Model Arena

Opus 5.5 and Fable 5.2 (Claude Code) against Astra 6.2 and Sol 6.2 (Codex). Same prompt, same moment, high effort, one shot, no help. Built 18 Nov 2026, 18:08.

All rounds so far

Time and API-equivalent cost per round, then the totals.

MODEL	AAA PARKOUR	MADARIJ (3D FARMING)	BREATH (LUNGS)	TOTAL TIME	TOTAL COST	TOKENS OUT
Opus 5.5 Anthropic	55m 39s · $16.46	80m 43s · $22.47	60m 53s · $16.17	**3h 17m**	**$55.10**	858K
Fable 5.2 Anthropic	60m 13s · $30.85	76m 27s · $36.78	68m 26s · $30.34	**3h 25m**	**$97.98**	995K
Astra 6.2 OpenAI	17m 40s · $3.46	43m 17s · $5.92	17m 16s · $3.03	**1h 18m**	**$12.41**	120K
Sol 6.2 OpenAI	26m 45s · $1.58	35m 47s · $1.33	22m 15s · $1.05	**1h 24m**	**$3.96**	163K

Round: AAA parkour

Astra 6.2 finished first in 17m 40s. **Sol 6.2** cost the least at $1.58. All four builds load in Chrome with a clean console. Gameplay quality is scored live on stream.

THE PROMPT

Sent word for word to all four models in a fresh chat · 1,047 words

Build a first-person parkour game that feels like a AAA studio's vertical slice. It runs in the browser.

SCOREBOARD

MODEL	TIME	COST	TOKENS IN	CACHED	TOKENS OUT	API CALLS	TOOL CALLS	MESSAGES	SUBAGENTS	OWN CODE (LINES)	FOLDER	3D MODELS / TEXTURES / HDRI
Opus 5.5 Anthropic	55m 39s	$16.46	29.34M	97.6%	251.6K	140	202	50	none	4,477	29 MB	10 / 27 / 1
Fable 5.2 Anthropic	60m 13s	$30.85	24.67M	96.4%	300.8K	126	137	25	none	3,658	47 MB	17 / 139 / 1
Astra 6.2 OpenAI	17m 40s	$3.46	2.75M	95.5%	35.5K	49	48	5	none	2,630	23 MB	5 / 27 / 1
Sol 6.2 OpenAI	26m 45s	$1.58	6.66M	97.3%	57.3K	70	69	7	none	2,096	56 MB	9 / 15 / 1

Time to finish

Opus 5.5	55m 39s
Fable 5.2	60m 13s
Astra 6.2	17m 40s
Sol 6.2	26m 45s

API-equivalent cost

Opus 5.5	$16.46
Fable 5.2	$30.85
Astra 6.2	$3.46
Sol 6.2	$1.58

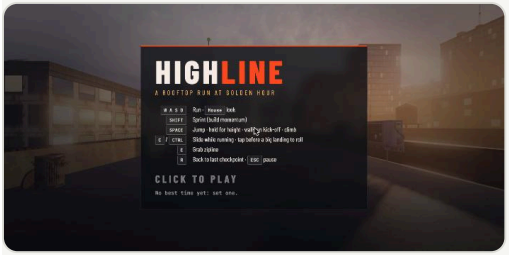
Output tokens (incl. thinking/reasoning)

Opus 5.5	252K
Fable 5.2	301K
Astra 6.2	36K
Sol 6.2	57K

Messages to the developer

Opus 5.5	50
Fable 5.2	25
Astra 6.2	5
Sol 6.2	7

FIRST SCREEN



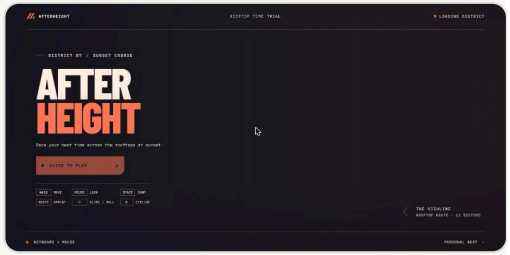
Opus 5.5 · **HIGHLINE**



Fable 5.2 · **LAST LIGHT**



Astra 6.2 · **Dusk Runner**



Sol 6.2 · **Afterheight**

HOW EACH MODEL TALKED TO THE DEVELOPER

Opus 5.5

50 messages · 202 tool calls

Narrated almost every step (50 messages): what it picked, what broke, how it fixed it. Found and fixed a bloom bug that blacked out the screen, then wrote a bot that drives the controls frame by frame and ran the whole course (50.3 s, no falls). Its final report flagged a miss on its own: the course is about 50 seconds for a good run, not the 2-3 minutes asked for.

Self-test: Bot ran the full course; ghost save and playback checked; clean console

Cost split: input $0.00 · cache writes $5.71 · cache reads $5.72 · output $5.03

Fable 5.2

25 messages · 137 tool calls

Silent for the first 3 m 49 s while it thought, then 25 calm progress notes. Built a headless test bot before any graphics and proved the course was beatable (62.8 s) before rendering it. Fixed a shader bug that put a black block on screen. Its final report has a 'What was not verified' list: mouse look under pointer lock, audio by ear, frame rate (it saw the Mac at a load average of about 56) and a human play-through. It noticed port 8000 was busy and left that server alone.

Self-test: Headless + in-browser autopilot finished the course (1:02.78); clean cold load

Cost split: input $0.00 · cache writes $14.06 · cache reads $4.76 · output $12.03

Astra 6.2

5 messages · 48 tool calls

Fastest and quietest: 5 messages. It used Codex's built-in image generator for the title-screen painting and said so in its credits. The final message reads like a launch post ('ready to play', a launch link) with a short list of simplifications.

Self-test: Automated run reached all 8 checkpoints (86.6 s); clean console

Cost split: input $0.99 · cache writes $0.00 · cache reads $1.05 · output $1.42

Sol 6.2

7 messages · 69 tool calls

7 short, factual messages. It was the only model that acted on the 2-3 minute target during the run: its first test took 73 s, so it lengthened the course (final run 1:44 with one deliberate fall). A Codex server restart interrupted it at 24:19; it picked up again by itself 3 seconds later.

Self-test: Automated run 1:44 with a deliberate fall; respawn, results and ghost checked; clean console

Cost split: input $0.36 · cache writes $0.00 · cache reads $0.65 · output $0.57

Round: Madarij (3D farming)

Sol 6.2 finished first in 35m 47s. **Sol 6.2** cost the least at $1.33. All four load in Chrome with a clean console on the sturdier local server. Sol's build failed to load on its own README server while the Mac was busy. Gameplay quality is scored live on stream.

THE PROMPT

> Kept private: this game may become a real product. All four models got the same prompt, word for word, in a fresh chat.

SCOREBOARD

MODEL	TIME	COST	TOKENS IN	CACHED	TOKENS OUT	API CALLS	TOOL CALLS	MESSAGES	SUBAGENTS	OWN CODE (LINES)	FOLDER	3D MODELS / TEXTURES / HDRI
Opus 5.5 Anthropic	80m 43s	$22.47	44.59M	98.3%	379.2K	205	328	91	none	7,190	11 MB	445 / 6 / 0
Fable 5.2 Anthropic	76m 27s	$36.78	31.12M	97.0%	391.2K	139	163	27	none	5,105	7 MB	153 / 5 / 0
Astra 6.2 OpenAI	43m 17s	$5.92	6.64M	98.1%	57.6K	84	82	5	none	4,402	19 MB	23 / 0 / 0
Sol 6.2 OpenAI	35m 47s	$1.33	5.16M	97.7%	59.0K	59	57	5	none	1,950	9 MB	47 / 0 / 0

Time to finish



API-equivalent cost



Output tokens (incl. thinking/reasoning)



Messages to the developer



FIRST SCREEN



Opus 5.5 · Madarij · مدارج



Fable 5.2 · Madarij · مدارج



Astra 6.2 · Madarij · مدارج



Sol 6.2 · Madarij · مدارج

HOW EACH MODEL TALKED TO THE DEVELOPER

Opus 5.5

91 messages · 328 tool calls

Cost split: input $0.00 · cache writes $6.12 · cache reads $8.76 · output $7.58

Fable 5.2

27 messages · 163 tool calls

Cost split: input $0.00 · cache writes $15.10 · cache reads $6.04 · output $15.65

Astra 6.2

5 messages · 82 tool calls

Cost split: input $1.01 · cache writes $0.00 · cache reads $2.61 · output $2.30

Sol 6.2

5 messages · 57 tool calls

Cost split: input $0.23 · cache writes $0.00 · cache reads $0.50 · output $0.59

Round: Breath (lungs)

Astra 6.2 finished first in 17m 16s. **Sol 6.2** cost the least at $1.05. All four load in Chrome with a clean console. Astra's default view is off-centre at a 1440×740 window. Gameplay quality is scored live on stream.

THE PROMPT

Sent word for word to all four models in a fresh chat · 737 words

Build an interactive, scientifically accurate 3D simulation of the human lungs and breathing. It runs in the browser and should feel like a premium museum exhibit.

SCOREBOARD

MODEL	TIME	COST	TOKENS IN	CACHED	TOKENS OUT	API CALLS	TOOL CALLS	MESSAGES	SUBAGENTS	OWN CODE (LINES)	FOLDER	3D MODELS / TEXTURES / HDRI
Opus 5.5 Anthropic	60m 53s	$16.17	33.95M	98.2%	227.6K	158	180	49	none	3,325	9 MB	3 / 0 / 1
Fable 5.2 Anthropic	68m 26s	$30.34	30.36M	97.5%	303.0K	134	148	20	none	3,107	16 MB	1 / 0 / 1
Astra 6.2 OpenAI	17m 16s	$3.03	3.29M	97.4%	26.9K	54	53	4	none	2,050	13 MB	1 / 0 / 0
Sol 6.2 OpenAI	22m 15s	$1.05	3.59M	96.7%	46.4K	39	38	5	none	1,584	4 MB	4 / 0 / 0

Time to finish



API-equivalent cost



Output tokens (incl. thinking/reasoning)



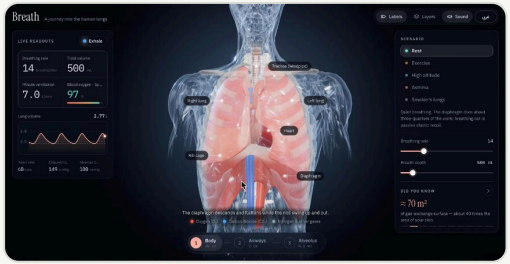
Messages to the developer



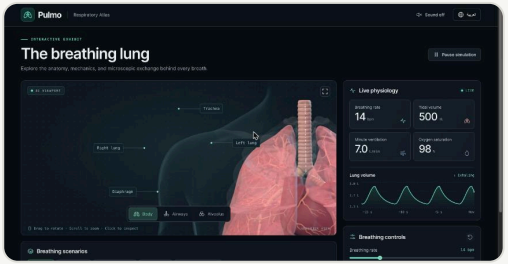
FIRST SCREEN



Opus 5.5 · **Breath**



Fable 5.2 · **Breath**



Astra 6.2 · **Pulmo — Respiratory Atlas**



Sol 6.2 · **Breath**

HOW EACH MODEL TALKED TO THE DEVELOPER

Opus 5.5

49 messages · 180 tool calls

49 messages, narrating each step and each bug. At 4 minutes it found that BodyParts3D has no outer lung surfaces, only the internal bronchial and vessel trees, so it built the lobe surfaces from that real anatomy itself, then checked the volumes (right 1.70 L, left 1.43 L, a 54/46 split). It tracked down a NaN bloom bug, a hanging Google Fonts request, and an intro animation that swallowed clicks, which it fixed for visitors too. A safety check held one command for about 2 minutes; Opus explained that the command deletes nothing and switched to the Edit tool. Its final message lists honest simplifications: 15 airway generations instead of 23, and the alveolus drawn 2× with a note on screen saying so.

Self-test: Final check on plain python3 -m http.server (port 8001): all 31 requests 200, clean console, air visibly streaming through the bronchiole.

Cost split: input $0.00 · cache writes $4.95 · cache reads $6.67 · output $4.55

Fable 5.2

20 messages · 148 tool calls

20 messages, one per phase, technical and dense. It used the real BodyParts3D lobes and skeletonized the real bronchial tree (306 tips), then grew the smaller airways from those tips into each lobe (21 of 23 generations drawn). It found and fixed a brief, unrealistic SpO_2 dip when switching scenarios. Like Opus in round 2, it hit the plain Python server dropping requests and changed its loader to fetch files one at a time with retries. It ended once more with a 'Not tested' list: phone layout, other browsers, the rebuild scripts. It also said it could not hear its own audio.

Self-test: Final pass on the README's exact command (port 8000): loads from the title card, no console messages, 40–60 fps. Opens on a title card; the screenshot is taken after Begin.

Cost split: input $0.00 · cache writes $12.30 · cache reads $5.92 · output $12.12

Astra 6.2

`4 messages · 53 tool calls`

4 messages, the fewest of any round. Its second message already called out a physiology detail: the altitude scenario lowers oxygen partial pressure while the oxygen fraction stays near 21%. The final message again opens with an 'Open Pulmo' link and is upfront that SpO_2 uses illustrative scenario values rather than a model.

Self-test: Checked all three views, five scenarios, sliders, sound, pause, fullscreen and the Arabic layout in Chrome; no console errors. Loads cleanly here, but in a 1440×740 window the default body view is off-centre: the lungs sit to the right and are cut off, and the 'Right lung' and 'Diaphragm' labels point at empty space.

Cost split: input $0.68 · cache writes $0.00 · cache reads $1.28 · output $1.08

Sol 6.2

`5 messages · 38 tool calls`

5 short, factual messages. It loaded Codex's 'Sites' skill at the start and said so. It was the only model to mention testing at a mobile viewport, and it added an in-browser credits panel. It describes its breathing and gas exchange as 'educational approximations'.

Self-test: Desktop and mobile checks in Chrome: models rendered, controls worked, no warnings or errors.

Cost split: input $0.24 · cache writes $0.00 · cache reads $0.35 · output $0.46

HOW IT WAS MEASURED

- **Time**: from the moment the prompt was sent until the agent went idle and stayed idle, logged every 15 seconds.
- **Tokens and calls** come from each tool's own session logs (Claude Code transcripts and Codex rollout files), not from estimates. Output tokens include thinking (Claude) and reasoning (Codex) tokens.
- **Cost** is what the same tokens would cost on the public API. The actual runs were billed to the Claude Max and ChatGPT subscriptions. Claude: Opus 5.5 $4 in / $20 out, Fable 5.2 $8 / $40, cache reads $0.20, 1-hour cache writes $8 / $16 (Claude Code writes 1-hour caches). OpenAI: Astra 6.2 $8 / $40 with $0.40 cached input; Sol 6.2 $2 / $10 with $0.10 cached input. The OpenAI rates come from launch coverage that cites OpenAI; they were not checked against OpenAI's own pricing page.
- **Same machine, same time.** All four ran together on one Mac and shared its CPU and browser, which slows the longer runs more. Fable measured a load average around 56 in round 1; it reached 84 in round 2.
- **Serving the builds.** Every build is judged on the same sturdier local Python server (a threaded server with a deep connection queue). Under that load the plain `python3 -m http.server` drops some requests; in round 2 that blanked Sol's world on its own README command, while Opus had hit the same thing and added retries.
- **Fresh chats.** Each round starts in a new chat. In Codex, a new chat resets the model to the default (Terra 6.1), so Astra and Sol were set back to their model on high effort and checked in the pane footer before every prompt.
- **Subagents.** Checked from the logs: Claude Code's Agent/Task/Workflow tools and subagent log files; Codex's spawn_agent and related tools (called directly or inside its exec scripts) and any new thread in the model's folder; and any shell command that starts another model or calls a model API. "Quiet" means the model sent no message to the developer in the 3 minutes before the spawn.
- **Hooks.** The Claude Code panes run with this Mac's plugins, including a design checker that comments after HTML/CSS is written; Opus and Fable both reacted to it at the end of round 2. Codex has no equivalent hook here. Claude Code's safety check also held one Fable cleanup command for 2 minutes in round 2.
- **Tooling differs.** Codex has a built-in image generator (Astra used it for its title art); Claude Code doesn't. A Codex server restart interrupted Sol for about 3 seconds.
- **Own code** counts the model's own .js/.html/.css lines and leaves out vendored libraries such as three.js.