

How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)

Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

In this document, I will explain how to develop a Nintendo 64 software-based emulator known as Project64 Version 1.6. I'll also be going over my experience with the Men in Black, AKA Project64 Team so you can understand why you should keep your distance and watch out for them, as they may try to silence you, delete messages from you when they are able to, or attempt to manipulate you by trying to mislead you into believing that you must give them all of your code or cannot develop off of their public repositories, all without them giving any regard for the GitHub terms of services they agreed to by using the GitHub platform. Some truly mean people. I'll begin by going over the basics of what you need to know to develop Project64 1.6 in steps, with optional steps as well. Starting with step 0...

Step 0: Have a [GitHub](#) account.

Step 1: Fork a [public Project64 1.6 repository](#) without having been blocked by its owner organization or user.

Step 1.5 (optional): While you're at it, fork every repository that belongs to [PJ64Team/MiB64](#) on GitHub while you can because when they see you've forked one of their public repositories or forked a forked public repository of theirs to develop it, they are inclined to block you like they did me, and once you're blocked, you can no longer fork their repositories. So make sure you fork everything they've got while you still can! Not only that, but history proves that [MiB64](#) and [PJ64Team](#) are [prone to deleting and/or privatizing their public repositories](#), which makes forking even more important because once a repository is forked, [the fork will stay up as an archive of the parent even if the parent is deleted/not available](#).

Step 2: Setup [GitHub Desktop](#) and clone your forked [public Project64 1.6 repository](#).

Step 3: Install Visual Studio, preferably its Community 2013 version with update 5 (1 month trial).

En_visual_studio_community_2013_with_update_5_x86_dvd_6816332.iso <- You need this file!!

This iso installer's MD5 is as follows: D84117E49747D03105CEB0D04695DC2E

Vs 2013: <https://mega.nz/file/NfJRgKAL>

Key: dkzMryaid0Cq81R8Wioad3Z6Ld8ZQ61afKeRh6L9ReY

Step 3.5 (optional): Download other useful tools like [Notepad++](#), [HxD](#), vs 2019/22. These tools can be used to help develop databases and plugins for the emulator.

Vs 2019: <https://mega.nz/file/1C5FyJxS> key:

uHKr9NchVRNCiGd8c_drqiDy4NUShtvpOKlelel_ihY

Vs 2022: <https://mega.nz/file/IDwxWA5I> Key:

4LTVpy8CBmJN-4R23kAOsrT_oW2S5h4ljkvNTvC0X1A

How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)

Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

Step 4: Open the vcxproj in Visual Studio and confirm you can successfully compile an executable. You may need to switch to an inferior toolset like v143 from the project's properties if not using visual studio community 2013 with update 5.

Step 5: If you can compile an executable, now you can begin making changes to the source to try to fix a bug, customize, or add a feature to it, then compile and test it to make sure it acts as you intend it to. If you can't compile the executable and it's because you're getting a missing include file error, you may need to download something like afxres.h from Windows Driver Kit 7.1 and put it in the source code folder and make sure the #include that calls for it is doing so with " " marks instead of < > marks. Remember that you can use ctrl-f as a crutch to help you find your way around the code to the parts you wish to change. It's written in the C language which is very easy to understand even without learning it, so once you start working with the code, you should be able to get the hang of this fairly quickly.

Step 6: Make sure the changes to the code are in the cloned repository and then send it as a commit to the online repository from the GitHub Desktop app, then use push to origin.

Step 6.5 (optional): Project64 has a per-ROM settings and per-ROM cheat system which functions using database text files. You can improve these databases by fine tuning the emulation settings and adding new entries to it from the user interface for ROMhacks. To read about the core/ROM settings and get an idea of what they do, please see the Surreal64 documentation on my Project64 1.6.2 repository. You can additionally modify ROMhacks internal names and checksums to apply Jabo's Direct3D8 per game hacks as appropriate. See my 1.6.2 user guide for more information about this.

Step 7: Tag all of your releases and include the necessary licensing information (and optionally, GitHub terms of service section 5) to ensure you remain legal about this like I have.

Step 8: Document any known issues you find that you don't have or don't plan to provide an immediate solution for, and fix any discovered issues that you are able to resolve without making any bad compromises.

And that's it, you're now developing Project64 1.6! You don't even need to know any programming languages either, just make appropriate changes based on what you can make of the code until it does what you want. Now for the 2nd half of this document: BEWARE of the [MiB/PJ64Team](#). Because if you allow yourself to get caught up in their personal affairs or vice versa, you put yourself at risk of being frustrated with their negative attitude towards addressing bad code and forking of their public repositories. But before I explain my experience with them, here's what you can do to protect yourself from those risks:

Step 1 of 1: Block the user known as [TheGent](#) and anyone else who is apart of the [MiB/PJ64Team](#) organizations who appears to associate with him on repositories managed under said organizations or anyone you see engaging in gossip about the forking of public repositories because god forbid someone [following GitHub's terms of service](#). To block a user,

How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)

Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

click on your profile icon in the top right, go to settings, click on moderation, click on blocked users from the dropdown, then type [TheGent](#) and click on the Block User button.

Congratulations! You've now taken proper action against a troll because blocking a user prevents the following on all of your repositories: opening or commenting on issues or pull requests, starring, forking, or watching, and adding or editing wiki pages

And now, I will get into my experience with the [MiB/PJ64Team](#) organizations. In order to do that, we need to rewind back to the beginning of the year 2024.

In 2024, [a vulnerability was discovered in Project64 1.6](#) and I wanted to use 1.6 safely, so I dug up some old code to compile it. The vulnerability didn't work on the google code variant so I made a video saying the vulnerability seems fixed. I also commented about something I heard about called [Project64 Legacy](#), which looked like an update to 1.6 which also claimed to have fixed the vulnerability issue but I kept running into issues with it where the games I used to play wouldn't work, so I said it was a sham and left it at that. This pissed off the Project64 Legacy devs, and I felt guilty about calling it a sham at the time because they convinced me that their legacy emulator was not intended to be an update to 1.6 and so I should not expect the same games to work. So I took down the video and then apologized to the team. The ringmaster of the legacy team known as Gent then proposed a new emulator project called Project64 1.6 Plus after he recovered source code for Project64 1.6.1 from his plugin developer friend Jabo. The idea of 1.6 Plus was [to establish a true vulnerability-fixed version of 1.6 which retains its compatibility with ROMhacks by emulating the exact same](#). However, there was reportedly a lot of internal drama about this because some team members didn't like the idea of working on an emulator that doesn't conform to N64 hardware limitations, so the team decided to stop working on the project and leave it up to others to "pick up the project". But just before release, while I was trying to work on a couple of databases for the project from my iPhone while my PC was unavailable, I had a conversation with Gent (**see page 6 of this document for the full convo**) in his [PJ64Team/MiB64 server](#) where I told him I omitted some cheat codes which are known to crash in Majora's Mask and he asked me to open up an issue about it on GitHub and record a video of it. Since I wasn't able to do that without my PC, I told him how to reproduce the crash by simply enabling the code in the game after starting up a save file. He refused to do this, and then said that fixing something is better than removing it like I did. So I said "then fix it", and he said "heh" "im going to ignore that" "you cant fix what you dont see" and then told me again to record a video of it even after I've told him my PC is unavailable at the moment. I asked him if this is the kind of person he wants to be, then he affirmed by threatening to delete the whole project altogether for me having brought this up with him. I then voiced my frustration with his attitude, stated that this behavior is childish, and talked about how glad I will be when I get a PC to work on a forked version of this myself. Shortly after this conversation, I got a notification from the server and saw I was muted and a couple of my messages had already been deleted from the conversation. So I got out of there and blocked Gent so that I wouldn't hear anymore childish crap from him, then I began planning my fork, beginning active development of it as soon as I got my PC back while keeping it private until I was ready to show the world the best of what I can do.

How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)

Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

Eventually, Christmas came and [I decided to publicize my fork under the name "1.6.2"](#) since 1.6 Plus was based on the source code of 1.6.1. Although I was not obligated to, I gave credit to the developers of 1.6 plus and put Gent in the special thanks section of the user guide as a way of being respectful for the contributions he made to the 1.6 plus that 1.6.2 derived from. I also included the licensing information that came with the source code in all of my releases, both binary and source. Development of 1.6.2 would still continue, but then I heard that the [MiB64/PJ64Team](#) people were angry about 1.6.2 because I didn't have an about section in the emulator. Gent then updated the readme on the 1.6 plus repository to say this: [Project64 1.6 Plus is a free and source-available emulator for the Nintendo 64 and It is written in C/C++ currently only for Windows. What does source-available mean? It means if anyone would like to contribute to the project they are welcome to but need to create a PR showing all changes without any releases before the PR is viewed and agreed on! What does it not allow? It does not allow forking of the project and changing without asking permission beforehand \(Create a PR and wait for a response\) It does not allow forking of the project, downloading and changing offline and creating releases without express permission beforehand via a PR that shows all changes\)](#)

But what's funny about Gent's readme is that not only was the project classified as "free" and "open source" upon its official release instead of "source available", but GitHub's terms of services explicitly states as follows: ["If you set your pages and repositories to be viewed publicly, you grant each User of GitHub a nonexclusive, worldwide license to use, display, and perform Your Content through the GitHub Service and to reproduce Your Content solely on GitHub as permitted through GitHub's functionality \(for example, through forking\). You may grant further rights if you adopt a license."](#)

And this is the license that was adopted for Project64 Versions [1.6](#), [1.7](#), and [1.6 Plus](#):

Project 64 - A Nintendo 64 emulator.

(c) Copyright 2001 zilmar (zilmar@emulation64.com) and Jabo (jabo@emulation64.com)
pj64 homepage: www.pj64.net

Permission to use, copy, modify and distribute Project64 in both binary and source form, for non-commercial purposes, is hereby granted without fee, providing that this license information and copyright notice appear with all copies and any derived work.

This software is provided 'as-is', without any express or implied warranty. In no event shall the authors be held liable for any damages arising from the use of this software.

Project64 is freeware for PERSONAL USE only. Commercial users should seek permission of the copyright holders first. Commercial use includes charging money for Project64 or software derived from Project64.


How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)


Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

The copyright holders request that bug fixes and improvements to the code should be forwarded to them so if they want them.

Gent then opened a discussion on my repository to say this in the general channel:


You need to read the Propject64 1.6 Plus ReadMe on the Github Page you forked #1


 Closed TheGent started this conversation in General

**TheGent** 3 days ago edited ▾ ⋮


<https://github.com/pj64team/Project64-1.6-Plus/blob/main/README.md>

And for everyone's information there was never a Project64 1.6.1, this guy is misleading you. This is a very early source of Project64 Legacy 1.6.1 with a 1.6 Rom Hack compatibility but no means a 1.6 source. Please do not be fooled

↑ 1 


Category
 General

Labels
None yet


2 participants
 

1 comment Oldest Newest Top

To which I responded with:

**BruceShankleIV** 8 minutes ago Maintainer edited ▾ ⋮

You called it 1.6.1 so I'm calling this one 1.6.2, honestly the numbers aren't even that important. What matters is that it's working. And to be honest Gent, I think anyone who joined GitHub is not going to fall for your readme prank. Anyone who has read github terms of service section 5 and read the license provided in the project64 source files anyways. If you really didn't want this to be forked, you shouldn't have made the repo public and released as "free" and "open source". Get over it. Closing this. Also, you've been blocked for what you tried to pull last year so don't even think about trying anything again.

↑ 1 

0 replies

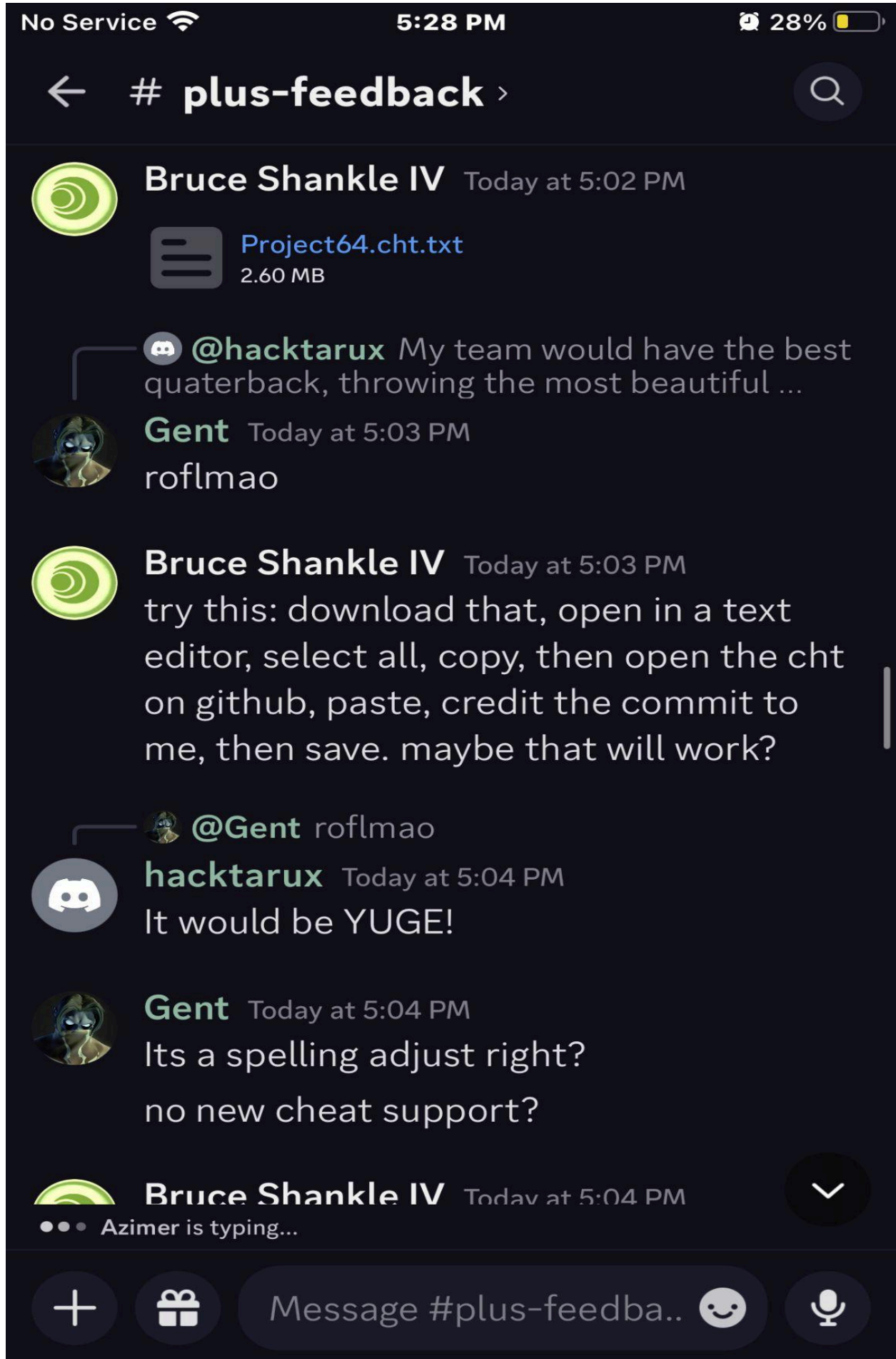
Write a reply

And then I deleted the conversation after Gent got the memo, because I don't want his name anywhere near my project after what he tried to pull on me before regarding his childish attitude in the [MiB64/PJ64Team server](#) and the subsequent discord mod abuse directed towards me, and now with this readme that doesn't even match up with the ToS we both agreed to.

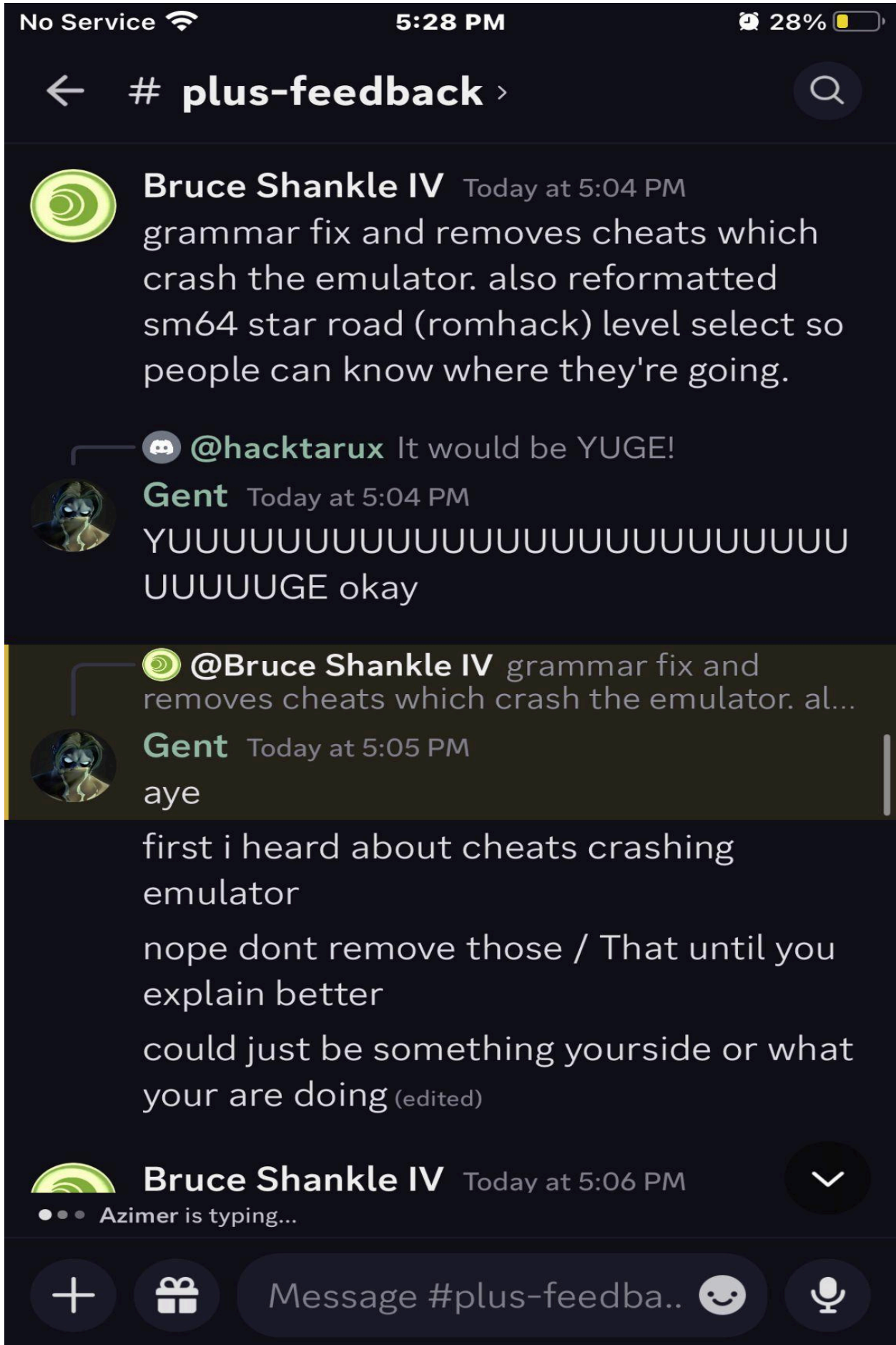
Around the same time, a troll entered the comments section of my release video and said "These releases are not authorized by the MIB64 team and violate the licensing of MIB64 and its associated projects." **I am implied to believe that Gent of [MiB64/PJ64Team](#) and potentially others are now spreading rumors about me violating their licensing and my releases being unauthorized. If the licensing was really violated, then it'd have to not be included in all of my distributions thus far (of which it is), and if my releases weren't authorized, then the public repository of the source code that I forked from would have to not be public because setting a repository as public authorizes anyone who abides by the license to fork the code, hence authorizing my and anyone else's releases.**

This was my experience with the [MiB64/PJ64Team](#). I hope this helps you understand why I see Gent as an annoying troll and tell you to beware of the [MiB64/PJ64Team](#) from now on.

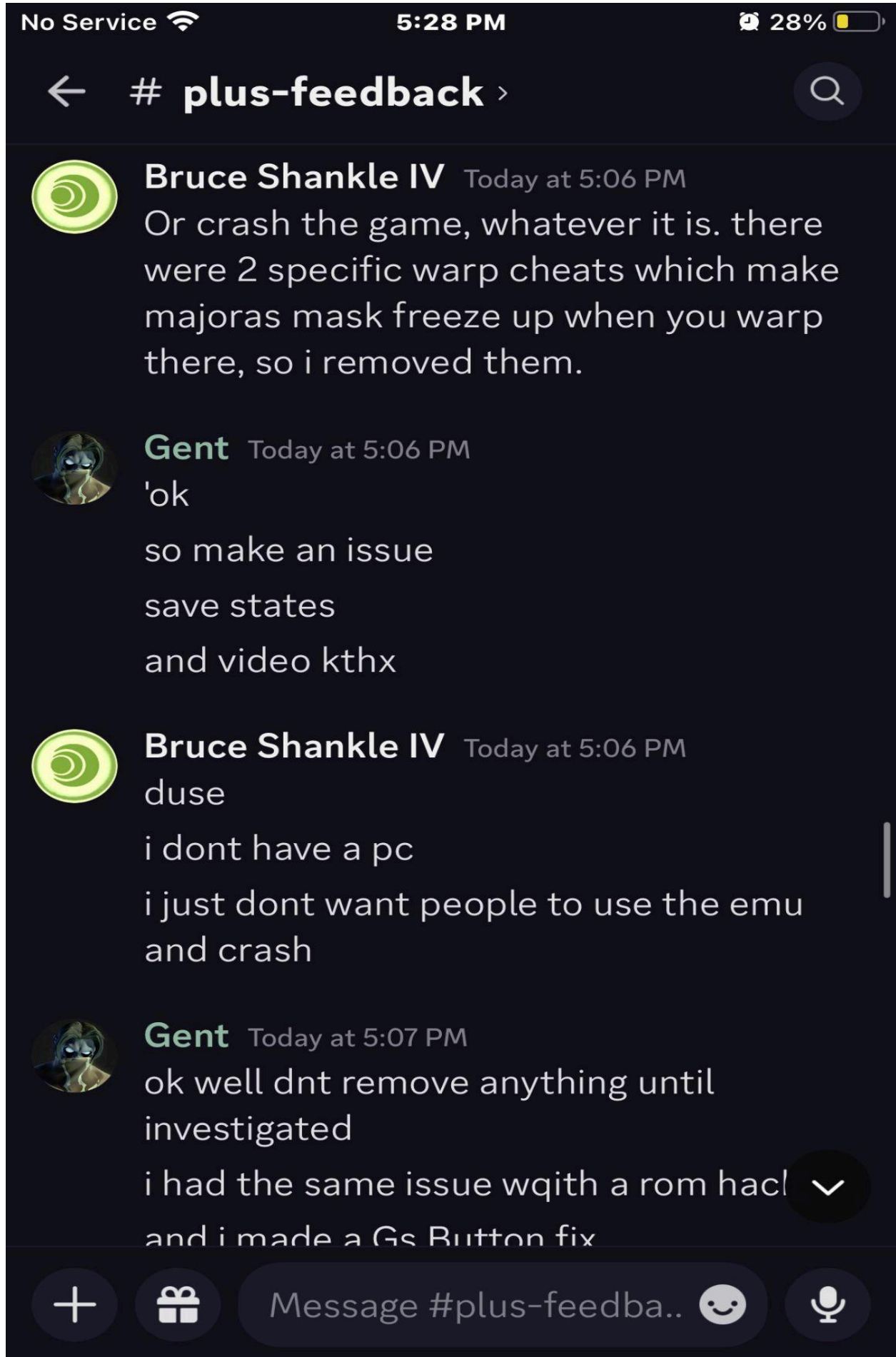
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



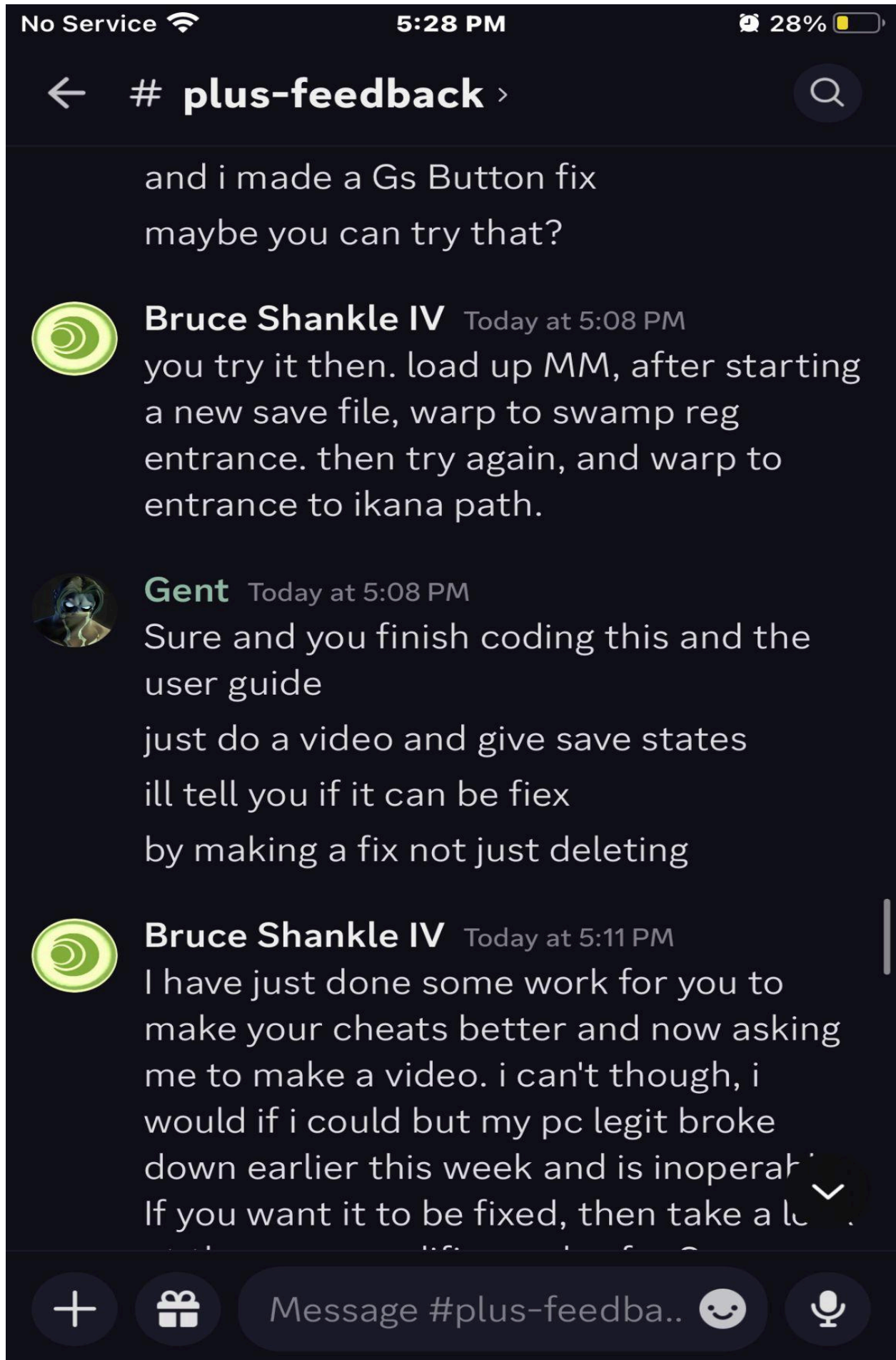
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



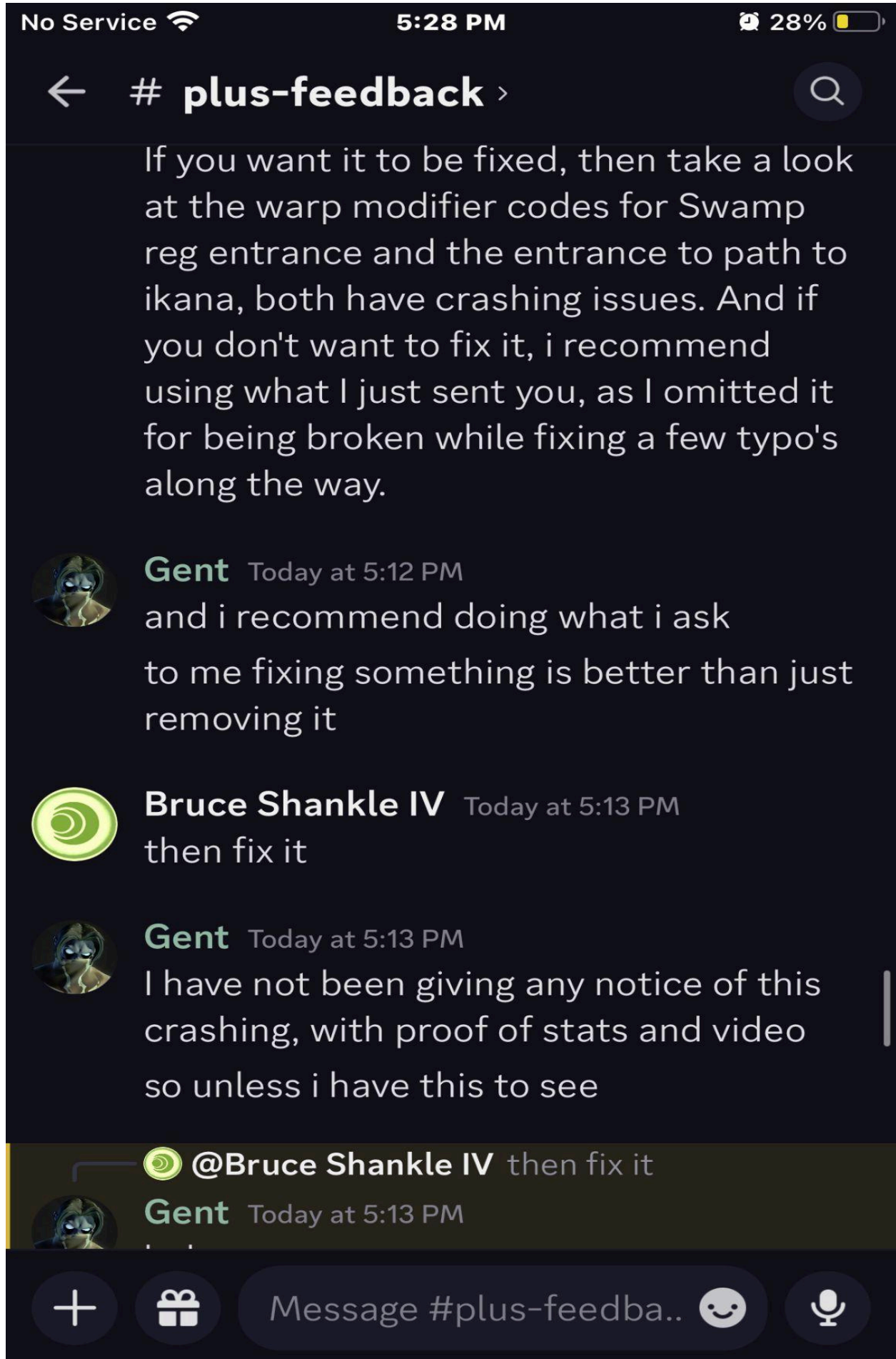
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



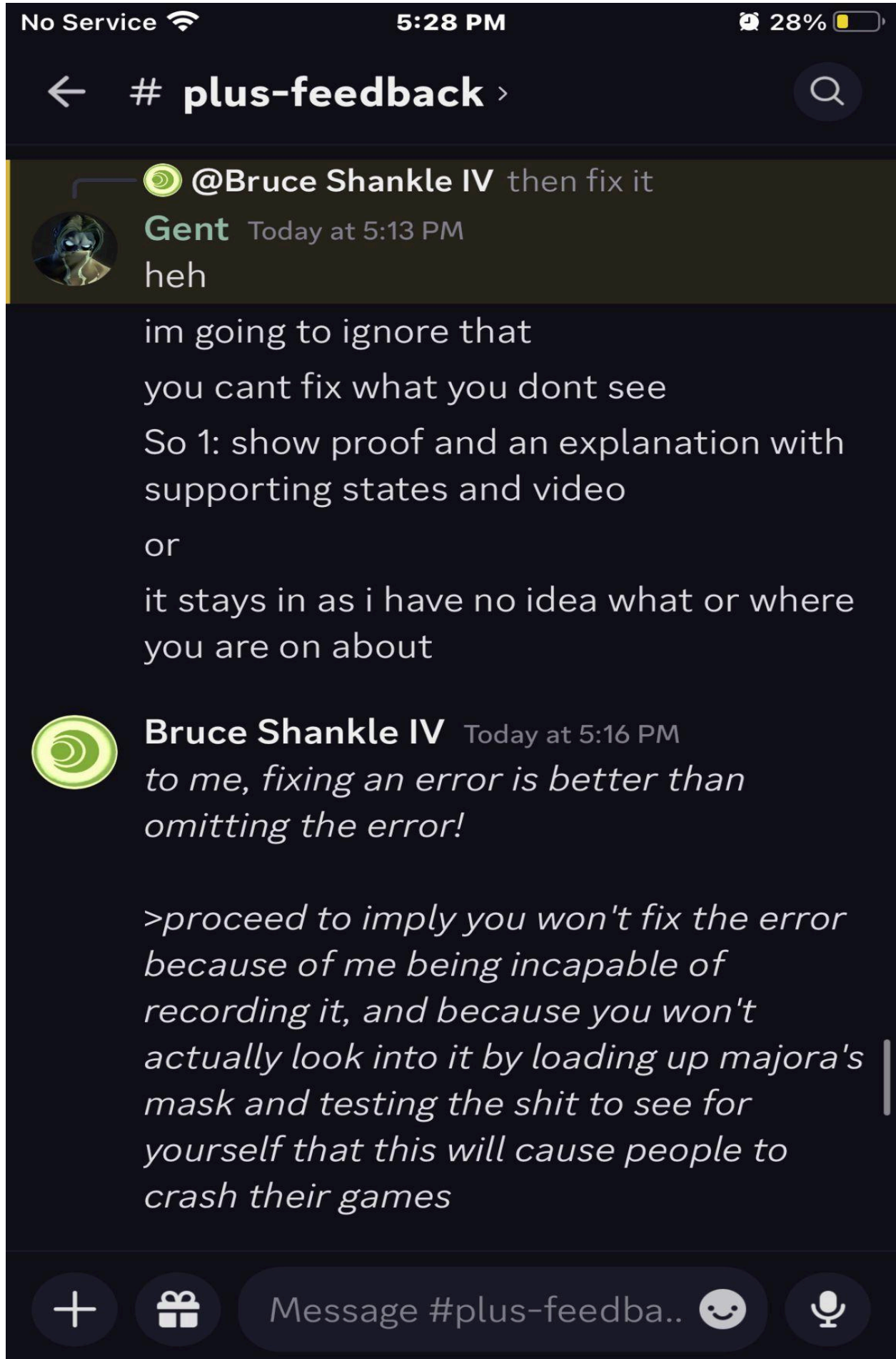
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



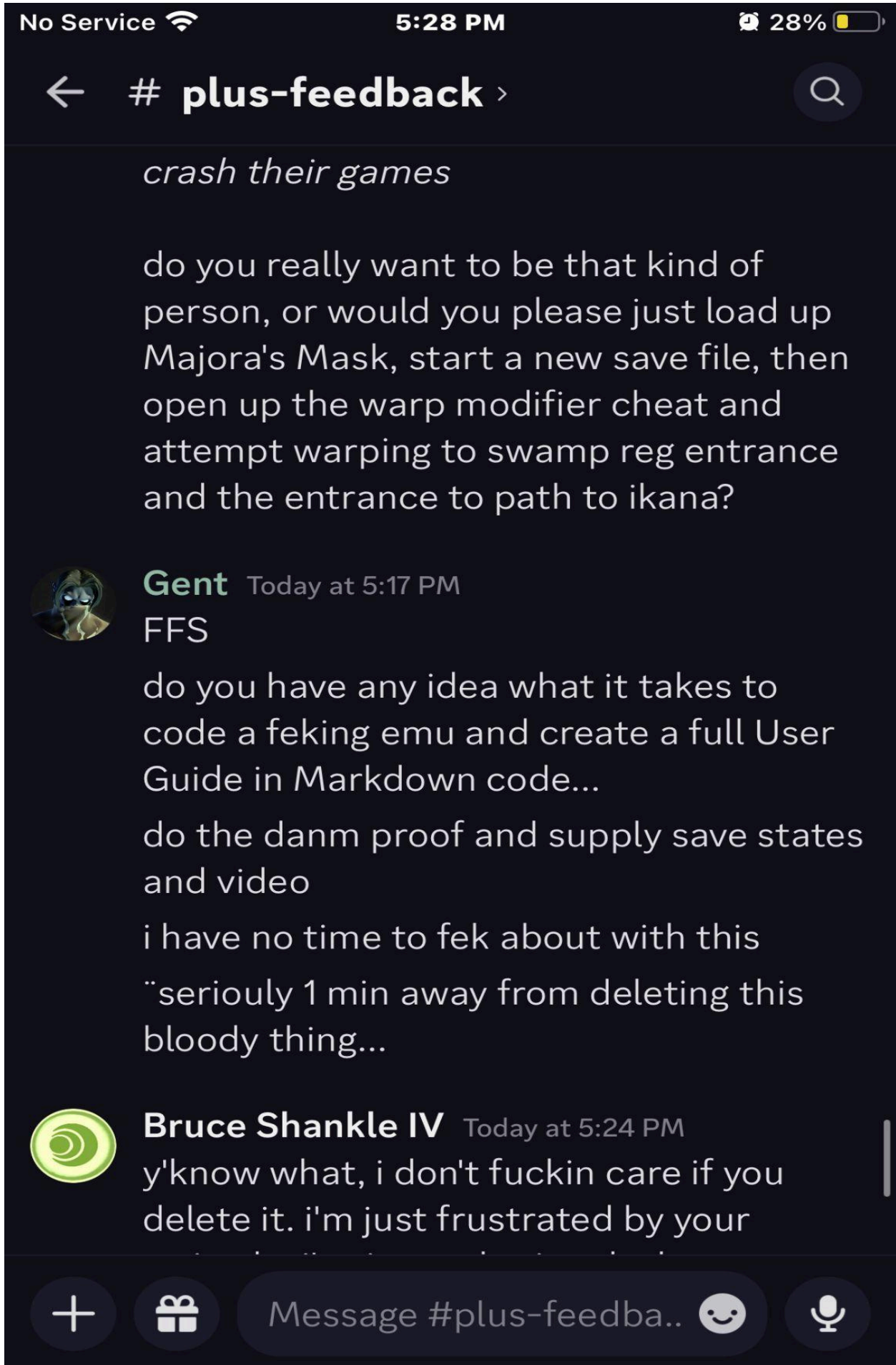
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



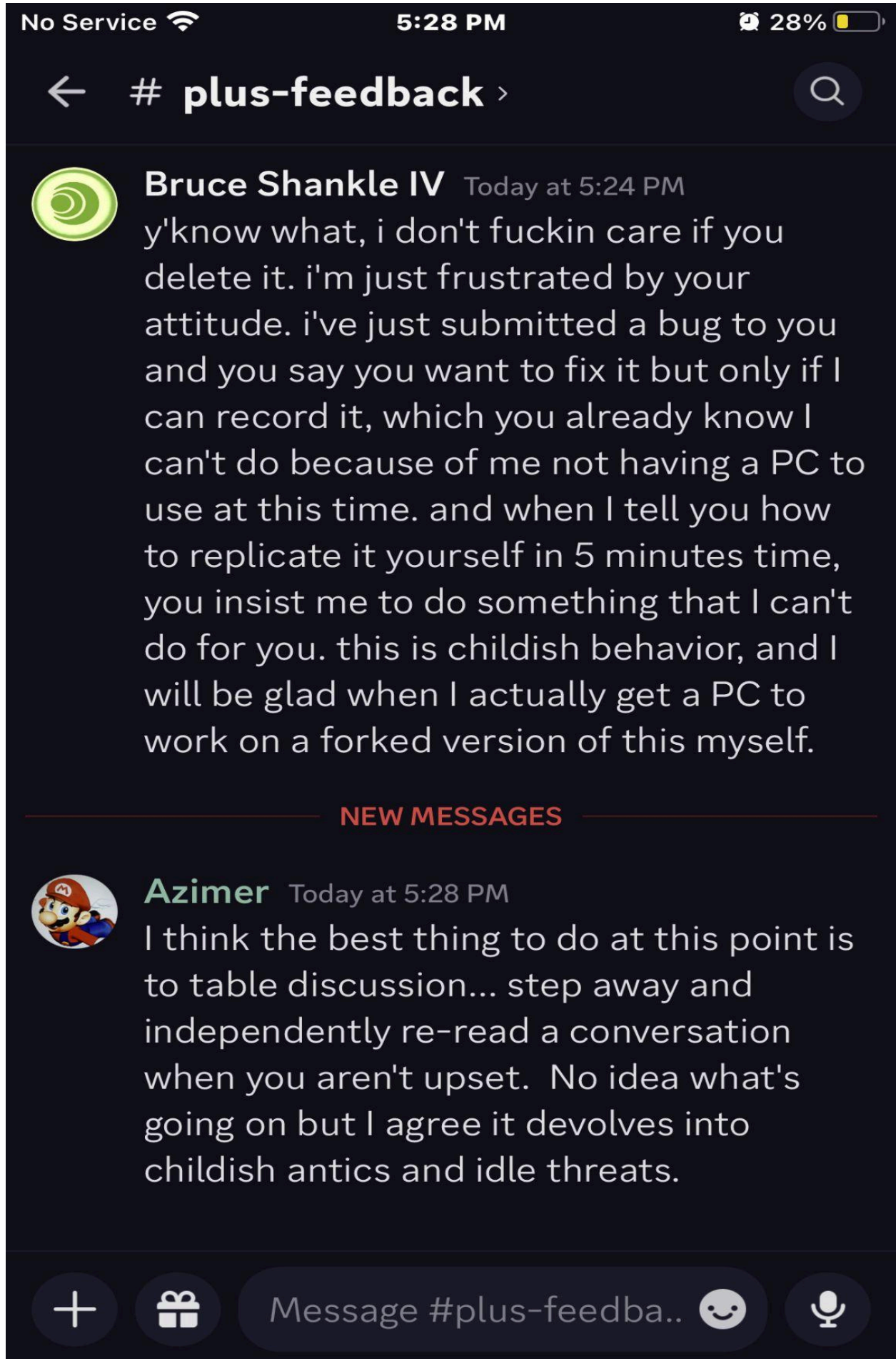
How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.



How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

Spectrum

LTE

10:43 AM

84%

1

plus-feedback

@Gent

I'm a Fabulous Coach

hacktarux

Yesterday at 5:02 PM

My team would have the best quaterback, throwing the most beautiful passes you've ever seen.

@hacktarux

My team would have the best quaterback, throwing the most beautiful ...

Gent

Yesterday at 5:03 PM

roflmao

Bruce Shankle IV

Yesterday at 5:03 PM

try this: download that, open in a text editor, select all, copy, then open the cht on github, paste, credit the commit to me, then save. maybe that will work?

@Gent

roflmao

hacktarux

Yesterday at 5:04 PM

It would be YUGE!

Gent

Yesterday at 5:04 PM

Its a spelling adjust right?

TIMED OUT

06d 16h 18m 12s

You can not chat, reply, react, or join voice channels during timeout. [Learn more](#)

How to Develop Project64 1.6 + BEWARE of [MiB/PJ64Team!](#)
Also, see [this post](#) for even more details about the extent of [MiB/PJ64Team's](#) Mean Behavior.

