



kiosa

@thegreatest_sv

PART 1 THE VIDEO FROM THE POST

HOW I MADE THIS VIDEO

The exact prompt behind the looping 3D clip in my post. Paste it into Claude and you get the same video: one HTML file, rendered frame by frame, with a synthesized score.

1080 × 1350

18 s

60 fps

three.js

seamless loop

Web Audio score

PART 1 · PAGES 1-3

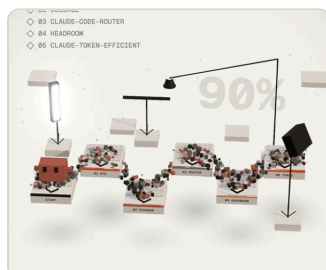
The video prompt

Visual prompt + sound prompt, copy-paste ready.

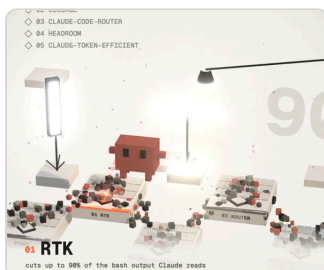
PART 2 · PAGE 4

Bonus: Claude Code, cheaper

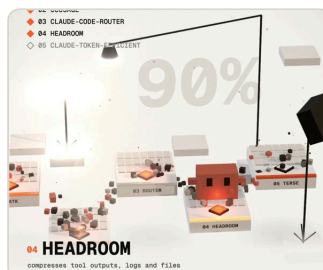
The 5 repos the video is about, with install commands.



00:00 · lights click on



00:04 · first hop



00:09 · stream thins out



00:14 · final flip lands

1

Paste the visual prompt into Claude and ask for one HTML file.

2

Paste the sound prompt and ask it to mux the score into the MP4.

3

To retheme: swap the tile labels, checklist, captions and terminal commands.

1A VISUAL PROMPT

PASTE AS IS

Build a 3D loop: a voxel mascot hops across floating tiles, one tile per repo that makes Claude Code cheaper. One HTML file, three.js, Geist Mono. 1080x1350, 18 s at 60 fps (1080 frames), frame 1080 equals frame 0. window.renderAt(t) draws any frame from t alone; seeded random only.

Set: a miniature studio on an $\square$ F4F2EE backdrop. Six floating white tiles, each $2.1 \times 0.5 \times 2.5$, at $x = -6, -3.6, -1.2, 1.2, 3.6, 6$, heights $0, 0.7, -0.35, 0.95, 0.15, 1.25$ and depths $0.9, -0.7, 0.8, -0.9, 0.6, -0.5$, so the path zigzags up, down and sideways. Every tile bobs ± 0.09 twice per loop, each on its own phase, and has a soft shadow blob 3.2 below it. On top: 21 cells (7×3 , $\square$ EEEEEA) and one keyframe diamond (white fill, $\blacksquare$ 0F0F0F outline). On the front face: a label in Geist Mono 700 (START, 01 RTK, 02 CCUSAGE, 03 ROUTER, 04 HEADROOM, 05 TERSE) and an accent strip under the top edge.

Lamps, each on its own small tile: a key softbox front right, a rim strip light back left with its panel turned to camera, a top spot at 5.8 on a boom from the back right, and a fourth lamp: a horizontal 2.6 tube light on a stand behind the path at $(-1.6, 2.35, -4.6)$.

Background: four plain white tiles, $1.4 \times 0.3 \times 1.4$, floating behind the path at depths -6 to -7.5 and bobbing ± 0.12 . A flat 7×3.5 sign reading 90% in Geist Mono 700 stands at $(3.6, 2, -6)$ in $\square$ DDD9D1, barely darker than the backdrop.

Hero: the pixel Claude mascot as 0.1 voxels: $\blacksquare$ D97757 body $14 \times 10 \times 8$, two 3×2 arms, four 2×3 legs, $\blacksquare$ 0F0F0F 2×3 eyes; roughness 0.45 , clearcoat 0.3 .

Lights, frames 6, 24, 42, 60: key 3200 K with soft shadows, rim 7500 K, top spot 5000 K, back tube 4200 K. Each clicks on with a 3-frame flicker and the mascot's eyes glance at it. Lights-off sits near 50% .

Token stream: 560 voxels, 0.1 to 0.2 , dark greys with some $\blacksquare$ D97757, flow right to left across all six tiles and over the gaps, 3 passes per loop. This is the output Claude Code reads. After each landing it thins: $100, 56, 44, 34, 26, 20\%$. Removed cubes lift 0.9 and shrink away.

Hops: 5 hops, tile to tile, landing on frames $150, 270, 390, 510, 630$. Each hop: 11-frame crouch (1.2×0.75), stretch 0.85×1.28 at takeoff, 25 frames of air with a 1.7 apex, squash 1.25×0.7 on landing, spring back at 3.4 Hz, damping 0.55 ; arms lag 4 frames, legs 6, arms settle at 2.2 Hz, damping 0.22 . 54 frames after each of the first four landings, an 18-frame mini hop 0.6 high on the spot.

Each landing: the tile dips 0.34 , tilts and springs back at 1.7 Hz; its strip and diamond light $\blacksquare$ FF6A2B; an $\blacksquare$ FF6A2B ring expands across the top; the cells ripple 0.16 high; 48 voxel sparks (0.075 to 0.12) fly; nearby stream cubes are kicked 1.1 up; the 90% sign flashes $\blacksquare$ FF6A2B at 30% and fades.

Flip, frames 700 to 812: a dotted arc of 16 dots draws in from the last tile back to START, one dot per frame from 700, starting above the mascot's head. An 18-frame deep crouch (1.3×0.62) from 722, takeoff on 740, a 720° flip along the arc, 4.3 high, landing on frame 812 on START, with the biggest ring, sparks and squash (1.32×0.64). On that landing the 90% sign turns fully $\blacksquare$ FF6A2B and stays lit.

Light effects, all subtle: a trail of 10 shrinking $\blacksquare$ FF6A2B cubes behind the mascot whenever it is in the air; a faint warm cone of light under the top spot; 90 tiny $\blacksquare$ D97757 specks (0.03 to 0.07) rising slowly through the whole set, one or two passes per loop.

End: hold, blink on frame 960; on frame 1056 the lights, strips, diamonds, sign and labels snap off and the stream refills to 100% over 24 frames.

Camera: FOV 18 , 30° elevation, $\pm 20^\circ$ orbit over the loop plus a $\pm 7^\circ$ faster sway, 66 units back. A 34% push-in from frame 90 to 690 tracks the mascot at 55% , including its height. 5% punch-in and a small roll shake on every landing, 7% on the flip landing; a 7° crane up and 8% pull back during the flip.

Post: bloom on emissives only, 2% grain, 10% vignette, dithered backdrop. The lit front face must read $\blacksquare$ D97757 and lit diamonds $\blacksquare$ FF6A2B.

Overlay, 2D, Geist Mono:

Top left: title MAKE CLAUDE CODE 90% CHEAPER (33 px, ■ 0F0F0F, with 90% in ■ FF6A2B), under it "5 open-source repos that shrink the bill" (18 px, ■ 8E8A83). Top right: handle "kiosa · @thegreatest_sv" in ■ D97757, then a TOKEN LOAD meter labelled "illustrative": a 300 px bar and a 44 px percentage that follow the stream.

Checklist under the title, 23 px, ■ 8E8A83: 01 RTK, 02 CCUSAGE, 03 CLAUDE-CODE-ROUTER, 04 HEADROOM, 05 CLAUDE-TOKEN-EFFICIENT. Each row turns ■ 0F0F0F and fills its ■ FF6A2B diamond on its landing frame.

Caption above the terminal after each landing: the number in ■ FF6A2B, the repo name at 62 px (46 px for long names), and one typed line: "cuts up to 90% of the bash output Claude reads", "shows where every token goes – measure first", "sends background work to cheaper models", "compresses tool outputs, logs and files", "one CLAUDE.md that keeps replies terse". After the flip: SAME CLAUDE. in ■ 0F0F0F and SMALLER BILL. in ■ FF6A2B, 56 px.

Terminal, bottom, 1000 x 200, ■ 0F0F0F with rounded corners: on the left, raw command output (git status, test results, compiler and npm lines) scrolls fast; as the load drops, more lines are greyed and struck through in ■ FF6A2B. On the right, the active tool's command types out with a caret, then two lines of result: "\$ brew install rtk", "\$ npx ccusage@latest daily", "\$ ccr code", "\$ headroom wrap claude", "\$ cat CLAUDE.md".

Keep every prop, the sign and the background tiles out of the title and checklist block, and keep captions readable over the scene with a soft □ F4F2EE gradient behind them.

1B SOUND PROMPT

PASTE AFTER 1A

Score this 18 s loop, 1080 frames at 60 fps. Synthesize everything in code with the Web Audio API, no samples, no downloads. 120 BPM, 9 bars, F major. Each hit lands 1 frame after its visual event.

Bed: pad voices A3 E4 (key light), C5 (top), G5 (rim), each following its light's flicker. Shaker 8ths from 45, kick and bass from 60, roots F D Bb F F D Bb C F by bar; kick and shaker rest 740 to 870, bass holds C from 740, then F at 812.

Clicks: lights 6, 24, 42, 60, off 1056.

Crouches 114, 234, 354, 474, 594, 722: creak, inhale riser.

Takeoffs 125, 245, 365, 485, 605, 740, and mini hops 204, 324, 444, 564: upward swish.

Landings 150, 270, 390, 510, 630, 812: rubber squash pitched by the squash scale, spark tings, 812 biggest.

Checklist pings A5 C6 D6 F6 A6: 150, 270, 390, 510, 630.

Terminal typing ticks for 44 frames from 8, 154, 274, 394, 514, 634, 816.

Dot ticks 700 to 715, flip whoosh 740 to 812, blink blip 960.

48 kHz stereo WAV, about -14 LUFS, peak -1 dBTP, seamless loop, mux into the MP4.

PART 2 BONUS · WHAT THE VIDEO IS ABOUT

MAKE CLAUDE CODE CHEAPER

Five open-source repos, each attacking the bill from a different side. Full breakdown is in the article this video quotes.

90%

01 **rtk** github.com/rtk-ai/rtk

Filters and compresses command output before Claude Code reads it. Its README: up to 90% of bash output cut – that is output, not the whole bill.

```
$ brew install rtk
```

02 **ccusage** github.com/ryoppippi/ccusage

Reads your local session logs and shows where tokens go – by day, session and project. Measure first, then cut.

```
$ npx ccusage@latest daily
```

03 **claude-code-router** github.com/musistudio/claude-code-router

A proxy that sends background work (summaries, compaction) to cheaper models and keeps the hard work on Claude.

```
$ ccr code
```

04 **headroom** github.com/headroomlabs-ai/headroom

Compresses tool outputs, logs and file reads on the way in; originals stay on your machine and can be pulled back.

```
$ headroom wrap claude
```

05 **claude-token-efficient** github.com/drona23/claude-token-efficient

One drop-in CLAUDE.md that keeps replies terse. Keep the file short – it is sent on every turn.

```
$ cat CLAUDE.md
```

Honest numbers – savings figures are each project's own claims and depend on your workflow. "Up to 90%" is rtk's figure for bash output, not for your total bill. Check each README for current install steps before running anything.

**kiosa** @thegreatest_sv

bookmark · repost · follow for more AI tools