EVERYONE NEEDS A SAMWISE / FREE PROMPT

# The gauntlet loop, with a brake.

A prompt that has several AI agents build something while a separate critic marks it, round after round. The original never says when to stop. This version does.

TL;DR

Matt Shumer's gauntlet-loop prompt tells Claude to keep going until a critic is wowed, and nothing in it ends the run. This version keeps the builders and the separate critic, and adds caps on builders, rounds and time, plus a report at the cap. Copy it, fill in the brackets and paste it after your own spec. In our two small tests the looped build beat the plain one with blind judges, at a much higher cost in time.

## 01 THE PROMPT

## Copy the prompt with the brake

Paste it after your own spec. Fill in the brackets. Keep the caps, or it runs until you stop it.

GAUNTLET LOOP, WITH A BRAKE

```
I want you to build [THING] at the level of [NAMED REFERENCE]. Start from the spec/design I
give you, do not invent the direction.

Fan out sub-agents, one per [PART 1 / PART 2 / PART 3 / PART 4]. Each builder owns one clearly
delimited section of the code. A SEPARATE critic sub-agent with fresh context reviews the
result visually (screenshots) against [REFERENCE], scores it out of 10, names the single
biggest gap, and sends it back if it is below [BAR].

Caps: at most [N] builders, [R] critic rounds, [T] minutes wall clock. At the cap, stop and
give me: what passed, what failed, the critic's last score, and the one thing you would fix
next.
```

The caps are the point. The original says “don’t stop until each sub-agent is utterly wowed,” and nothing in it ever ends. The README of Shumer’s own repo notes that parallel builders kept breaking each other’s assumptions on coupled systems (lighting, sky, tonemapping) and that a single sequential owner did better. So give each builder one part, cap the rounds, and stay in charge of when it ends.

## 02 THE ORIGINAL

## Matt Shumer's prompt

This is the prompt behind Claude of Duty (github.com/mshumer/Claude-of-Duty), a Three.js first-person shooter Matt Shumer published as built from a single prompt. Copied as published, for reference. Credit to him for the idea.

ORIGINAL PROMPT, MATT SHUMER

```
I want you to build a first-person shooter at the level of the most recent Call of Duty games.
It should be utterly perfect, visually beautiful, with every single thing done at AAA quality—
from textures to physics to anything you could think of.

Fan out sub-agents and have sub-agents tackle each one individually so that the game is utterly
perfect. You should /loop on each item and have a separate sub-agent check it visually to
ensure it looks triple A. That separate sub-agent should be a really harsh critic, and if it
doesn't look triple A, it should keep going.

Don't stop until each sub-agent is utterly wowed with the quality when compared with the actual
Call of Duty game. It should literally compare them side by side blind and say which one looks
better. Do this in ThreeJS. /loop until it's utterly perfect. Fan out sub-agents and ultracode.
```

One model building and a second model judging, in a loop, has a name. Anthropic calls it the evaluator-optimizer pattern in Building effective agents (December 2024) and says it works best when you have clear evaluation criteria. A community skill, gauntlet-loop (github.com/duolahypercho/gauntlet-loop), packages the prompt, and its README says it plainly: “You are the brake. The loop will not finish on its own.”

---

**03** WHAT I FOUND

# What the tests showed

## The original, by its own README

- Its “Honest assessment” section asks whether the game matches modern Call of Duty. The answer is “It does not.”
- In a blind side-by-side, every critic in every round picked the real Call of Duty frame.
- Its process note compares two ways of working. Three rounds of six parallel agents moved the score +0.46 and left the frame-ruining defect count at 60, then 47, then 66. One sequential pass by a single owner moved it +1.00 and cut defects from 66 to 26.

## Test 1: a log cabin, plain prompt against the loop

Same task for both: a first-person walkthrough of a cozy log cabin in Three.js, one HTML file, same model tier, same 45-minute cap.

| | PLAIN PROMPT | GAUNTLET LOOP |
|---|---|---|
| *Time* | 7 min 45 s | 38 min 04 s |
| *Agents* | 1 | 7 |
| *Lines of code* | 736 | 1,234 |
| *Blind judges* | 5/10, 5/10 | 8/10, 8/10 |

Both blind judges picked the loop build. The loop’s own critic scored it 7.5, then 7 after a fix pass, so the score went down. One builder, the lighting one, produced nothing in more than 30 minutes and the orchestrator wrote that section itself. One task, so it does not show which part of the loop did the work.

## Test 2: an animated reel in a 1940s cartoon style

Here a fresh critic scored each round against real frames from 1940s cartoons. Its overall score went 3, 3.5, 3.5, 4 out of 10 over three rounds. The biggest moves were background (2 to 5), effects (2 to 5), posing (3 to 5) and motion (2 to 4). The run took about 53 minutes and 17 agents.

The brake fired after round two because the score had not risen. I overrode it once, because the stall was my own mistake: I had given the motion builder the wrong target. Two blind judges, with the order swapped, both picked the looped cut over the original.

> MY TAKE *Keep the separate critic and give it something real to compare against. Own one part per builder. Cap everything, because a prompt that says "don't stop" will not end on its own.*

Small tests, one model tier, run on 11 October 2026. Treat them as a hint, not a benchmark.

---