Google Antigravity Quotas: Why the Pro Plan Feels Unpredictable - and When Ultra Actually Helps

TechRounder PDF Edition

Live article:
https://www.techrounder.com/ai/google-antigravity-quotas-why-the-pro-plan-feels-unpredictable-and-when-ultra-actually-helps/

By Vipin PG | Published March 4, 2026 | Updated March 4, 2026 | Format: Analysis | 5 min read

In brief

Google Antigravity's Pro plan frustrates users not because quotas exist, but because they feel unpredictable, poorly communicated, and inconsistently enforced.

Google Antigravity has become a go-to setup for developers who want a "multi-model" workflow-fast planning with one model and deep coding/debugging with another. But alongside the hype, there's a recurring pain point: quotas that feel inconsistent, poorly explained, and sometimes drastically restrictive on the Pro plan. Meanwhile, some Ultra subscribers report the opposite: long coding sessions without hitting limits, especially when they route planning through lighter models and reserve premium models for implementation.

This article unpacks what's happening with Google Antigravity's quotas, why the issue is less about "limits existing" and more about trust and predictability, and what practical workflows can reduce quota pressure.

The Core Issue: "Generous Limits" vs What Users Experience

The most common frustration is not that Google Antigravity has limits-it's that the limits can feel unpredictable.

- Pro access feeling fine briefly, then becoming severely constrained
- High-end model usage (especially heavy coding/debugging) triggering long cooldowns
- Confusion about reset timing (short reset windows vs longer "hidden" constraints)
- A quota monitor that exists but doesn't clearly show where you stand-particularly on any longer-term caps

When customers pay for a tier that's marketed as workable for daily use, they expect:

- clear constraints up front,
- consistent enforcement,
- and a UI that makes usage understandable.

When those things aren't true, the subscription starts feeling risky even before the quotas are "too low."

Weekly Quotas: The Change That Triggered the Trust Backlash

A major accelerant has been the perception that weekly quotas were introduced or tightened without strong in-product communication.

The backlash comes from a specific dynamic:

- subscribing is simple and clear

- using the product feels unclear

If users can’t answer basic questions like:

- "What is my weekly cap?"
- "How close am I to hitting it?"
- "Is this temporary because of abuse/attacks, or permanent?"
- "Why did I suddenly get restricted after a few high-end model runs?"

...then they’re not really buying a "plan"-they’re buying uncertainty. And uncertainty is deadly for coding workflows, where you might need 30-60 iterations in a single day.

Why Quotas Tighten: Security and Abuse Responses (and Why That’s Not the Whole Story)

There are legitimate reasons Google Antigravity may tighten quotas:

- demand spikes
- infrastructure constraints
- abuse, automation, or aggressive extraction behavior that forces defensive throttling

But even if quota tightening is justified operationally, the product obligation remains the same:

- disclose changes clearly,
- instrument them properly,
- and communicate them in the app, not only via external channels.

When paid users feel changes were "stealthy," they interpret it as either:

- poor product management, or
- monetization pressure disguised as "policy."

Either way, trust erodes.

The Ultra Experience: Why Some Users Say It Feels "Limitless"

A subset of Ultra users report a noticeably smoother experience-often describing:

- longer coding sessions without hitting hard stops
- less frequent cooldowns
- better productivity when rotating models strategically

A common Ultra pattern:

- Use Gemini 3.1 (high) for planning
- Use Opus for coding/debugging
- Optionally use Sonnet for planning if preferred

This approach works because planning can consume huge tokens without needing the most expensive model. Reserving the premium model for implementation reduces your "high-cost" usage footprint.

However, Ultra doesn’t automatically solve the broader concern: if quota rules can change without clear notice on one tier, users worry the same can happen on the highest tier too.

"Just Upgrade" Isn’t a Real Answer

Even if Ultra genuinely improves the experience, "upgrade to Ultra" fails as a product answer for three reasons:

- Price gap: the top tier can be dramatically more expensive than Pro
- Trust gap: after unclear quota changes, users hesitate to pay more
- Policy risk: users fear weekly quotas and stricter enforcement expanding upward

So even satisfied Ultra users don’t eliminate the underlying demand: clear rules and stable expectations.

Quality and Reliability: When Quotas Combine With "Servers Busy," It Feels Worse

Some complaints aren’t purely quota-related. They’re availability problems:

- "servers are busy"
- inconsistent throughput
- support that can’t clarify restrictions or fix account-level issues quickly

To a user, it doesn’t matter whether the cause is capacity, throttling, or demand shaping. The effect is the same: the tool becomes unreliable in the middle of work.

Data and Code Concerns: The Hidden Barrier for Professionals

A separate concern is what happens to your prompts and code:

- Does Antigravity train on conversations and code by default?
- Is there an opt-out?
- Are paid tiers treated differently?
- What should you avoid pasting if you care about IP?

For developers, unclear answers to those questions can be a deal-breaker-regardless of quotas-because it limits professional usage.

Practical Ways to Reduce Quota Pressure in Google Antigravity

1. Split planning from execution
    - Do architecture/planning with a high-throughput model.
    - Use the premium model only for:
        - tricky bug hunts
        - refactors
        - tests
        - hard implementation details
2. Don’t "chat" when you don’t have to
    - Treat prompts like jobs: one prompt = one deliverable.
    - Avoid long conversational loops that burn tokens.
3. Prepare a compressed "brief" before invoking premium models
    - Draft:
        - requirements
        - constraints
        - failing logs
        - minimal repro
        - current code snippet
        - expected output

- Then send a single high-quality prompt.

4. Keep context lean
    - Large context windows are quota multipliers. Trim:
        - redundant logs
        - duplicated code
        - long histories
5. Work in batches
    - If cooldowns exist, shift to:
        - writing tests locally
        - preparing the next prompt
        - running builds
        - debugging in IDE
    - Then return with a tight prompt.

What Google Antigravity Needs to Fix (If It Wants Trust Back)

- Checkout disclosure: clear daily/weekly caps, not vague "generous limits"
- Accurate quota meter: real-time, reliable, and visible
- Reset clarity: simple explanation of how cooldowns and weekly caps work
- In-app change notices: not only external posts
- Policy stability: if changes are temporary, say so explicitly; if permanent, own it
- Better escalation: support that can explain restrictions and resolve edge cases fast

Users don’t need infinite usage. They need predictable usage.

Bottom Line

Google Antigravity can be extremely productive-especially when you split tasks across models and reserve premium models for high-value work. But the quota controversy isn’t just about limits. It’s about whether paying users can trust the rules to be clear, consistent, and communicated like a real product.

If Google Antigravity wants Pro to feel like a serious plan-not a funnel-it has to make the quota system understandable, measurable, and stable. Until then, the smartest strategy is to optimize your workflow around token economics and treat premium model calls like expensive operations-because, functionally, that’s what they are.

References

1. antigravity.google - pricing - https://antigravity.google/pricing

2. getmaxim.ai - articles / beginners-guide-to-tracking-token-usage - https://www.getmaxim.ai/articles/beginners-guide-to-tracking-token-usage/