的道具；两次碰挡板之间连续打砖有连击倍率（最高 ×8）。

材料

玻璃 1 下　砖 2 下　混凝土 3 下

钢：打不坏

道具（按次数失效）

加宽 · 8 次接球　多球　慢速 · 6 次

粘板 · 3 次　穿透 · 6 块砖　+1 命

缩短（坏）· 5 次

开局头六个道具正好是六种好道具各一个。

← → · A D · 鼠标　移动挡板

Space　发球、放开粘住的球

Facade Study

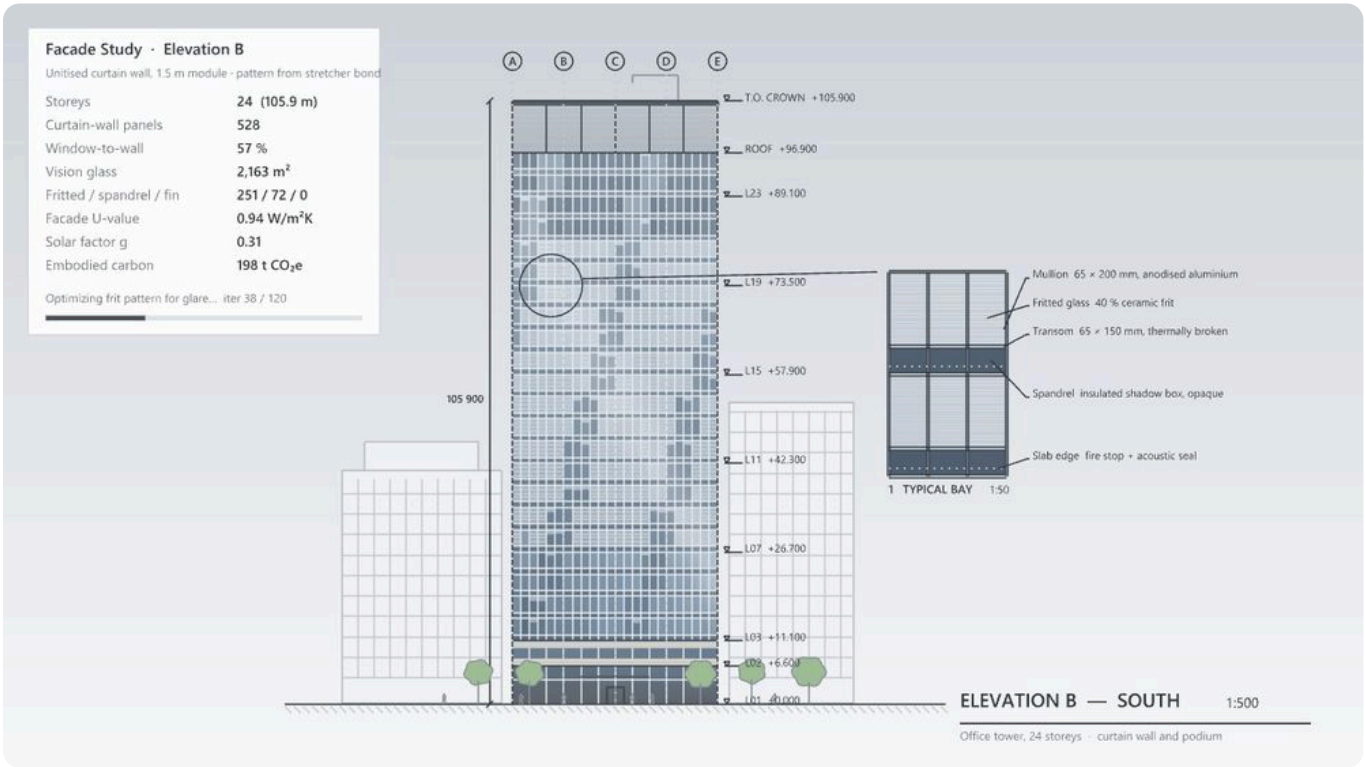
游戏本体。Wall（不接就六种砌法逐关轮换），Mode 0 经典 3 条命 / 1 三分钟闪电 / 2 禅（不掉命），Level，Plane（默认 World XZ），Size 模数（默认 2.4 m）。

Brick Bond

Type −1 轮换、0–5 指定砌法，Courses 皮数；或者 Curves 接闭合曲线当板块，按大小分成玻璃 / 砖 / 混凝土。



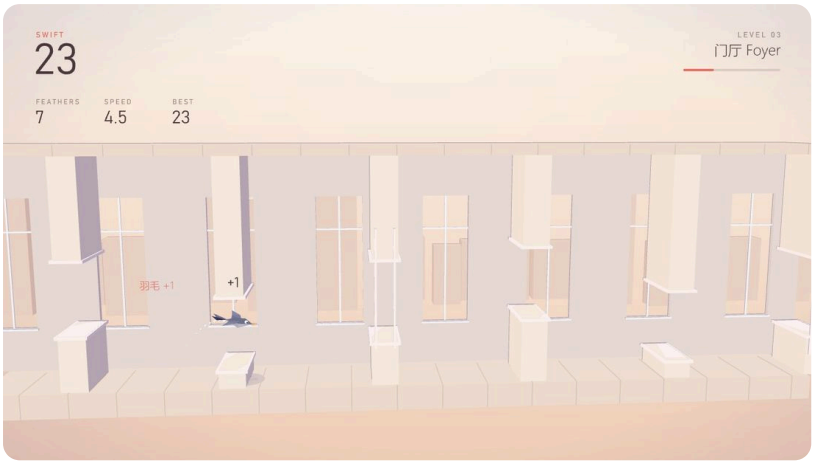
六种砌法 · 顺砌、通缝、英式、荷兰式、花墙、拱（图）。



老板模式：塔楼幕墙立面 · 每块幕墙板取它背后的砖：打掉的是透明玻璃，剩下的是印釉玻璃、窗槛板或铝板；窗墙比、U 值等指标按当前墙实算。

Swift 雨燕 · 剖面里的单键飞行

一只雨燕在一条无尽的建筑剖面里飞：按一下扇一次翅膀往上蹿，不按就往下掉。穿过每堵横墙的开口得 1 分，碰墙或落地就结束。



墙之间有羽毛可以捡。开头的开口很高、都在中间附近、速度慢；越往后开口越小、越快、高低差越大，第 26 堵墙起有上下滑动的窗扇。

Space / ↑ / W	扇一下翅膀（鼠标左键也行）
右键拖动	转视角

每 10 分升一层，每 3 层换一章配色：门厅、回廊、中庭、天井、黄昏。每个难度都用自动驾驶在五个种子上飞满 150 秒测过，关卡总能飞过去。



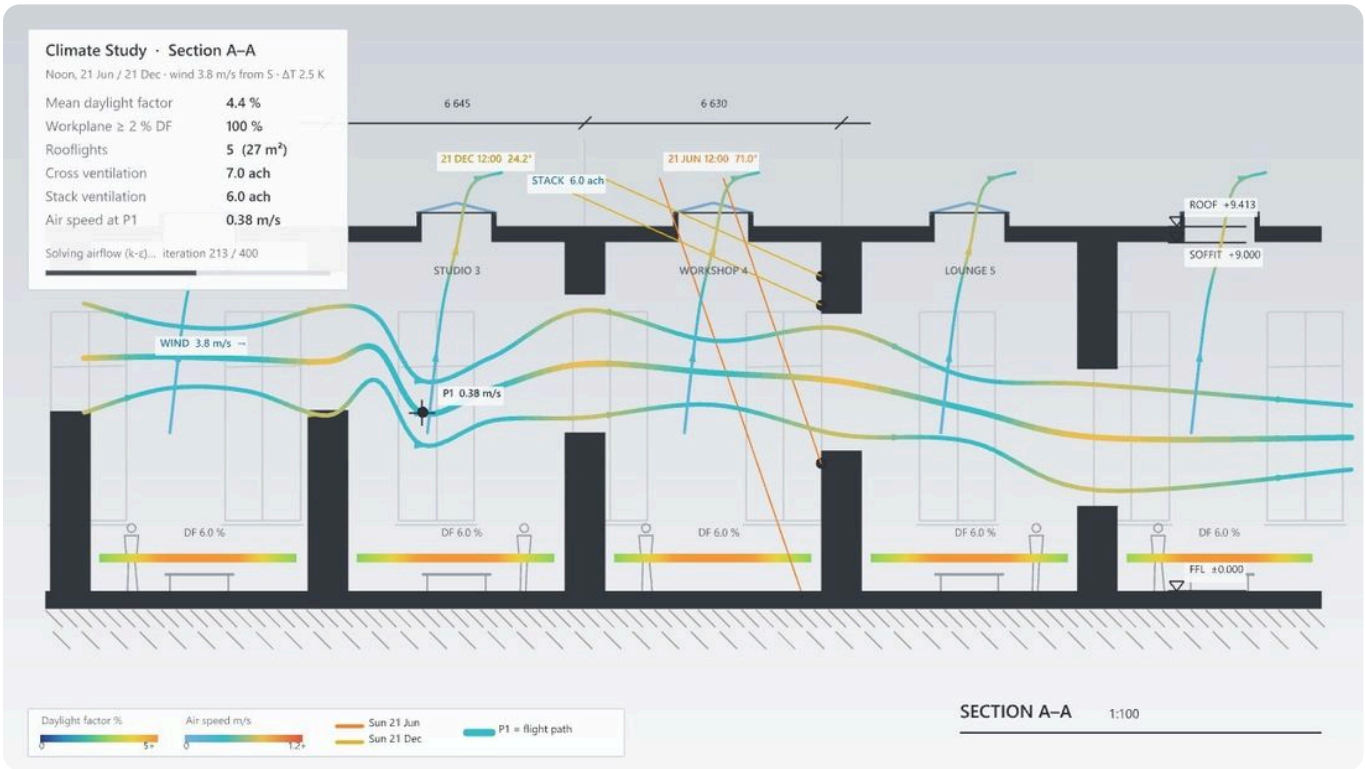
结算画面 · Enter / 空格再来一局，Esc 退出。

Airflow Study

游戏本体。Grid（Storey Grid 的输出，不接就是普通风压、经典模式），Plane（默认 World XZ），Size 场地单位长度（默认 0.75 m）。输出飞行路径、剖面轮廓和指标。

Storey Grid

Exposure 0 避风 / 1 普通 / 2 迎风（越大越快越窄），Mode 0 经典 / 1 Zen（撞墙穿过、落地弹起），Opening 开口高度 0.8–1.4，Bay 开间 0.8–1.5，Seed 固定关卡（0 = 每次随机）。最高分按这些设置分开记。



老板模式：采光与通风剖面 · 就是游戏里那个剖面：墙和开口在原来的位置，加上天窗、冬至 / 夏至的正午阳光、采光系数色带和气流线；主气流线就是雨燕刚才飞过的路线。

常见问题

游戏里按键没反应
焦点不在 Rhino 主窗口。点一下游戏视口，再按 Esc 继续。

游戏突然暂停了
对话框、Grasshopper 窗口或 Rhino 命令拿走了键盘。处理完点回视口、按 Esc。

点 ▶ Run 没反应
组件被禁用（锁定）了，或者组件上有报错气泡，原因写在气泡里。

求解器显示 Waiting
缺输入，通常是荷载或支座没接上。

求解器显示 Unstable
结构是机构，能自由移动或转动。加支座，或者让支座锁住两个方向。

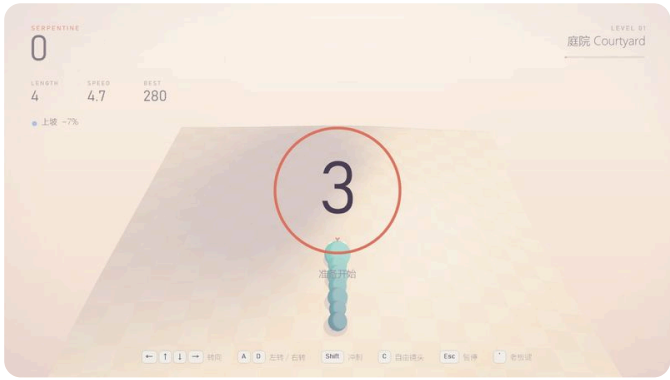
3D 每步很慢
体素太多。降低 Resolution 或过滤半径；超过 15 万体素每步要几秒，超过 100 万会警告（第 6 页）。

"Moyu data cannot be internalised"
右键 Internalise 存下来的 Moyu 数据无法还原。把产生它的组件重新接上。

"No Timer drives this solver"
文档里没有 Timer。接一个 Params › Util › Trigger 到求解器上，或者手动反复触发。



暂停画面 · 键盘被别的窗口拿走时也会停在这里；点回视口按 Esc 继续。



3-2-1 倒计时 · 从暂停或老板画面回来时先倒数，再继续。

文件都在哪

包管理器安装
`%APPDATA%\McNeel\Rhinoceros\packages\8.0\moyu\1.0.0\`
net48\、net7.0\（两份插件）、examples\、两份手册 PDF。

手动安装
`%APPDATA%\Grasshopper\Libraries\Moyu\`
你复制进去的那一份 Moyu.gha。

最高分
`%APPDATA%\Moyu\`
每个游戏一个纯文本文件，例如 swift-scores.txt。插件只写这里。

卸载

包管理器
`_PackageManager` › Installed › moyu › Uninstall

手动复制的
删掉 `%APPDATA%\Grasshopper\Libraries\Moyu\`

最高分
想清掉就删 `%APPDATA%\Moyu\`

关于。Moyu 1.0.0 · 作者 Ming · MIT 许可。代码、测试和文档主要由 Claude Code 编写，图标由 OpenAI Codex 生成。完整的组件输入、按键、方案文档和开发用时统计在 https://github.com/ChenmingHe0126/moyu ；问题反馈请开 GitHub issue。拓扑优化结果只用于学习和概念设计，不提供任何担保。