Main Cheating Allegation

The runner NikocraftHD must have cheated the only two Any% runs in the channel.

1. https://www.youtube.com/watch?v=sacfl8WAyno (1.8 SSGP FWR, Historical sub4)
   Seed: 2004103968544575047

2. https://www.youtube.com/watch?v=Ay_jhHT0LUI (1.8 SSGP FWR, 2:49)
   Seed: 225874918561344128

Important Note (with the Evidences):

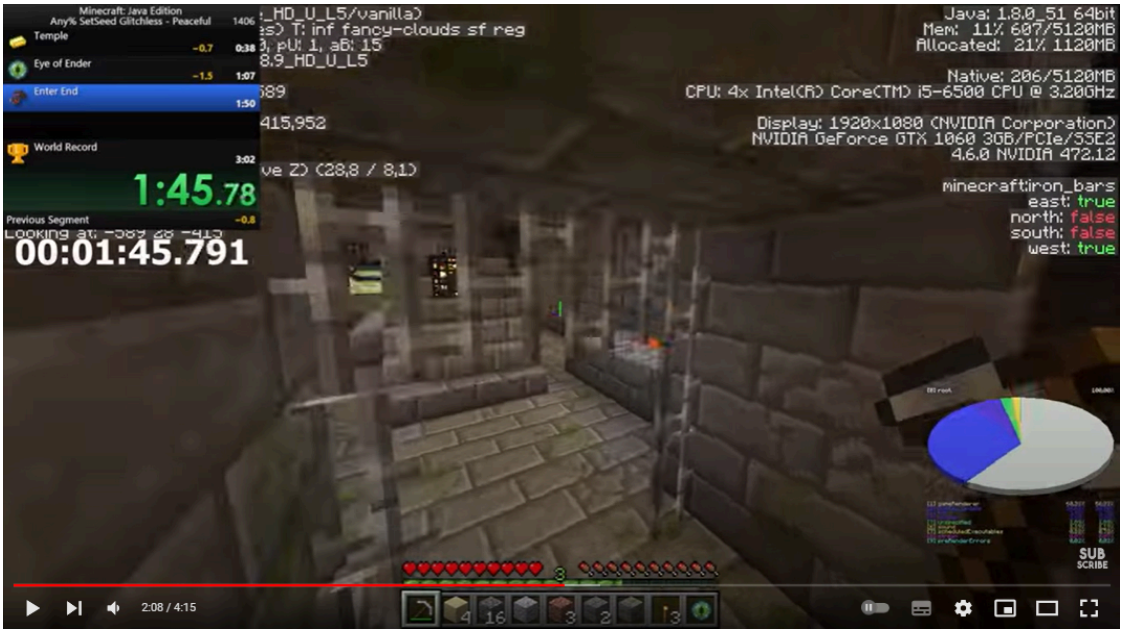
- The chance that the 1st run is cheated: Very high (99.99999999999… %)
- The chance that the 2nd run is cheated: 100%

Suspected Method of Cheating

“Splicing” using a recreated world file & using a NBT Editor to manually copy the previous world “level.dat” values for getting good End Towers.

It also appears that both runs were cheated with the exact same method & the version used seems to be 1.8.9 with OptiFine HD_U_L5. (OptiFine version is only confirmed for 2nd run)

Proof (F3 screen):



The actual end fights are probably legitimate, because the suspected cheating method gives an easier way to retry the Dragon fight, making it faster to just grind a good The End segment.

(Important Context: These runs were verified on SRC, prior to the World Files and Logs rules, so we sadly have no access to anything other than the videos)

Brief Introduction (Why did we check these runs?)

On the 3rd of March 2023, we received allegations from a member of our team, showing that NikocraftHD's runs were suspected of cheating.

These provided a very deep analysis of the Ender Dragon charges and the behavior of the last charge was very suspicious.

However, we haven't really managed to find anything conclusive enough to prove that the game had modifications on the Dragon fight, despite the last charge on the 1st run being hard to replicate.

The most important fact to note is that OptiFine was used in both the runs we mentioned before, preventing tools like Minecraft Coder Pack to be an easy option for modding the game and getting these rare charges.

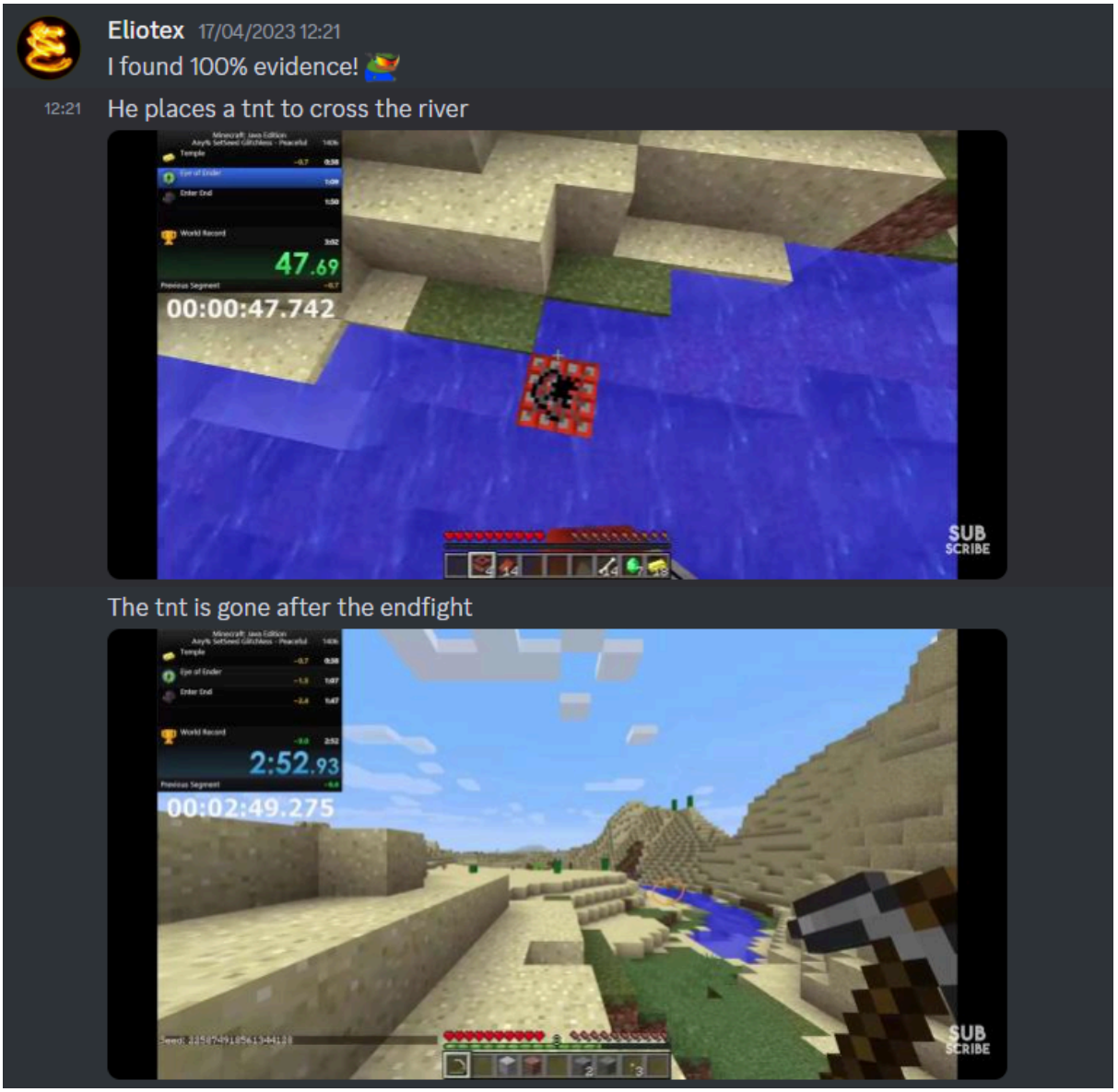
(Historical note: Legacy Fabric was **not** allowed at the time)

This made us quickly lose interest in the allegations and we were going to a point where we assumed that Niko couldn't have possibly cheated, due to the investigation showing no direct proof of attempted cheating.

At least, that's what we thought...

Evidence 1# (2nd run)

On the 17th of April, a very trusted member of our community, Eliotex, found this important evidence in the 2nd run, which immediately revived the interest in the allegations:



The evidence shows that Niko placed 2-3 TNT blocks in a location that is visible from the spawn point.

However, when Niko finished the run, the expected TNT blocks were gone!
How is that possible?

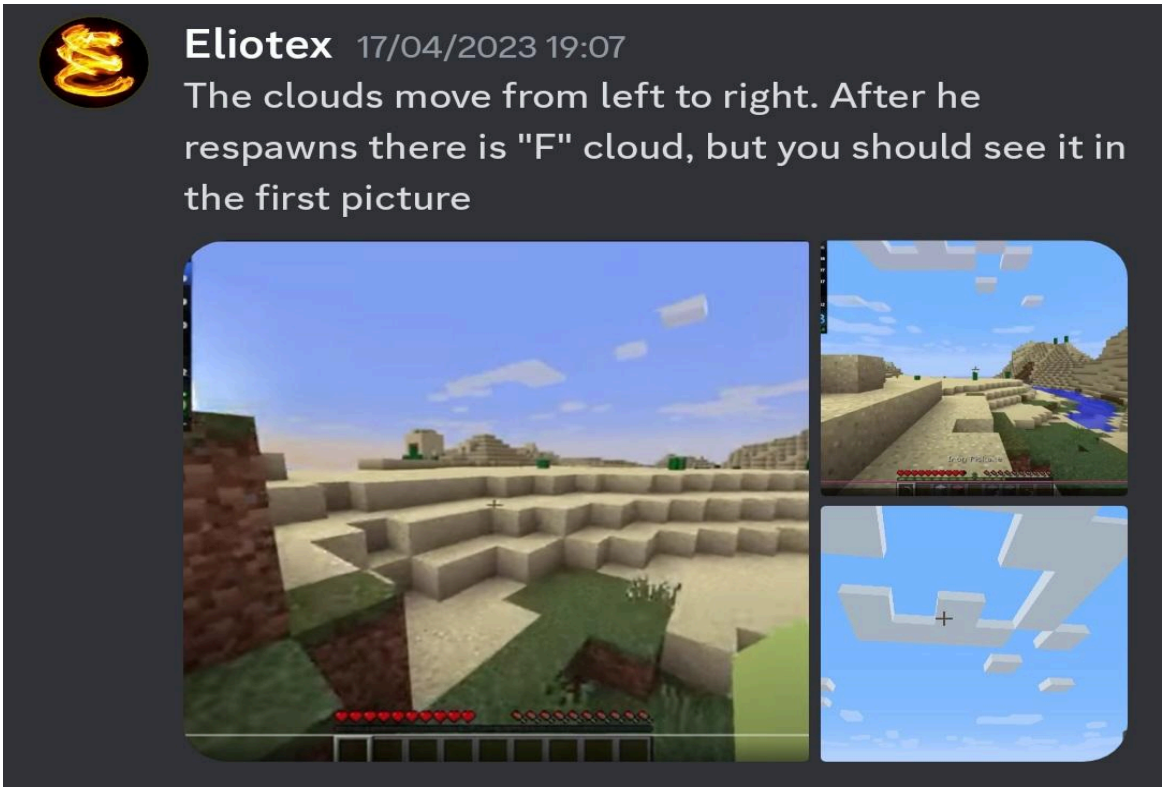
I (Crafterdark) personally theorized the possibility of a World Corruption glitch, but this is easily disproved if you consider that the autosave in these old versions only takes 45 seconds…

…but this is not the only evidence that Eliotex found in the 2nd run.

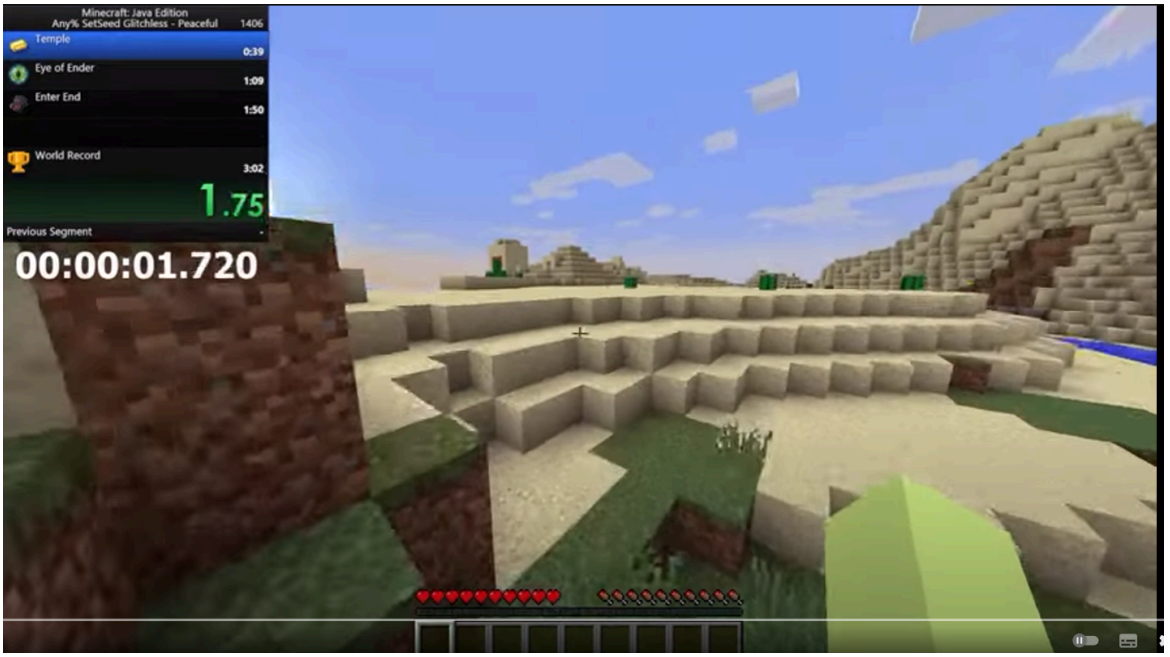
Evidence 2# (2nd run)

On the same day, Eliotex noticed something odd about the clouds in the Overworld. (Shown by Eliotex):

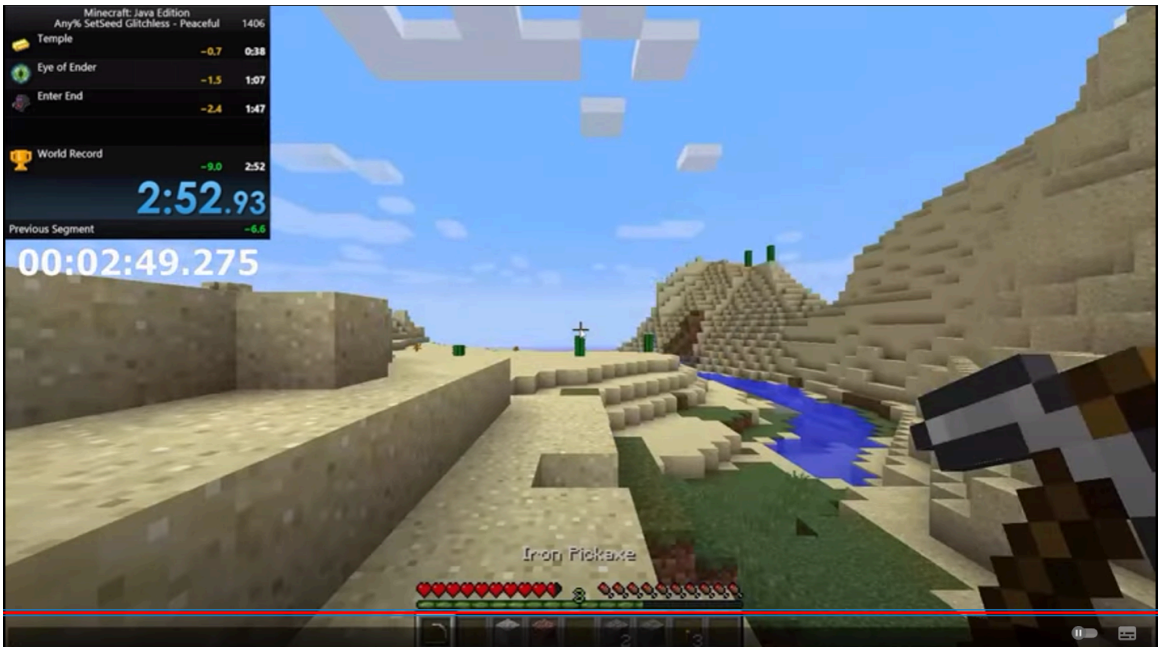
[Original message]



[Start of the run]



[End of the run]



[End of the run, different angle]

If you didn't notice, the pattern of the clouds seen at the start of the run doesn't match at all the one found at the end… this “might” look possible if you wait enough time...

Unfortunately, the clouds move very slowly and we know that ***only less than 3 minutes*** have been spent in the Client Side.

We also know that the pattern at the start of the run only appears after **approximately 30 minutes of in-game time from previous runs**, but the one at the end only shows up **during the first 5 minutes** of the start of the cloud cycle. (Since the clouds will **cycle after about 90 minutes**, there’s approximately **60 minutes of gameplay** that are lacking from this run or… it was **spliced**)

Testing commands & clouds info (If you want to verify in-game):

[Remember to reload your client instance before testing these 3 commands]

/tp -423.5 68.0 -471.5 0 0 (end run clouds 1#) [Approx. 127s from client reload]
/tp 768.5 65.0 -470.5 0 0 (start run clouds 1#) [Immediately after client reload]
/tp 2648.5 68.0 -471.5 0 0 (end run clouds 2#) [Approx. 127s from client reload]

every 3072 blocks = same clouds
cloud speed (tick): 0.03
cloud speed (second): 0.6

These clouds, consistent and moving only in the Client Side part of the game, will also keep track of their original pattern when you re-join a world/dimension.

This shows that Niko must have spliced the run after The End loading screen segment, since it’s the only viable part for a good splice & the only explanation for the over 60 minutes of gameplay that are missing.

What we knew so far (2nd run must be cheated)

Evidence 1# combined with Evidence 2# proved that NikocraftHD must have spliced the 2nd run without any doubts and, specifically, by splicing on The End segment.

Now that we were aware that the 2nd run was cheated and that unfairly stole the WR from the holder of the time (PoisonousSkely, the “beaten” wr run)… it become almost natural to check for the legitimacy of the 1st one.

Did we really lack evidence for the 1st run?

At first, you might think so, since the audio is completely absent, the part after Credits is entirely cropped and if you re-watch the 1st run before reading the incoming evidence, I bet you'll absolutely think that nothing goes wrong.

Against all odds, there's something wrong in it… that I (Crafterdark) noticed after 3 years since watching the inspiring SSGP run… the one that immediately got very popular in the minecraft speedrun community & was defended by many people including me.

Historical background for the 1st run

Back in 2018, the WR holder was SpoonTy, with a very solid time (4:46) and the seed (2004103968544575047) used was considered "unreliable", due to the insane luck required for getting single Enter End in a session of runs.

This fact combined with End Towers and the Dragon charges made setting a good time on this seed very hard, extremely challenging.
SpoonTy's time was considered very good by many members of the community and very few expected this run to be beaten soon.

After 11 months (2019), NikocraftHD showed up with the most unexpected achievement on Any% SSGP, a sub4 (3:59) on the same seed, over 47 seconds saved [!], almost perfect The End… all so good that it became a huge inspiration for many runners.

We were verifying runs with lower standards and this run was very easily deemed legitimate, because if you look at it with not very advanced game code knowledge, you'll pretty much see the same behavior as any other legitimate run.

But of course, we are here to show that there's ONE issue in the run and this small issue happens to be extremely important to prove that the chances that this run had no splice (and was legitimate) are very low… to the point where it's just almost impossible to objectively believe it…

Evidence 3# (1st run & 2nd run)

After a few minutes from Eliotex's findings, I (Crafterdark) did discover this evidence completely by accident and I immediately went checking the code…

Before going forward with the evidence, I want the reader to check the speedruns starting from the a) timestamp and ending to the b) timestamp. Just to try to see the same intuition I had that day… where we should look…?

First run:

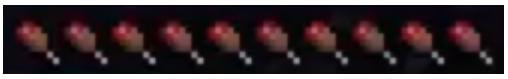
a) FORMER MINECRAFT WORLD RECORD in 2:49
b) FORMER MINECRAFT WORLD RECORD in 2:49

Second run:

a) NEW MINECRAFT WORLD RECORD! | 3:59.71 Speedrun
b) NEW MINECRAFT WORLD RECORD! | 3:59.71 Speedrun

Did you notice something?

What you are looking for will be in the page below.



That’s a Hunger Bar.

Yes.

Why would we look at this thing on Peaceful difficulty?

Let’s start with a very short technical explanation.

Hunger Bar “Movement” in the game code

Instead of showing the code, I’ll try to explain the specific parts that are relevant here.

The Hunger Bar will never deplete on Peaceful difficulty, so the Hunger Level is always 20. (10 food piece x 2)

The code specifically prevents the Hunger Level to become 19 or less on Peaceful.

The known consequence is that you’ll never be able to eat anything other than golden apples, but that’s something you probably already know if you played the game enough.

Now, what is the parameter that allows the Hunger Bar to deplete on a regular gameplay?

This parameter is called “Saturation” and the Player starts with this value being set to “5.0”

What you expect is that on Peaceful difficulty, both Saturation and Hunger Level are restored, but… that’s not what happens and I can easily show you.

Remember the previous links, keep watching Niko’s Hunger Bar from this part of the 1st run:

NEW MINECRAFT WORLD RECORD! | 3:59.71 Speedrun

You’ll notice that the Hunger Bar pieces are doing “Random movements” (“Up” OR “Down” OR “No movement”) every 3 seconds or so.

Is this just a funny random coincidence? No.

Saturation does deplete on Peaceful difficulty and it won’t get REFILLED (!)
[Unless you eat golden apples or respawn]

This is even true on Release 1.19.4 (The current latest release), except that Saturation can be obtained in more ways.

The **Client Side** is also coded to trigger a “Random Hunger Bar Movement” when Saturation is less or equal to “0.0” & after a specific timer has elapsed.

This repeating timer is also not random, it will follow the formula:

- **“3 * Hunger Level + 1” ticks**

If you followed this explanation until here, you’ll already know that the cycle of the timer will be set to:

“3 * 20 + 1” ticks OR 61 ticks OR 3.050 seconds

So now we know that when the player loses all the Saturation on Peaceful, then the Random Hunger Bar Movement will:

1) Trigger every 3.050 seconds
2) 10 Food Pieces will be visually “Up” OR “Down” OR “No movement” (RNG)
3) Persist on dimension travel, reloading the world, ecc. (Saturation is stored on level.dat)

So…

A) NikocraftHD had the first Random Hunger Bar movements in these 2 specific timestamps in the two runs:
   - NEW MINECRAFT WORLD RECORD! | 3:59.71 Speedrun
   - FORMER MINECRAFT WORLD RECORD in 2:49

B) What you expect from these timestamps and on is that, barring potential frameskip (unlikely in both runs, due to high recording FPS), the runs will have regular 3.050 seconds of Random Hunger Bar movements and this is mostly true…

C) Unfortunately… this is not the case!

If you look at this other spreadsheet that I (Crafterdark) created, you can see that there’s a very specific range where the Hunger Bar didn’t move. I already showed you this range before, but I decided to be even more fair and I tried to compare this with other existing runners… because you never know, I could have been simply wrong…

Hunger Bar ANTI-CHEAT Real Data From Runs (Note: This is just supporting evidence, I could have definitely gathered more data)

This confirms the code I’ve analyzed and it also shows that the behavior on NikocraftHD's runs is apparently impossible by the game code (!!!)

In BOTH runs.

What if NikocraftHD experienced the extremely odd event… twice? [Legitimately]

Let's try to assume that what we saw in those two runs is … somewhat… "plausible" without cheating.

Theory A [The runner is legit]: The game code simply worked as intended!

1) The game code does actually "allow" for the Hunger Bar to stay still, but only if all 10 food pieces do "nothing". (Since that's what you see visually)

2) The probability for a food piece to stay still is 0.333.. (⅓)

3) Since each of these 10 pieces should have not jumped 10 times (about 34 seconds) in the 1st run, there's a probability of "(⅓) to the power of 100", approximately 1.94E-48, that this event occurred on the 1st run.

4) Let's not forget that the 2nd run should not have had 7 jumps (23 seconds), with a final probability of 3.9E-34.

5) Both events happened on a WR run too.

6) It's clearly absurd to think that this is what truly happened… due to the extremely low chances. (12 eyes portals are about 1E-12)

Theory B [The runner is legit]: The video skipped all frames where the Hunger Bar jumped for a tick!

1) Even if this theory was included here, thinking that in both runs, on Enter End and up to 23+ seconds, this issue happened is just… incredible to think of.

2) There's evidence starting from here, that the framerate is very good, so we can definitely exclude this idea.
   NEW MINECRAFT WORLD RECORD! | 3:59.71 Speedrun

Theory C [The runner is legit]: The Hunger Bar simply lagged for 23+ seconds.

1) The Hunger Bar runs Client Side and doesn't follow any delay of the Server, especially after being set by level.dat on End load.

2) This means that the 3.050 seconds depends purely on the fps and framerate of the client. (Checking my previous spreadsheet easily proves this statement)

Theory D [The runner is legit]: The runner used an unknown modification that changed the game! (There are rumors that the version was “scuffed”)

1) I (Crafterdark) did originally think of this, but if you see carefully… the Main Menu… is visible in both runs and Forge was not used! (No label + 2nd run clearly doesn’t have it)

2) Since the runner did not use Forge and we know that OptiFine is used in both runs… it’s impossible to think of a way that is legitimate to add a modification.

   (Reminder: **Fabric** was **not** used at the time and **Forge** was **necessary** for using OptiFine with other mods)

So really, if we try to think of a reason this behavior happened without cheating, we are really just stuck…

We then thought, if NikocraftHD cheated, then **HOW** did he cheat.

What if NikocraftHD experienced the extremely odd event… twice? [Cheating]

Theory A [The runner is NOT legit]: The runner just did a backup of the newly created world after entering The End, re-opened it and spliced the videos together.

1) This might look it, but unfortunately there are several problems that go against this simple theory.

2) Niko needed good End Towers, but this method only allows you to re-use the same End over and over.

3) The runner also had to simulate an Enter End, which is only possible by dying, thus losing the xp and items.

4) Saturation, like previously explained, is stored in level.dat (Tested) and if you backup the world and re-open it, the Random Hunger Bar movement will still happen every 3.050 seconds … !

5) So incredibly.. even this theory looks wrong!

We thought this was gonna be impossible to figure out, until we think of this…

Theory B [The runner is NOT legit]: The runner did a backup of the Overworld segment world and, after entering The End, recreated a newly fresh world with the same seed, only to be used for The End segment and applied NBT values from the previous world to the new one.
Then spliced everything together.

1) This is plausible, because Niko needed good End Towers, so he needed a new world where he could enter repeatedly The End to generate a good Tower preset.

   (Reminder: we are assuming that he couldn't have used mods other than OptiFine)

2) Niko could have used very known NBT editors (to name one that is well known since 2011, NBTExplorer) to bring back the items and xp bar + level from the previous Overworld segment, without the requirement of changing his coordinate.

3) This would effectively allow the perfect seamless splice done on The End segment, which would remain, in fact, very well done, BUT … this is exactly one of the possible explanations of Evidence 3# (below)

4) Since the Random Hunger Bar Movement mechanics were ignored by pretty much everyone (including me), it's plausible that Niko only copied the items (1), the xp bar value (2), the level value (3) [Inventory is never shown, so we'll never know] and forgot to copy the most important value in level.dat for avoiding to get caught… Saturation! Then the original Saturation of 5.0 was used, instead of the one stored in the original world (0.0).

5) After the previous idea [Entering End with Saturation 5.0], I (Crafterdark) decided to manually check with a simulated The End of Niko's 1st run, if it could have been a plausible explanation and I did manage to replicate the odd event myself in this video. I did also manage to replicate the 2nd run odd event in this other video.

   (Note: The movements are very off, but the damage taken were calculated to allow for a similar Random Hunger Bar Movement start)

6) Unsurprisingly, the respawn / new world Saturation almost perfectly matched the impossible behavior of the 1st run, further proving that Niko must have had Saturation 5.0 from a newly created world OR from respawn JUST before entering The End in BOTH runs… thus making this the **only known theory** that can simulate the odd events.

Extra evidences regarding the runs

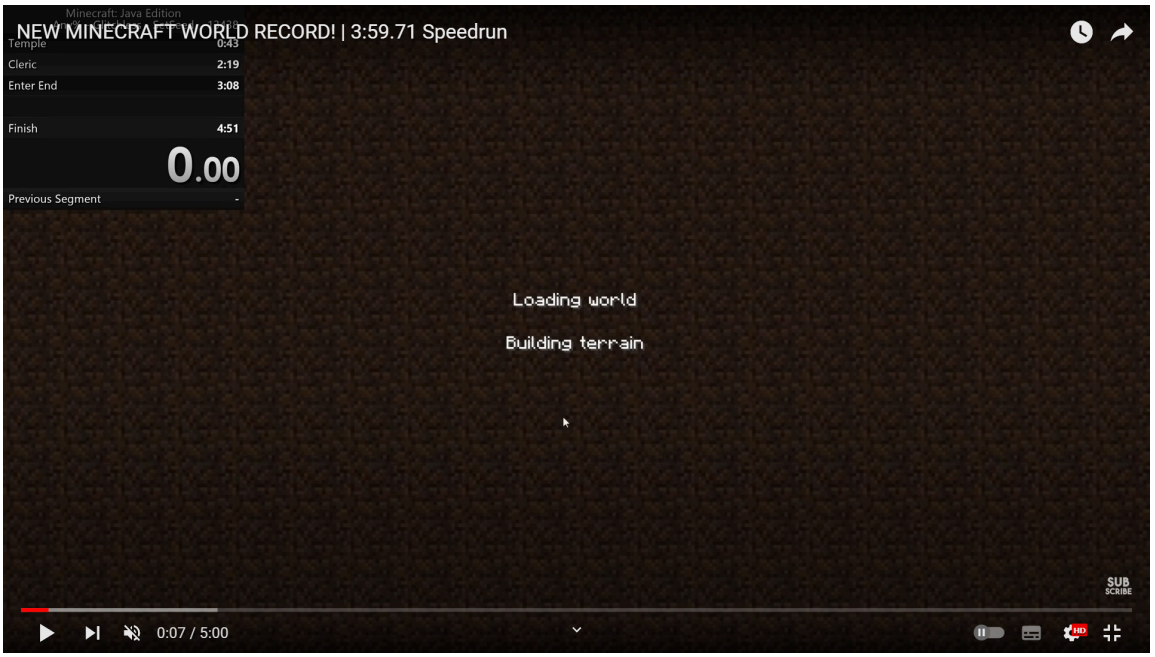
Important Note:

These evidences aren't enough to prove that the runs are cheated in any way.

They might be inaccurate, as the available data comes from bad quality of the videos…

Visible New World Generation “Potential” Splice (1st run)

[Note: This evidence was found by many members of our community]



(Difference is a single frame, unlikely to be just lag, but **possible**)

Theory B [Cheating] would change if this **extra** splice was real:

1) There would have been an **indefinite** amount of time to splice and **complete** the Overworld segment and The End segment. (Very convenient)

2) He only had to show in the world selection screen that the 1st run was done after 24th, by having several worlds with date *around* the sharing of the run (26th).

Delay / Inconsistency with the Played Date (1st & 2nd run)

[1st run]

Submitted the 1st run twice on SRC (It was shared to the public on 2019-04-26 on YT):

1) First submission :
   - Played Date 2019-04-26 at 14:00
   - Submitted Date 2019-04-26 at 20:11

2) Second submission
   - Played Date 2019-04-27 at 14:00
   - Submitted Date 2019-04-27 at 03:53

[2nd run]

Submitted the 2nd run once on SRC (It was shared to the public on 2022-03-26 on YT):

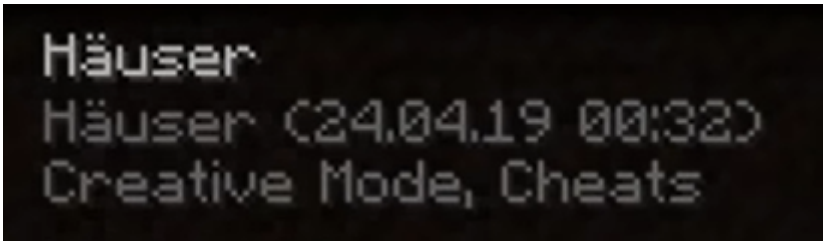
1) First submission :
   - Played Date 2022-03-26 at 13:00
   - Submitted Date 2022-03-26 at 03:21

Note: Previous data might not be easy to verify due to a possible SRC bug affecting older timestamps, anyway it was included to show that there was a supposed Played Date…

- In the 2nd run, we already know the **real** Played Date and it must be 2022-02-05 after 19:51, the proof being a previous attempt world date (**100% visible** after the fade in) seen on this timestamp… meaning that NikocraftHD shared the 2nd SSGP WR on Youtube **after about 2 months from doing the run (!)**

Anyway, let's also remember that we have Evidence 1# and Evidence 2# proving that the 2nd run is necessarily spliced, so it isn't that surprising to see that the Overworld was done that early on…

- In the 1st run, we can also see a date from a world, which gives us the minimum time for the run being started.



This is a random world that Niko used in the past and that we don't have any information about.

So, for now, we know the run necessarily happened between 2019-04-24 and 2019-04-26, but expected to be done on 26th, since the Played Date was manually placed.

Now…

The 1st run doesn't just start with a fade in, but it also included a very important frame where Niko deletes the previous attempt*.

*This last statement can't be proven, only reasonable to think of.

However, if this is not the previous attempt, then it would be very odd… (First try? WR right after deleting all runs in a session? Weird)

Anyway, the quality of the frame is terrible, but what we can get with editing is this:

[Closer]

It's reasonable to see a **(24),** it can't be a (25, 26) & it can't be (27, 28, 29) , because otherwise it would be in the future.
It can't be 20, 21, 22, 23, because we know that there's a world with **24th** as the previous latest date.

Now if we remember the previous theories, Theory B [Cheating] for 1st run required a splice and several attempts to get The End part done well.

If only the Overworld segment was spliced (*excluding* the previous New World generation potential splice), Niko **had at least 2 real days** to complete The End segment with the chosen End Towers layout… so the theory would remain plausible.

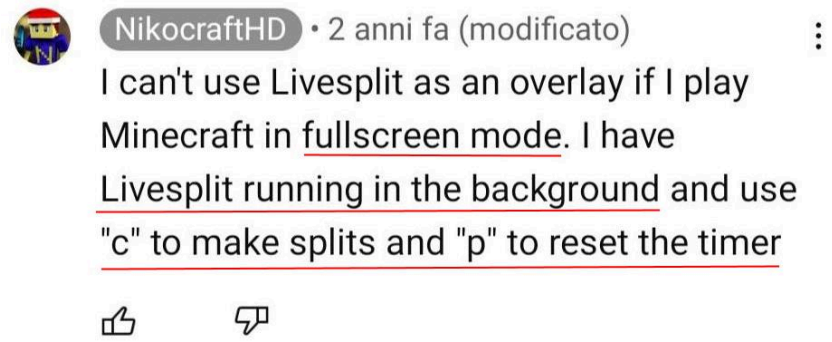
If this number was **(26)**, then the splice couldn't have happened easily, because Niko only had **less than 14 hours to do all the cheating** and he probably had to sleep too.

(He couldn't have done The End splice *easily* without doing the Overworld first, because the xp & level would be difficult to recreate in the Overworld & other RNG stuff)

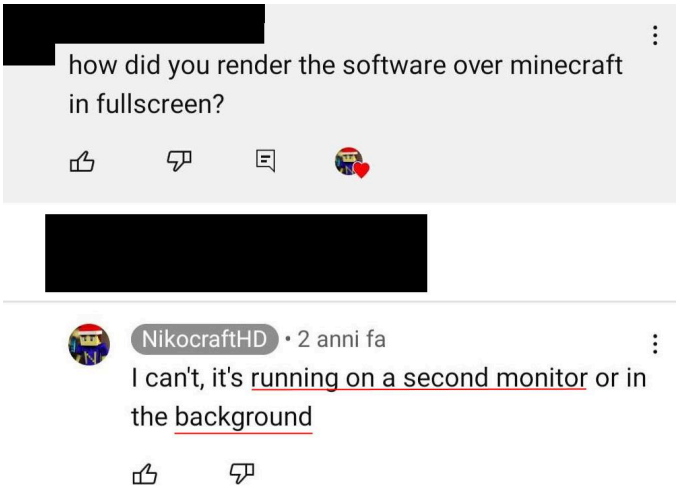
Livesplit Confusion & Fade Out (1st run)

NikocraftHD has claimed (in various messages) that the game was recorded Full Screen. He also claimed that Livesplit was recorded on the top left, so it was not recorded separately at the end, but during the run. (No post-editing was allegedly done for it)

This is a comment where he explains the keybindings used during the run:

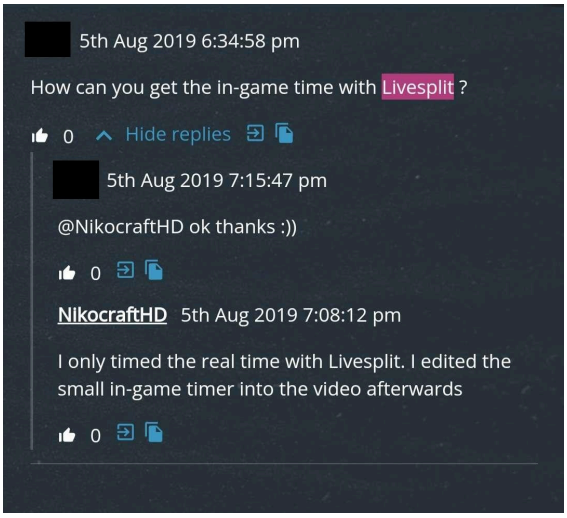


Actually quite reasonable, except that Livesplit *couldn't be visible in the background (!)* if the game was played Full Screen. Then another viewer of the run did ask:



We can see that there was a newly introduced detail (second monitor), probably because just saying "in the background" would make it impossible (and in fact, it would be).

There's also evidence that the small in-game time was edited in, in 2 different comments (this is obvious, but there's more to it):



The last comment is somewhat contradictory with the previous knowledge, but the previous comment to it … must be definitely more accurate, since it's about 4 months after sharing the run...

This gave me suspicion & several info:

1) If you remember again Theory B [Cheating], the run is spliced and split in (at least) two parts, so Livesplit could have **NOT** been recorded in a single take.

2) But then… the Livesplit overlay shouldn't be consistent (unless edited very well in) with the video at all, assuming it was really added **after the Overworld segment and The End segment were both created**. (Covering the splice)

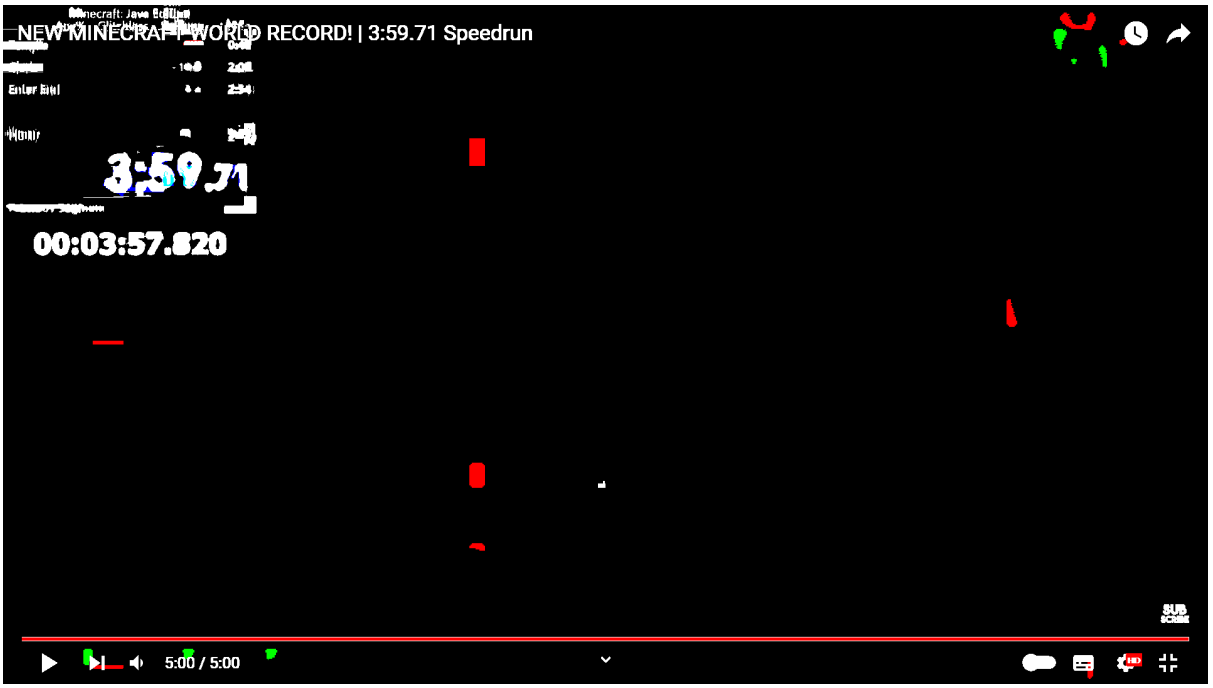
3) We know the small in-game timer was added later, so it is expected to have potential editing differences from the main video.

Finally I decided to look at a part of the run that many people probably missed, the very end…

There, we can see that the credits screen is simply fading out.

Everything looks normal at first, but then…

You can notice this frame at the very end [Edited to highlight the remaining fading out colors] (300.356 on YT):



What is wrong?

If Livesplit was recorded with the original run, then how did **its overlay resist the fade out** when the **credits** and the **mouse pointer** are **already gone**? Special fade out? The starting frames of the video *did not have* this behavior.

Also interestingly, why does the *small in-game time have a similar fade out delay to the Livesplit overla*y, if only the small in-game time was edited in?

… I have no answer for this, but you can clearly see that this would still support the Theory B [Cheating], where Livesplit was added **later** to cover a potential **splice** …