

BORN AGAINST, OR WHY HOBBY PROGRAMMING COMMUNITIES ARE AGGRESSIVELY AGAINST LLM USAGE

Fogus

2026.08.04

I came across a [GH thread related to chess engine development](#) that made me think of why hobby programming communities are increasingly hostile toward LLM development. While the thread doesn't give a lot of insight into answering the question, it prompted me to think about it a bit. I've seen similar sentiments expressed in other niche hobby programming communities like OSDev, LangDev, TxtDev, EmuDev, RLDev, the demoscene, and code golfers. The general consensus seems to be that the knowledge that these communities work in is hard-fought and the use of LLMs is a form of missing the point entirely. In these communities (keep in mind that there is an implicit "not all..." throughout) the process of mastering a difficult field itself is the product, and something that runs is generally a nice-to-have.

Further, I've noticed that even in the instances where there was earnest early engagement with LLMs in some of these niche communities, the well was quickly poisoned by a combination of a lack of a deep understanding by the LLM practitioners, and a vitriolic subset of those communities that view the LLM enterprise as a form of cheating. Granted these communities have, in general, historically been characterized by feverish gatekeeping and painstakingly slow progress, so it makes sense that there might be a desire to grab some easy cachet by bursting onto the scene like the Kool-Aid man. OH YEAH.... OH NO!

In traditional niche dev circles, respect is earned slowly through years of activity in their respective fora, sharing elegant code, displays of genuine curiosity, and through sharing deep domain knowledge along the way. At the end of the day, these communities don't care if your code works at all, but instead care that you know why and how it works. To me, an LLM functions best as a force multiplier, not a surrogate. In the hands of an expert who already understands a domain deeply, it could act like a lever.¹ But in these niche communities, the entire exercise is in the learning. Using an LLM to generate the finished piece doesn't make us craftsmen; it just robs us of the craft.

:F

This is the latest in my evolving thoughts on LLMs. Also see: [LLMe](#) and [Mind the van Emden Gap](#)

1.

That said, expertise offers no natural immunity against being fooled by LLMs.↵