GeyserMC Test Report

Version History

V.	Implemented by	Revision date	Approved by	Approval date	Reason
1.0	J.R. and K.G.	27.11.20	J.R. and K.G.	27.11.20	Check missing particles
2.0	J.R. and K.G.	10.12.20	J.R. and K.G.	12.12.20	Utilities for next tests
3.0	J.R. and K.G	19.12.20	J.R. and K.G	19.12.20	Check application integrations
4.0	J.R. and K.G	11.01.21	J.R. and K.G	12.01.21	Check application performance
final	J.R. and K.G	14.01.21	J.R. and K.G	14.01.21	Refactoring

Shortcuts

J.R. - Jędrzej Racibor

K.G. - Konrad Gabrukiewicz

V. - version

TABLE OF CONTENTS

1. *Introduction*

This document is the software test report of the GeyserMC application. It contains the results of tests, which were executed during the whole testing phases.

1.1. Purposes:

- **Check the correctness of the translation.**
- **Find missing particles.**
- **Find missing sounds.**
- **Check modules integration.**
- **Check the speed of packet translation.**
- **Check the application under load.**
- **Check the application efficient in normal work.**

1.2. System environment:

Device specifications: Intel Core i5-9600KF 4.3GHz, 32GB RAM DDR4 3200MHz CL16.

Operation systems: macOS 10.15 with JRE 11, Windows 10 PRO with JRE 14.

Minecraft Java Edition Server protocol in version 1.16.4.

Minecraft Bedrock Edition Server protocol in version 422(1.16.200).

Minecraft Bedrock Edition Client protocol in version 422(1.16.200).

Geyser Connector in version 1.20.

2. *Overview of Tests Results.*

2.1 Test summary:

- SoundMappingCoverage - In these tests we wanted to check if some sounds are missing.
- EffectMappingCoverage - In these tests we wanted to check if some effects particles are missing.
- ConnectorIntegrationTest – In these tests we wanted to check if Geyser (bridge between Bedrock Client and Java Server) works properly.
- Code review - In these tests we wanted to check if some unexpected things in coding exists.
- Gameplay tests using Geyser - General gameplay tests to check the operation of the application.

2.2 Overall assessment of tests:

- SoundMappingCoverage – passed.
- EffectMappingCoverage – failed.
- ConnectorIntegrationTest – passed.
- Code review - passed (only one unexpected thing in code found).
- Gameplay tests using Geyser – passed.

3. Testing details.

3.1. SoundMappingCoverage

- We iterate through every value of BuiltinSound enum and verify that SoundRegistry contains translated to Bedrock value.

3.1.1. Result:

Geyser pass this test, all sounds are mapped.

3.2. EffectMappingCoverage

- We iterate through every value of ParticleType enum and verify that EffectRegistry contains translated to Bedrock value.

3.2.1. Result:

Geyser did not pass this test. We found and report missing particles. (5.4.).

3.3. ConnectorIntegrationTest-

Utilities for integration tests:

- IntegrationClientPacketHandler - class responsible for capturing expected incoming packet to the Bedrock Client.
- IntegrationServerAdapter – class responsible for capturing expected incoming packet to the Java Server.
- TestConfiguration – mocked Geyser connector configuration.
- TestLogger – mocked Geyser logging class.
- IntegrationConnectorEventHandler – mocked Geyser connector server event handler.
- TestHelper - functions to create java server, bedrock client and geyser connector.

3.3.1. passFromClientToServer

- Using steveice10.packetlib we started java server.
- Using nukkitx.protocol.bedrock.BedrockClient we created Bedrock Client.
- We created Geyser Session.
- We connected Bedrock Client to Java Server.
- We sent packet from Bedrock Client to Java Server via Geyser.
- We checked if packet is received.

3.3.2. passFromServerToClient

- Using steveice10.packetlib we started java server.
- Using nukkitx.protocol.bedrock.BedrockClient we created Bedrock Client.
- We created Geyser Session.
- We connected Bedrock Client to Java Server.
- We sent packet from Java Server to Bedrock Client via Geyser.
- We checked if packet is received.

3.3.2 pingPassthrough

- Using steveice10.packetlib we started java server.
- Using nukkitx.protocol.bedrock.BedrockClient we created Bedrock Client.
- We created Geyser Session.
- We pinged Geyser instance and checked if we got informations about Java server.

3.3.3. Results

3.3.3.1. PassFromClientToServer - passed, packet received as expected.

3.3.3.2. PassFromServerToClient - passed, packet received as expected.

3.3.3.3. PingPassthrough - passed, we received expected information in pong.

3.4. *PerformanceTest*

Utilities for performance tests:

- SpigotRunnable – runs spigot Minecraft: Java Edition server that is needed to capture game session packets.
- SetUp - runs SpigotRunnable and Geyser instance, captures and saves packets recived from Minecraft Bedrock client.
- RandomJoinTestClientRunnable - runnable used in thread to connect Bedrock Client to Geyser instance and send the test packet set once.
- UnderLoadTestClientRunnable - runnable used in thread to connect Bedrock Client to Geyser instance and send the test packet set multiple times.
- PerformanceConnectorEventHandler - mocked Geyser connector server event handler.
- PerformanceServerEventHandler - class responsible for setting correct PacketHandler for incoming connection.
- PerformanceServerPacketHandler - class responsible for capturing expected incoming packet to the Bedrock Server.
- PerformanceServerAdapter/UnderLoadServerAdapter - class responsible for responding on incoming packets to the Java Server.

3.4.1. Direct connection vs connection using Geyser

- Using setUp we received the test packet set from game session.
- We created Bedrock Client and Bedrock Server.
- We connected Bedrock Client to Bedrock Server.
- We measured the time taken to pass the test packet set.
- We created Bedrock Client, Geyser Instance and Java Server.
- We connected Bedrock Client to Java Server.
- We sent the test packet set from Bedrock Client to Java Server via Geyser.
- We measured the time taken to pass the test packet set.
- We repeated this process one hundred times and compared the results.

3.4.2. Geyser efficiency under load

- Using setUp we received the test packet set from game session.
- We created {20,30,40,50,80,100} Bedrock Client, Geyser Session and Java Server.
- We connected {20,30,40,50,80,100} clients to Java Server.
- We sent the test packet set from Bedrock Clients to Java Server via Geyser.
- We measured the time taken to pass the test packet set.
- We repeated this process one hundred times and compared the results.

3.4.3. Geyser efficiency during normal work

- Using setUp we received the test packet set from game session.
- We created {20,30,40,50,80,100,120,160,200} Bedrock Client, Geyser Session and Java Server.
- We connected {20,30,40,50,80,100,120,160,200} clients to Java Server in random time.
- We sent the test packet set from Bedrock Clients to Java Server via Geyser.
- We measured the time taken to pass the test packet set.
- We repeated this process one hundred times and compared the results.

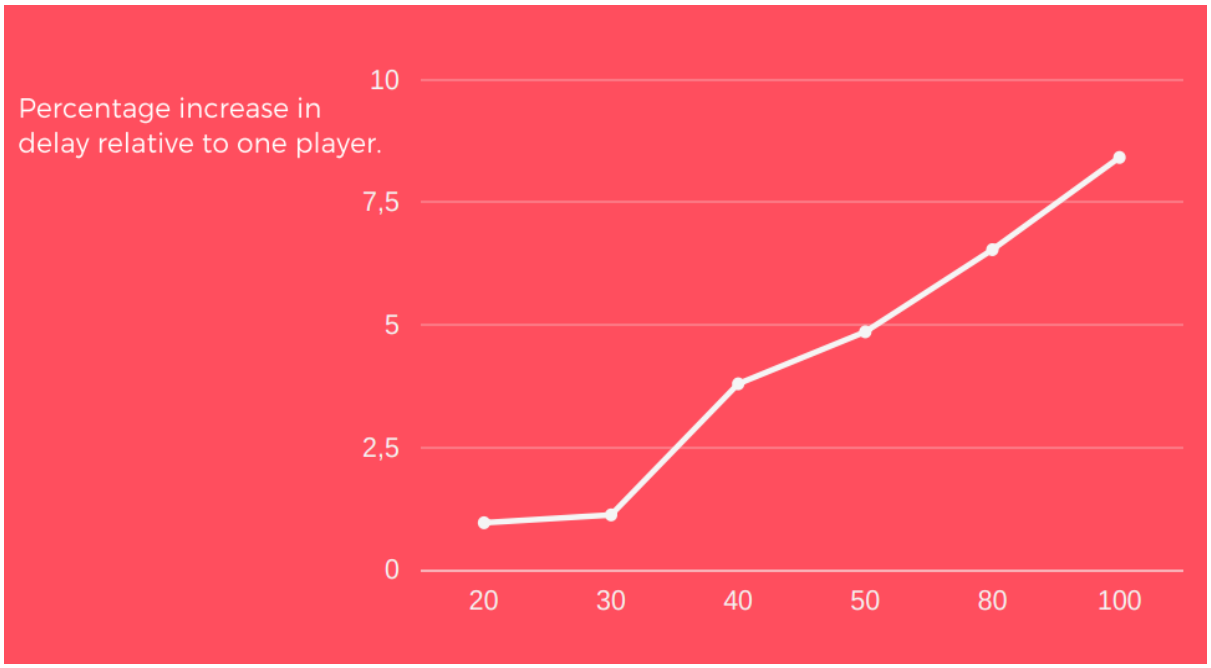
3.4.2. Results:

3.4.2.1. Direct connection vs connection using Geyser

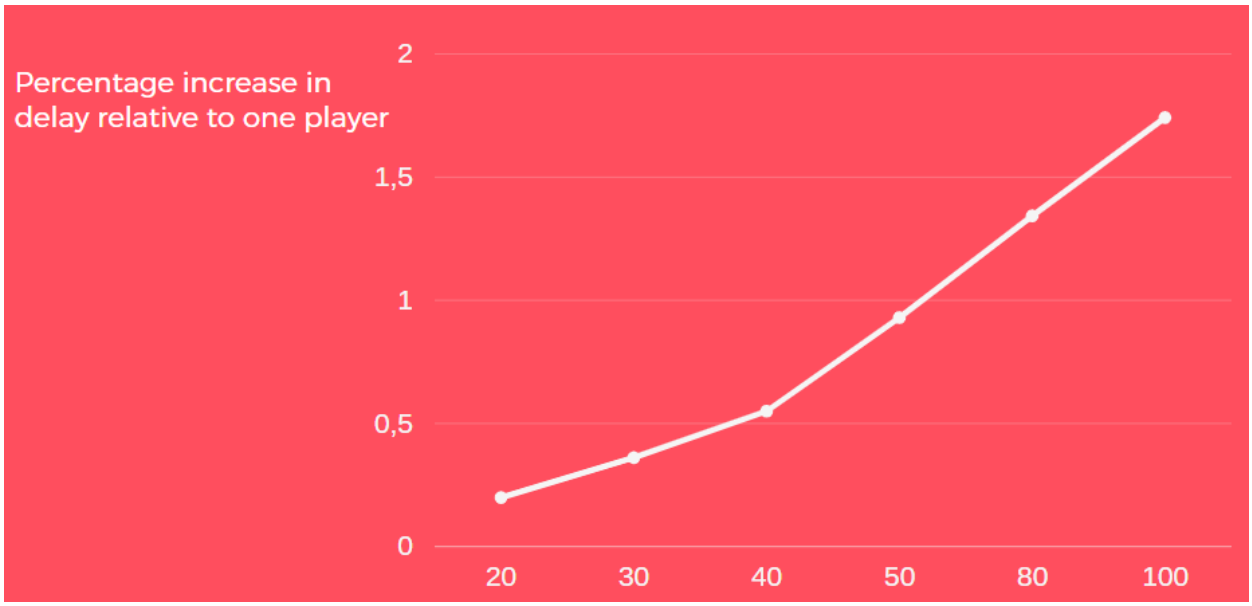
The test was a total success. Geyser shows a latency of less than one percent (0.18%) relative to the direct connection.

3.4.2.2. Geyser efficiency under load

By running this test on macOS, we obtained the data that is illustrated in the graph below. We see a slight increase in the load on Geysers relative to the range [0,30]. In the range [30,40] we can see a significant decrease in efficiency, which increases up to ~ 8.42% for 100 players.

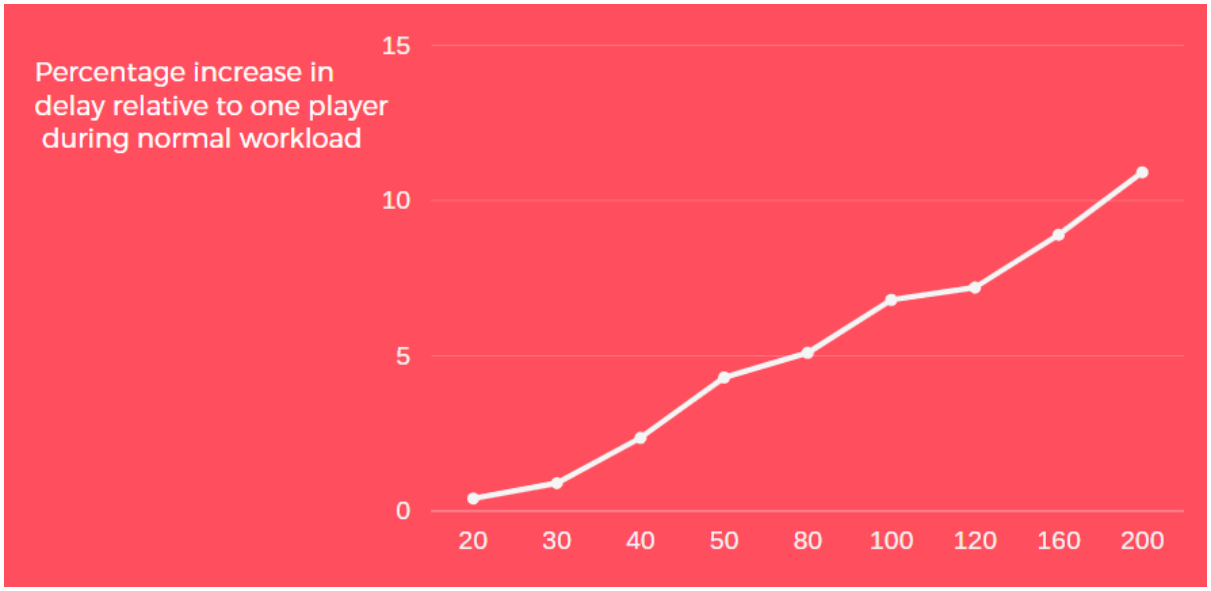


By running this test on Windows 10, we obtained the data that is illustrated in the graph below. We see a slight increase in the load on Geysers relative to the range [0,30]. In the range [40,50] we can see a significant decrease in efficiency, which increases up to ~ 1.74% for 100 players.

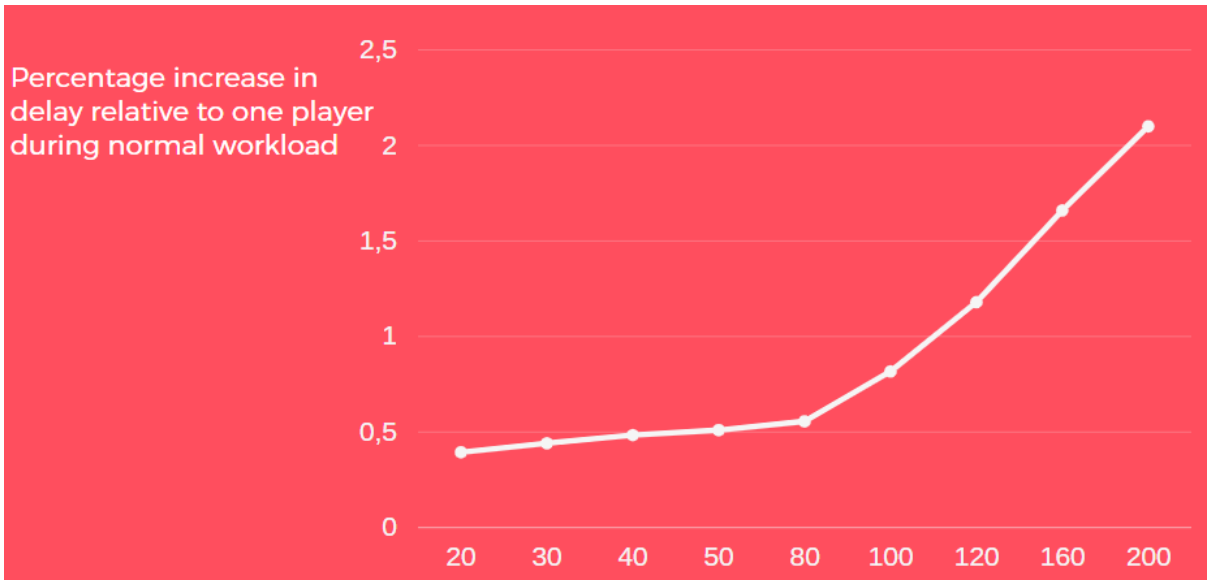


3.4.2.3. Geyser efficiency during normal workload

By running this test on macOS, we obtained the data that is illustrated in the graph below. We see a slight increase in the load on Geysers relative to the range [0,30]. In the range [30,40] we can see a significant decrease in efficiency, which increases up to ~ 10.91% for 200 players.



By running this test on Windows 10, we obtained the data that is illustrated in the graph below. We see a slight increase in the load on Geysers relative to the range [0,30]. In the range [80,100] we can see a significant decrease in efficiency, which increases up to ~ 2.08% for 200 players.



3.5. Gameplay tests using Geyser

3.5.1. Results

During normal use of the application, we encountered one bug (5.6).

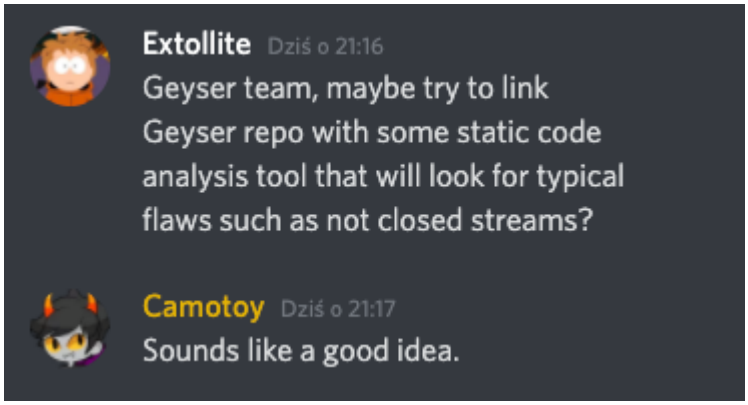
3.6. Code Review

We noticed two main problems with code.

- a lot of streams are not closed. (5.3., 5.5.)
- where it is not required for the proper operation of the application often the operating logic is not followed. (5.2.)

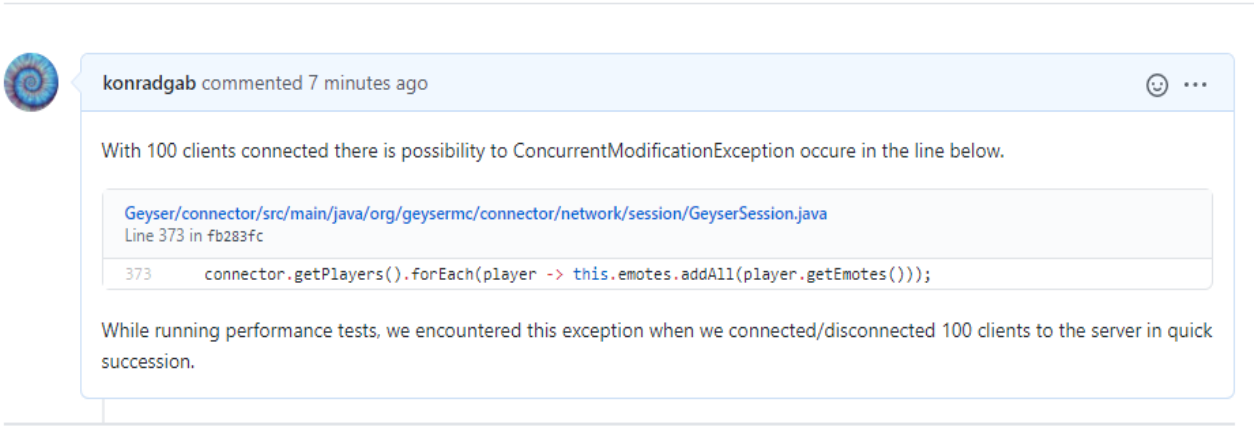
4. Suggested Actions

Code is not fully synchronized. Some shared resources are not properly synchronized what came out during our tests. As a result, in some critical situations code throws an exception (5.1.). We discovered that the code has many minor bugs (5.3., 5.5.). We corrected some of the issues that prevented us from running tests and proposed a way to resolve minor problems existing in code.



5. Bugs reported

5.1.



https://github.com/GeyserMC/Geyser/issues/1830

5.2.

konradgab commented 9 days ago

Hello i found something strange in code.
In the score class, when using the setTeam method, change updateType to UPDATE, but do not set a timestamp.
When the setTeam method is used there is a double state change with one timestamp - setTeam changes state to update and then setUpdateType changes state to Add.

Geyser/connector/src/main/java/org/geysermc/connector/scoreboard/Score.java
Line 75 in 1a08e11

```
75      currentData.updateType = UpdateType.UPDATE;
```

Geyser/connector/src/main/java/org/geysermc/connector/scoreboard/Score.java
Line 82 in 1a08e11

```
82      currentData.updateType = UpdateType.UPDATE;
```

Double status change with one timestamp-

Geyser/connector/src/main/java/org/geysermc/connector/scoreboard/Objective.java
Line 105 in 1a08e11

```
105     .setTeam(scoreboard.getTeamFor(id))
```

Geyser/connector/src/main/java/org/geysermc/connector/scoreboard/Objective.java
Line 106 in 1a08e11

```
106     .setUpdateType(UpdateType.ADD);
```

https://github.com/GeyserMC/Geyser/issues/1788

5.3.

Close en_us.hash stream #1833

Merged Camotoy merged 1 commit into GeyserMC:master from Extollite:close-stream-for-hashing 2 hours ago

Conversation 0 | Commits 1 | Checks 1 | Files changed 1

Extollite commented 2 hours ago — Contributor

No description provided.

Close en_us.hash stream — b9273a5

Camotoy approved these changes 2 hours ago — View changes

Camotoy merged commit c0a6465 into GeyserMC:master 2 hours ago — View details — Revert
1 check passed

https://github.com/GeyserMC/Geyser/pull/1833

5.4.

Extollite commented 5 days ago • edited — Contributor

Listed particles are missing mappings in Geyser:

- `soul_flame`
- `falling_obsidian_tear`
- `soul`
- `flash`
- `falling_nectar`
- `falling_honey`

According to java wiki corresponding particles are:

- `minecraft:blue_flame_particle`
- `obsidian_tear_particle`
- `minecraft:soul_particle`
- `minecraft:flash`
- `minecraft:nectar_drip_particle`
- `minecraft:honey_drip_particle`

https://github.com/GeyserMC/Geyser/issues/1817

5.5.

Close locale streams #1832

Merged — **Camotoy** merged 2 commits into `GeyserMC:master` from `Extollite:stream-closing` 2 hours ago

Conversation 2 | Commits 2 | Checks 1 | Files changed 2

Extollite commented 3 hours ago — Contributor

It is part of unclosed streams that for multiple clients stacks and hold file descriptors. Parts of code that uses InputStreams need refactor to close streams.

`Close locale streams` — `7ecd5bf`

New changes since you last viewed — View changes

`Fix formatting` — `3621084`

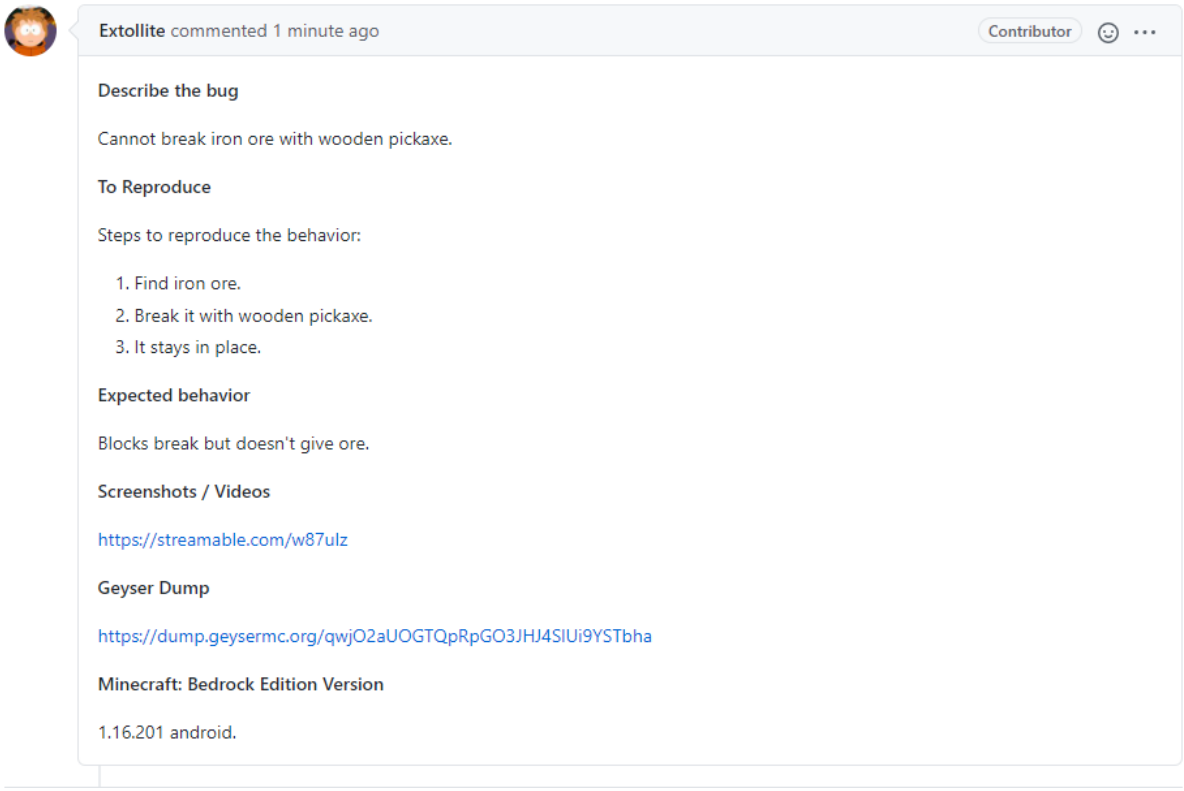
rtm516 approved these changes 2 hours ago — View changes

Camotoy commented 2 hours ago — Member

Thanks!

https://github.com/GeyserMC/Geyser/pull/1832

5.6.



Extollite commented 1 minute ago Contributor

Describe the bug

Cannot break iron ore with wooden pickaxe.

To Reproduce

Steps to reproduce the behavior:

1. Find iron ore.
2. Break it with wooden pickaxe.
3. It stays in place.

Expected behavior

Blocks break but doesn't give ore.

Screenshots / Videos

https://streamable.com/w87ulz

Geyser Dump

https://dump.geysermc.org/qwjO2aUOGTQpRpGO3JHJ4SlUi9YSTbha

Minecraft: Bedrock Edition Version

1.16.201 android.

https://github.com/GeyserMC/Geyser/issues/1837

6. Test & Test Cases

Test Cases and summary tests:
https://jira.frege.ii.uj.edu.pl/projects/GEYS

Tests:
https://github.com/Extollite/Geyser/tree/jr-tests/test
https://github.com/Extollite/Geyser/tree/jr-tests/connector/src/main/java/org/geysermc/connector