AI Pilot Report: Agentic Coding Tools, Costs, Usage & Playbook

November 22, 2026 · Sam Posten

To: Engineering Manager. Over seven weeks I used two AI coding agents on two machines to ship 22 HDR screensavers for both Windows and Mac, two games now in Apple TestFlight for iPhone and Mac, a web/Windows/Mac/iOS arcade game, and a restored and redesigned website. The total cost is about $76/month.

About this version. This merges three reports: Claude Code on the Windows laptop, Google Antigravity on the Mac, and Claude Code on the same Mac. The Mac Claude Code work came before and after the Antigravity session: setup and porting from Nov 16 to Nov 21, then shipping from Nov 22 at noon, four minutes after Antigravity pushed Deep Void. Mac Claude Code figures come from its local session logs and the live plan meter, read at about 6:45 PM ET on Nov 22.

1. Summary

I acted as product owner and QA tester, and the agents wrote the code. I gave short plain-English prompts, tested on real hardware, and pushed back when results were wrong.

Agent	Machine	Period	Headline output
Claude Code (Claude desktop app, Code tab)	Windows 11 laptop	Oct 2 – Nov 22 42 sessions, 25 active days	22 HDR10 Windows screensavers with installers (17 of them on Nov 21 alone), the restored website (first version), 3 web games, an HDR video pipeline YouTube accepts, local AI images, CAD, scheduled monitoring agents
Claude Code (Claude desktop app, Code tab)	Mac (M5 Max)	Nov 16–22 17 sessions, 5 active days, about 7.5 h active	Mac setup; Mac ports of all 22 screensavers (signed and notarized); Gravwell and Deep Void into TestFlight for iPhone and Mac; Deep Void downloads for Windows and Mac; took over navesink.net with SSH deploys and a redesign
Google Antigravity 2.0	Mac (same machine)	Nov 21–22 about 3.5 h active	Deep Void: a 3D arcade game from one codebase for macOS, iOS, Web and Windows, with CI

Main findings

- **Agents can take a project from idea to shipped build.** On the Mac that went as far as Apple Developer ID signing, notarization and TestFlight uploads, all scripted. I still made the product decisions, tested on hardware, and did anything that needed my identity: sign-ins, Apple's terms, downloading secret keys.
- **Saved lessons make each new project faster.** Each agent saved what it learned after every project (Claude Code: memory files; Antigravity: a global plugin). The first screensaver took several sessions. The Mac port of SpookySaver took a day, the next four took one session, and the last 17 were ported, signed and published in about an hour, using a translator script the agent wrote.
- **Agents can hand work to each other through git.** Windows wrote the screensavers, the Mac ported them, Antigravity built Deep Void, and Mac Claude Code shipped it. That only worked once one machine owned the website (see Lessons).
- **The cost is a few subscriptions.** About $76/month all-in, $68 of it for AI subscriptions. Both Claude Code machines run on one Claude Pro seat, so the Mac added no cost. No agent

charged an overage. The constraint was the plan's usage limit: Claude Code hit its 5-hour limit once on Nov 22 and resumed after the reset.

2. How the work moved between machines

Read top to bottom (oldest first). Mac Claude Code rows are shaded.

When	Agent and machine	Hand-off
Oct 2 – Nov 16	Claude Code, Windows	Builds the games, restores navesink.net, ships SpookySaver for Windows (Nov 16), copies its sessions and memory to the new Mac
Nov 16–18	Claude Code, Mac	Sets up the Mac, ports SpookySaver to Mac with Developer ID signing, takes over the blog from update 20 (SSH deploys), redesigns the site
Nov 20	Claude Code, Windows	Builds 4 holiday savers and pushes them; also uploads straight to the site, overwriting two Mac pages
Nov 21, 12–1 AM	Claude Code, Mac	Ports the 4 holiday savers, repairs the overwritten pages, puts Gravwell on the iPhone Duo
Nov 21, daytime	Claude Code, Windows	Builds 17 more savers (10 parallel sessions) and pushes them to GitHub
Nov 21, 8–9 PM	Claude Code, Mac	Ports all 17, builds their Windows installers on the Mac, publishes a 22-saver hub
Nov 21, 8:31 PM – Nov 22, 11:59 AM	Antigravity, Mac	Builds Deep Void and pushes sposten/DeepVoid
Nov 22, 12:03 PM onward	Claude Code, Mac	Ships Deep Void (TestFlight, notarized Mac download, Windows exe, website), then Gravwell to TestFlight, new Gravwell features, this report

3. Claude Code work log: Windows

Newest first. Each row is what I asked for and what the agent delivered.

Dates (2026)	Project	Delivered
Nov 22	This report (first draft)	Merged the Antigravity report with the Claude Code logs, the plan meter and vendor balances
Nov 21	Code signing research	Looked into Microsoft code-signing options to stop SmartScreen and Defender false positives
Nov 20–21	17 more themed screensavers	Shamrock, Easter, Sci-Fi, Dino, Aurora, New Year's, Día de los Muertos, Hanukkah, Winter, Lunar New Year, Mardi Gras, Ocean, Firefly, Synthwave, Sakura Zen, Dragons, Steampunk. Each was checked for a 1,995-nit peak and its frame times, scanned with Defender, given an icon and SHA-256, and pushed to its own private GitHub repo. Nov 21 alone was 798 model calls across 10 parallel sessions
Nov 20	Love Burst, Harvest, Jolly Christmas, Star Spangled	Turned the engine into a reusable template: Blender builds the models (headless) and the agent writes each theme's shaders and three animated scenes. Saved a themed-screensaver playbook to memory
Nov 16–17	SpookySaver (first screensaver)	C# with OpenGL 3.3 and HDR10 output (D3D11/DXGI), plus an installer and uninstaller. Published with a download page on navesink.net. Copied my sessions to the new Mac

Dates (2026)	Project	Delivered
Nov 13–16	navesink.net restoration	Rebuilt my lost blog as a static site from archives (Python generator). Published to GoDaddy cPanel through Claude in Chrome: I signed in, and the agent uploaded and extracted the update zips. Handed to the Mac at update 19
Nov 10–15	Web games	Finger Drums and Drum Trainer, Liar's Dice (multiplayer), Gravwell (later given an AI opponent). Each has its own private repo
Nov 11	Scheduled agents	Recurring price-monitor tasks that check retail listings and report back
Nov 10–11	HDR video for YouTube	A 4K60 HDR10 video that YouTube treated as SDR. Diagnosed missing container metadata (colr, mdcv, clli), built an injector script, then settled on a ProRes MOV master that YouTube confirmed as HDR
Nov 9–10	CAD and ArtCraft apps	Installed SolidCraft and other ArtCraft MCP apps; designed a parametric throwing-star part
Nov 1–9	Local AI image generation	Installed MOSS-Image (stable-diffusion.cpp, CUDA) on the laptop's RTX 5080: about 4 min per 1360×768 image, $0 per image. Made the RYE logo and other images
Nov 1	Meshy MCP setup	Connected the Meshy 3D-generation API to Claude Code (local Node process with an API key)
Oct 28–31	Trading tool, HDR demo	A trading-account management tool; more 4K HDR demo-scene tooling
Oct 8–12	Utility apps	A Path of Exile 2 stash value finder and the Gods Eye View project
Oct 2–6	Tool integrations, Blender	Connected Suno and evaluated Midjourney. Drove Blender for a koala animation and a 4K HDR demo scene. Made a challenge-coin graphic

4. Claude Code work log: Mac

Newest first. The Mac became the release machine: it is the only one that signs Apple builds and the only one that deploys navesink.net.

Dates (2026, ET)	Project	Delivered
Nov 22, evening	This report (third version)	Read the Windows/Antigravity draft and added the Mac Claude Code work from its session logs, the live plan meter and the Suno balance
Nov 22, 1–5 PM	Gravwell: new features	Duel shield: trap the AI so its shot flies back over its own line and you get +50 and a one-shot gold shield. The AI always takes its best shot but starts loose and sharpens over about 30 shots. Pushed, live on the site and uploaded as TestFlight build 2. A daily scheduled task submits each new build to Apple's beta review once the one ahead of it clears. Overdrive option (in progress): a checkbox for extra particles, HDR colour flashes, sub-bass and announcer voices made with Suno through its MCP. Gameplay was verified identical with it on or off. It is on its own branch with a private, unindexed preview page, not merged yet
Nov 22, 12:38–1 PM	Gravwell to TestFlight	Made the iPhone app a single target that also builds a native, sandboxed Mac app. Created the App Store Connect record, uploaded build 1 for iPhone and Mac, set up public and internal beta groups, submitted to beta review, and posted a /gravwellinstall page and announcement

Dates (2026, ET)	Project	Delivered
Nov 22, 12:03–12:37 PM	Deep Void: shipping (after Antigravity)	Picked up Antigravity's repo and fixed the web rocket orientation, lighting, worldwide score submission from the native apps, iPhone Silent-mode sound and Mac sandbox entitlements. Built the Windows installer on the Mac and a notarized universal Mac download. Created the App Store Connect app, made an Admin API key (I approved it and clicked Download), created the distribution certificates, uploaded TestFlight builds for iPhone and Mac and submitted them for beta review. Published /deepvoid (playable web game with a PHP leaderboard on my host), /deepvoidinstall and a launch post
Nov 21, 8–9 PM	17 Mac screensavers + collection hub	Cloned the 17 new repos and ported every one to Swift/Metal with HDR10 at 1,600 nits. The agent wrote a C#-to-C++ translator for the animation code; it matched the hand ports to 0.0002. Signed and notarized each one (about 1 minute each), built the Windows installers on the Mac (Roslyn against .NET Framework 4.8) and pushed all 17. Published a /screensavers/ hub for all 22 and a blog post
Nov 21, 12–1 AM	Holiday savers, site repair, Gravwell on iPhone Duo	Ported the 4 holiday savers to Mac through one shared engine and a GLSL-to-Metal script, then published them. Found that a PC upload had overwritten two Mac pages; restored them and merged a diverged Gravwell (3-way, no conflicts). Installed Gravwell on my new iPhone Duo (folding phone) and fixed silent audio in Silent mode
Nov 17–18	navesink.net takeover and redesign	Made the site build incremental: each update zip holds only the changed files, and the sidebar became one shared script, so a new post touches about 7 files instead of 600. GoDaddy blocks the built-in browser, so the agent set up SSH key deploys (I enabled SSH and ran one command). Deploys now back up, unzip, check every file's SHA-256 and record what's live, with no browser. Designed six redesign mockups; I picked "Riverfront" and it went live with thumbnails for every post (app icons or public-domain aerials). Updates 20 to 31 have shipped this way
Nov 17	SpookySaver for Mac	Ported the Windows saver to a Swift/Metal .saver with HDR10 metadata at 1,600 nits for the M5 Max XDR display. Fixed macOS's dead Options-sheet buttons. Set up Developer ID signing and notarization: the Xcode route failed, so the agent made the CSR with openssl and used the developer site in the built-in browser while I signed in, then an App Store Connect API key for notarization. Published the Mac download and a post
Nov 16–17	Mac setup, tools, Gravwell for iPhone	Installed Homebrew, Xcode tools, MOSS-Image (local images on Apple Silicon, about 30–60 s each), Blender, uv, and the Craft suite with its MCP servers; connected Xcode, GitHub and Unreal Editor. Tests: a 30 s HDR10 dancing-robot Blender animation and a 15 cm scaled dragon egg in SolidCraft (STEP/STL). Turned the Gravwell web game into an iPhone app: iPhone Duo fold handling, a worldwide scoreboard, fonts, an icon, and an install on my phone

5. Antigravity work log: Mac (Deep Void)

Five phases took about 3.5 hours of active time over Nov 21–22 and ended in the private repo sposten/DeepVoid. Mac Claude Code then took it to TestFlight and the website (section 4).

Phase	Time (ET)	Outcome
5. Ship and save lessons	Nov 22, 11:33–11:59 AM	Pushed the private repo; wrote a global cross-platform-shipping plugin
4. Cross-platform	Nov 22, 10:30–11:28 AM	Added 5 power-ups and a Classic mode. Built a Web version (Vite, Three.js) with a worldwide leaderboard, a Windows version (Electron) and GitHub Actions CI. Removed the trademarked word from 60 files

Phase	Time (ET)	Outcome
3. Polish	Nov 21, 11:21–11:55 PM	Made the app icon, renamed the game Deep Void, added a UFO and hero rocket from Meshy, added macOS gamepad support
2. iPhone Duo bring-up	Nov 21, 10:46–11:15 PM	Patched signing directly in project.pbxproj. Fixed a SceneKit deadlock and a SwiftUI crash. Reduced the meshes in Blender (load time down to 0.49 s), disabled HDR auto-exposure, moved controls to where the thumbs rest
1. Native app and assets	Nov 21, 8:31–10:44 PM	Built a SwiftUI/SceneKit app for macOS and iOS. The browser subagent generated 5 PBR rock models in Meshy and a chiptune track in Suno

6. Tools and costs

AI subscriptions total about $68/month: Claude Pro $20, Google AI Pro $19.99, Meshy Pro $20 and Suno Pro about $8 on the annual plan. Apple's $99/year brings it to about $76/month. One Claude Pro seat covers both Claude Code machines. A single-platform setup costs $20–$48/month.

"Used by": Win = Claude Code on Windows, Mac CC = Claude Code on the Mac, AG = Antigravity on the Mac.

Tool or service	Used by	Role	Plan and price	Actual usage
Claude Code (Opus 5.5 main; Fable 5.2, Sonnet 5.5 on Windows)	Win, Mac CC	Coding agent: shell, files, browsers, MCP tools, parallel sessions, scheduled tasks, memory	Claude Pro, about $20/mo (one seat, both machines)	Win: 3,534 model calls, 42 sessions. Mac: 1,463 calls, 17 sessions, all Opus 5.5. Extra usage is off, so no overages
Google Antigravity 2.0 (Gemini 4 Neon)	AG	Coding agent, browser subagent, image generator	Google AI Pro, $19.99/mo	457 model calls, no throttling
Claude in Chrome and the built-in browser	Win, Mac CC	Web automation: cPanel deploys, YouTube HDR checks, Apple developer site, site previews	Included	Win: 728 Chrome + 453 built-in browser actions. Mac: about 270 built-in browser actions (Chrome is not installed)
Computer use	Mac CC	Reading my signed-in Safari (read-only) for private repo names	Included	About 30 actions
iOS Simulator and Xcode tools	Mac CC	Building and testing the iPhone apps on the iPhone Duo simulator	Included	165 simulator actions
Meshy AI	Win, AG	AI 3D models	Pro, $20/mo (1,000 credits/mo)	Balance 2,813 (493 monthly + 2,320 rollover; resets Dec 1). Deep Void used 337. The screensavers used free procedural Blender models
Suno AI	Win, AG, Mac CC	AI music and voice	Pro annual, about $8/mo (2,500 credits/mo)	60 of 2,500 monthly credits (2.4%) and 5 of 20 downloads used. Mac CC added the Gravwell announcer voices
Blender 5.2 / 5.3	All	Procedural models, mesh reduction, renders, icons	$0	All screensaver models, Deep Void assets, Mac test animation

Tool or service	Used by	Role	Plan and price	Actual usage
MOSS-Image (local)	Win, Mac CC	Text-to-image on the local GPU	$0 (open weights)	Logos and images, no per-image cost. Mac: about 30–60 s per image
ArtCraft / Craft suite MCP apps	Win, Mac CC	CAD, PDF editing	Local apps (cost not tracked)	Win: 56 SolidCraft and 28 PDFCraft calls. Mac: dragon-egg CAD test
.NET Framework csc / Roslyn, ffmpeg, Python	Win, Mac CC	Screensaver and installer builds, HDR video	$0	All Windows builds; the Mac builds Windows installers too
Xcode, Apple Developer Program, App Store Connect API	AG, Mac CC	macOS and iOS builds, Developer ID signing, notarization, TestFlight	$99/yr	23 notarized Mac downloads (22 savers + Deep Void); 2 apps in TestFlight for iPhone and Mac
Node, Vite, Three.js, Electron (+ makensis)	AG, Mac CC	Web and Windows versions of Deep Void	$0	Web and Windows builds
GitHub, gh CLI, Actions	All	Private repos, CI	Free tier	26 new private repos (25 Windows, 1 AG), 1 CI workflow. Mac CC pushed Mac ports to 22 saver repos plus Gravwell and Deep Void
GoDaddy hosting	Win, Mac CC	navesink.net	Existing plan	Win: cPanel uploads through Chrome (I signed in). Mac: SSH deploys with no browser, updates 20–31

7. Token usage and allotment

Both Claude Code machines draw on one Claude Pro meter. With 4 days left in the week it stands at 76%, and the 5-hour limit was hit once on Nov 22. Antigravity reported no throttling.

Claude Code plan meter (Claude Pro, shared by both machines)

Limit	Used, ~6 PM ET (Windows reading)	Used, ~6:45 PM ET (Mac reading)	Resets
5-hour window	28%	44%	About 10 PM ET tonight
Weekly, all models	74%	76%	Nov 26
Monthly	No monthly cap on Pro	No monthly cap on Pro	Not applicable
Extra (paid) usage	Off, $0 overage	Off, $0 overage	Not applicable

One Pro seat carried two machines, but it is close to full. Nov 21 alone was about 1,100 Claude Code calls across both machines. If one person regularly runs two machines or parallel sessions, the Max tier is worth pricing.

Claude Code tokens: Windows (from local session logs)

Window	Model calls	Uncached input	Cache writes	Cache reads	Output
Today (Nov 22)	8	0.1K	104K	2.0M	5.7K

Window	Model calls	Uncached input	Cache writes	Cache reads	Output
Peak day (Nov 21)	798	1.8K	5.42M	297.7M	1.45M
Current plan week (Nov 20–22)	863	about 2K	5.70M	309.7M	1.51M
Month to date (Nov 1–22)	3,121	0.35M	15.86M	902.3M	3.65M
All time (Oct 2 – Nov 22)	3,534	0.35M	18.57M	958.0M	4.19M

Claude Code tokens: Mac (from local session logs)

Window	Model calls	Uncached input	Cache writes	Cache reads	Output
Today / peak day (Nov 22)	400	0.9K	2.90M	155.1M	255K
Current plan week (Nov 20–22)	714	1.5K	3.66M	245.6M	508K
Month to date = all time (Nov 16–22)	1,463	3.3K	5.61M	396.2M	0.96M
Both machines, all time	4,997	0.35M	24.18M	1,354M	5.15M

- **Model mix:** Windows: Opus 5.5 87%, Fable 5.2 13%, Sonnet 5.5 under 1%. Mac: Opus 5.5 100%.
- **Tool activity, Windows:** 1,421 shell commands, 1,181 browser actions, 650 file edits and writes, 511 file reads, 68 web searches and fetches.
- **Tool activity, Mac:** 803 shell commands, about 270 browser actions, 165 iOS Simulator actions, about 35 Suno calls, about 30 computer-use actions, 76 file edits and writes, 52 file reads.
- **Caching:** about 98% of input tokens on both machines came from the prompt cache, which keeps long agent sessions affordable.

Antigravity tokens (from its local logs)

Window	Model calls	Uncached in + out	Total including cache
Nov 22	188	1.86M	37.8M
Nov 21	269	1.66M	34.4M
Total	457	3.52M	72.2M

Antigravity doesn't write its remaining quota to local files; it is in the app under Account, Usage.

8. Lessons learned

Each of these cost time on the first pass. Telling the agent up front avoids them.

All agents

- Set final names in the first prompt (product, repo, bundle ID). Renaming later meant scrubbing 60 files in Deep Void. Both "Deep Void" and "Gravwell" were already taken on the App Store, so check that too.

- At the end of every project, have the agent save what it learned. This was the biggest speed-up of the pilot.
- Logins, CAPTCHAs, passkeys, legal terms and secret-key downloads are the human's job. The agents refused to enter credentials or solve CAPTCHAs, and asked before accepting Apple's API terms. Plan a 30-second hand-off, and sign in in the tab the agent is using.
- Test on real hardware early. Lighting, pacing, ergonomics, sound and GPU-load problems only showed up on real devices.
- "Boring" is a valid bug report. The first dragon saver worked but was dull; asking for big heroes, speed changes and banking flight fixed it.

Working across machines

- **Give each release channel one owner machine.** On Nov 20 the PC uploaded straight to the website and overwrote two newer Mac pages and part of Gravwell. The agent only noticed because it now checks the server against its record before every deploy. After that the Mac became the only machine that deploys.
- **Push before you switch machines.** Gravwell diverged when the PC uploaded a build without pushing it. The Mac merged it cleanly (3-way), but the PC still has to pull before it pushes again.
- **One plan seat across two machines works but fills up.** Check the meter before starting parallel or cross-machine work.

Windows and Claude Code

- OneDrive folders block elevated launches ("Access is denied"), so installers copy themselves to a temp folder before the UAC prompt.
- Unsigned installers get flagged. Defender quarantined one as a false positive. The lasting fix is a code-signing certificate; until then, ship a SHA-256 hash and explain "More info, Run anyway".
- HDR video for YouTube needs a ProRes MOV master with container colour tags; x265 HDR flags alone aren't enough.
- GitHub rate-limits repo creation after about 10 in a row. Wait a few minutes and retry.
- Backslashes in Bash heredocs get mangled and break C# and Python strings. Put scripts in files instead.
- GLSL reserved words (cast, patch) caused shader compile failures. Keep a list in the playbook.
- Parallel sessions work. Separate Claude Code sessions in separate folders ran at the same time without collisions.

Mac and Claude Code

- **Automate the release path once.** Developer ID signing, notarization and TestFlight are now one script per app. Creating the Developer ID certificate in Xcode failed with an "Account Holder" error even though I am the account holder; an openssl CSR uploaded on the developer site worked. Notarizing with an app-specific password returned HTTP 401 twice; an App Store Connect API key worked.
- **Hosting that blocks agent browsers can still be automated.** GoDaddy rejects the built-in browser as an "unusual browser". A one-time SSH key gave hands-off deploys with backups and hash checks, and was better than installing Chrome or uploading by hand.
- **Translate, don't re-port.** Hand-porting each screensaver's C# would not have scaled to 17. A translator script plus a check against the hand ports turned each port into a one-line command.
- **iPhone apps with sound must set an audio session** (.playback). Otherwise Silent mode mutes them; a WKWebView game was silent until this was set. It is now a standing rule.
- **TestFlight quirks:** only one build per version can be in beta review at once, so a scheduled task submits the next one later. App Store Connect shows a blank icon until a build is attached to the version.

- **macOS screensavers:** the Options sheet's buttons do nothing unless its controller outlives the throwaway view System Settings creates. On macOS 27, savers are listed under Wallpaper, Screen Saver, Other.
- **Apple Silicon with no Rosetta:** electron-builder's bundled NSIS is Intel-only; a small shim pointing at Homebrew's makensis lets the Mac build Windows installers.
- **Don't say a window is open until it's verified.** Early on the agent asked me three times to sign in to a browser pane that wasn't visible. It now opens the pane and confirms it is showing before asking.

Mac and Antigravity

- Generated Xcode projects need 24-character hex object IDs and must pass plutil -lint.
- Tell the agent to read the local team ID and patch project.pbxproj itself, rather than asking you to click through signing screens.
- Reduce AI-generated meshes to 2.5K–8K faces. Drive SceneKit from @MainActor, and never call DispatchQueue.main.sync in render callbacks.
- On Apple Silicon, build Windows packages with -c.win.signAndEditExecutable=false and leave the final installers to CI.
- Passkey logins (GitHub device flow) must happen in Safari or Chrome, not the embedded browser.

9. Playbook for a teammate

Setup takes about 15 minutes, plus an hour once for Apple signing if you ship Mac or iPhone apps. Every project then follows the same loop.

Step 1: one-time setup

- Pick one coding agent per machine. Windows: the Claude desktop app's Code tab on Claude Pro. Mac: Claude Code, or Google Antigravity 2.0 on Google AI Pro. One Claude seat can cover two machines, but watch the meter.
- Install the build tools: Git and gh (gh auth login --web --scopes workflow, in a real browser), Python, Blender and ffmpeg. Add Node and Xcode for Mac, iPhone or web targets.
- Apple targets: have the agent set up Developer ID signing, a notarization profile and an App Store Connect API key. You sign in, accept the terms and download the key file yourself.
- Website: give the agent a scripted deploy (SSH key or API) rather than a browser login, and make one machine the only deployer.
- Optional: connect paid creative tools as MCP servers (Meshy API key, Suno sign-in). Connect Claude in Chrome if the agent needs sites you are signed into.
- Use one working folder, with a subfolder and a private repo per project.

Step 2: put guardrails in the first prompt

> *Final name <Name>, repo sposten/<Name> (private), folder Documents\Claude\<Name>. Configure signing and builds from the command line; don't ask me to click through IDE menus. Pause for me only for logins, CAPTCHAs, terms, UAC and secret-key downloads. Verify every build yourself (run it, screenshot it, measure it) before telling me it's done. Push to git before we finish so other machines can pick it up. When we finish, save every lesson to your memory or plugin so the next project is faster.*

Step 3: work like a product owner

- Get a minimal version compiling first, then test it on the real device right away.
- Give feedback in plain language ("too fast", "boring", "red looks pink", "no sound on my phone") and let the agent decide the fix.

- Add targets (web, other platforms, CI, installer, TestFlight) only after the core feels right.
- Ship: push to the private repo, produce signed installers and SHA-256 hashes, and let the deploy machine publish.
- Save the playbook, then clone the finished project as the template for the next one.

Step 4: watch the quota

- Claude Code: check the usage card (5-hour and weekly %) before parallel sessions or a second machine. One heavy day used most of a weekly Pro allotment, and the 5-hour limit stopped work once.
- Antigravity: check Account, Usage in the app.
- Meshy and Suno: check credit balances before large batches. Free procedural Blender models are often good enough.

10. Open items as of Nov 22

- Gravwell and Deep Void TestFlight builds are waiting for Apple's beta review; the public join links open once they are approved.
- Gravwell Overdrive is on a branch, not merged. Its private preview folder must be removed from the site afterwards. The Mac app and real iPhones haven't been benchmarked yet.
- Windows installers and the Deep Void exe are still unsigned (SmartScreen warnings) until a code-signing certificate is bought.
- Forwarding my other GoDaddy domains to navesink.net is waiting on a GoDaddy API key.
- Deep Void and Gravwell were tested on the iPhone Duo and its simulator only. The minimum is iOS 18 for Deep Void and iOS 17 for Gravwell.
- The PC copy of Gravwell needs a git pull before its next push.