

**IN THE UNITED STATES DISTRICT COURT  
FOR THE WESTERN DISTRICT OF TEXAS  
WACO DIVISION**

WORLDS INC.,

Plaintiff,

v.

MICROSOFT CORPORATION;

Defendant.

CIVIL ACTION NO. 6:20-cv-872-ADA

**PLAINTIFF WORLDS INC.'S PRELIMINARY INFRINGEMENT CONTENTIONS**

Plaintiff Worlds Inc. (“Worlds”) makes the following disclosures pursuant to the Court’s Order Governing Proceedings – Patent Case:

1. Attached hereto is Worlds’ preliminary infringement contention claim charts for U.S. Patent No. 8,082,501 (“the ’501 Patent”) being asserted against Defendant Microsoft Corporation and its Minecraft products;

2. As described on the face of the ’501 Patent being asserted in this action, Worlds claims the priority date of no later than November 13, 1995 for each asserted claim. This priority date corresponds to the filing date for U.S. Provisional Patent Application No. 60/020,296, filed on November 13, 1995. According to the face of the ’501 Patent and col. 1, lines 7-21, the ’501 Patent’s priority chain extends back to U.S. Provisional Patent Application No. 60/020,296, filed on November 13, 1995, and the disclosures of this provisional patent application and all intervening patent applications are incorporated into the ’501 Patent by reference. Each claimed invention in the ’501 Patent was fully conceived no later than November 13, 1995, and reduced to practice no later than November 13, 1995, the filing date of

the above-referenced U.S. Provisional Patent Application No. 60/020,296, which shares a common specification with the '501 Patent;

3. As required, a copy of the '501 Patent's prosecution file history and documents evidencing conception and reduction to practice for each asserted claim (*see* U.S. Provisional Patent Application No. 60/020,296 at WDDD-MS\_0001863-1907), are being provided to Defendant Microsoft on this date. Also provided are copies of the file history from IPR2015-01319 and copies of consolidated filings made in IPR2015-01264 relevant to the remand and termination of IPR2015-01319.

DATED: December 21, 2020

/s/ Wayne M. Helge

Wayne M. Helge

Alan A. Wright

Donald L. Jackson

James T. Wilson

DAVIDSON BERQUIST JACKSON & GOWDEY, LLP

8300 Greensboro Drive, Suite 500

McLean, VA 22102

Tel: (571) 765-7700

whelge@dbjg.com

awright@dbjg.com

djackson@dbjg.com

jwilson@dbjg.com

James L. Etheridge, TX Bar No. 24059147

Ryan S. Loveless, TX Bar No. 24036997

ETHERIDGE LAW GROUP, PLLC

2600 E. Southlake Blvd., Suite 120 / 324

Southlake, TX 76092

Tel.: (817) 470-7249

Jim@EtheridgeLaw.com

*Attorneys for Plaintiff Worlds, Inc.*

**CERTIFICATE OF SERVICE**

I hereby certify that all counsel of record, who are deemed to have consented to electronic service are being served with a copy of this document via email on December 21, 2020.

*/s/ Wayne M. Helge* \_\_\_\_\_  
Wayne M. Helge

**Infringement Claim Chart for Claims 1, 2, 5, and 10 of U.S. Patent No. 8,082,501**  
**Microsoft's Minecraft Products**

Worlds' preliminary infringement contentions for U.S. Patent No. 8,082,501 ("the '501 Patent") being asserted against Defendant Microsoft Corporation and its Minecraft Products (collectively, the "Minecraft Products" and also "the Accused Products and Services" as set forth below) (including all versions of any Minecraft Product sold, offered for sale, or imported within the damages period, such as the Minecraft Java Edition Product and associated software, the Minecraft Bedrock Edition Product and associated software, the official Minecraft Realms subscription-based servers and their associated server software, and the official Minecraft server software used for establishing a private Minecraft server).

Worlds' statements herein are based on publicly available information and documentation that Worlds has obtained. Discovery is ongoing, and certain documentation and information may not yet be available to Worlds that are fundamental to its infringement claims. In particular, Worlds has not yet deposed officers or employees of defendant or any third parties concerning the operation, functionality and integration, packaging, or terminology of the accused products. Worlds reserves the right to amend its infringement disclosures in light of further discovery and consistent with this Court's Order Governing Proceedings – Patent Case.

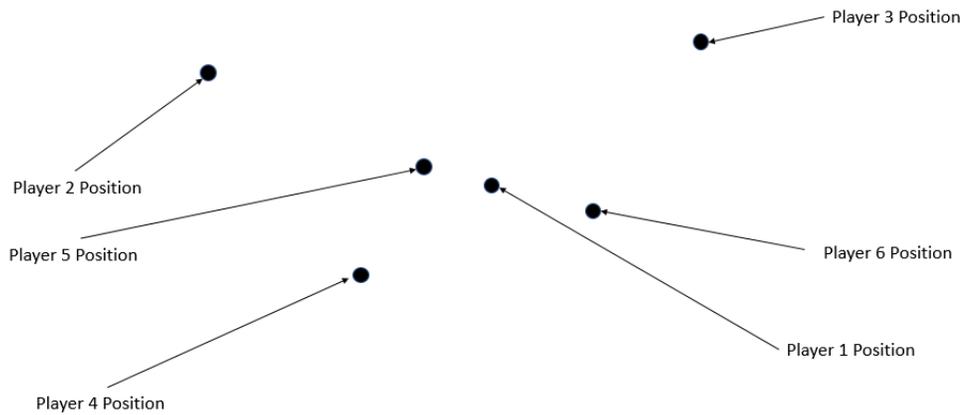
In addition to the charted infringement contentions below, Worlds incorporates the allegations set forth in its Complaint as if fully set forth herein. As set forth in Worlds' Complaint, Worlds claims that each element of each asserted claim has been directly infringed by Defendant Microsoft. Upon information and belief, Microsoft has tested and used its Accused Products and Services in the United States during the damages period, to include the practice of the claimed methods of the Asserted Claims, and which provides a basis for Microsoft's direct infringement of at least claims 1, 2, 5, and 10 of the '501 patent under 35 U.S.C. § 271(a).

As set forth in Worlds' Complaint, Worlds also claims that Defendant Microsoft has jointly infringed each asserted claim through client devices performing each claimed method step in a manner attributable to Defendant Microsoft. Microsoft benefited from users who used their Microsoft Minecraft accounts and played Minecraft with customized avatars in a multiplayer mode, which provides a server/client architecture with filtering/crowd control features for multiplayer use, using either partnered servers with licensed Minecraft server software, or through the Minecraft Realms service, and a licensed version of Microsoft's Minecraft Product. The manner and timing of the activities of the users was controlled by Microsoft by virtue of authentication, accounting, and authorization based on Microsoft Minecraft accounts, Microsoft software, Minecraft Terms and Conditions, Minecraft Realms Terms and Conditions, and Microsoft licenses and agreements, including Microsoft's Minecraft End User's License Agreement in effect during the period of infringement. Additionally, Microsoft provided assistance to customers having difficulty accessing Minecraft's features. This assistance included Microsoft's links to help articles, which assisted users wishing to customize their avatars and needing assistance to resolve problems with customization encountered along the way, and Microsoft's provision of Minecraft Realms

and other technical support to help customers having problems with internet/multiplayer gameplay. Therefore, the asserted claims of the '501 Patent were directly infringed by Microsoft under 35 U.S.C. § 271(a) because all of the steps of the asserted method claims were attributable to Microsoft through the activities of Minecraft's users, pursuant to Microsoft's direction and control.

If for any reason any accused product is found not to directly meet each element of any asserted claim, Worlds alleges that any difference(s) between the claim element in the asserted claim(s) and the accused product is insubstantial and Defendant Microsoft has infringed the asserted claim(s) under the doctrine of equivalents.

<b>U.S. Patent No. 8,082,501 Claims</b>	<b>Microsoft's Minecraft Products</b>
<p>1. A method for enabling a first user to interact with other users in a virtual space, each user of the first user and the other users being associated with a three dimensional avatar representing said each user in the virtual space, the method comprising the steps of:</p>	<p>To the extent the preamble is limiting, Microsoft's Minecraft Product, such as the Minecraft Java Edition Product, performs a method for enabling a first user to interact with other users in the Minecraft virtual space, wherein the first user and the other users are each associated with a three-dimensional avatar representing the user in the virtual space, and where each of the client processor is in communication with a server process, such as programmed Minecraft Realms server software hosted by Microsoft's subsidiary Mojang, and/or other server partners pursuant to the Terms and Conditions and End User License Agreements which provided Microsoft the ability to, timing and manner for, and benefit of using a server process, such as the Minecraft server software, to manage multiple users within the Minecraft virtual space.</p> <p>Microsoft's Minecraft Products are played in part on a "server" and multiple users interact with the server, each using "client" source code. The server hosts a "world" in which the users interact.</p> <p><i>See:</i></p> <p><a href="https://minecraft.gamepedia.com/World">https://minecraft.gamepedia.com/World</a>  <a href="https://minecraft.gamepedia.com/Server">https://minecraft.gamepedia.com/Server</a>  <a href="https://minecraft.gamepedia.com/Client.jar">https://minecraft.gamepedia.com/Client.jar</a></p> <p>For example, six players could be positioned in a world as follows:</p>



Each user is represented in the world as a three-dimensional avatar, with the default avatars being “Steve” or “Alex.”



*See:*

<https://minecraft.gamepedia.com/Player>

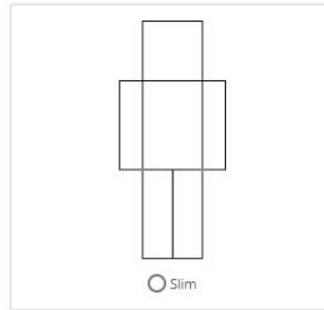
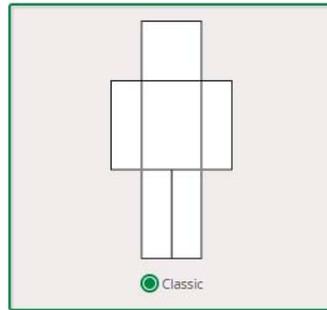
Thus, the Microsoft Minecraft Products perform a method for enabling a first user to interact with other users in a

	<p>virtual space, each user of the first user and the other users being associated with a three dimensional avatar representing said each user in the virtual space.</p>
<p>customizing, using a processor of a client device, an avatar in response to input by the first user;</p>	<p>The use of Microsoft Minecraft Products, such as use of a Minecraft Java Edition Product, included running software programming for a first user’s client processor to perform customizing an avatar in response to an input from the first user. Microsoft Minecraft Products provided the user’s client device processor with the ability to, timing and manner for, and benefit of performing the step of customizing an avatar. The Microsoft Minecraft Product allows a user to customize the “skin” of their avatar by, for example, uploading an image file, downloading a skin pack, or selecting a customized option in a skin selection menu.</p> <p>“In Java Edition, players can change skins on the preferences page of minecraft.net or the launcher by uploading a PNG image file, which then replaces the default skin. Players also have the option to have three or four pixel wide arms on the character model.</p> <p>In Bedrock Edition, the player can change the skin by opening the settings from the main menu and going to the skin settings. The two default skins are Alex and Steve but the player can download and use Skin Packs from the Marketplace or, on the Windows 10, iOS and Android versions of the game, use their own skin by selecting the "Custom" option in the skin selection menu.”</p> <p><i>See:</i></p> <p><a href="https://minecraft.gamepedia.com/Player">https://minecraft.gamepedia.com/Player</a>  <a href="https://minecraft.gamepedia.com/Skin">https://minecraft.gamepedia.com/Skin</a>  <a href="https://help.minecraft.net/hc/en-us/articles/360034635452-Minecraft-Skins-">https://help.minecraft.net/hc/en-us/articles/360034635452-Minecraft-Skins-</a>  <a href="https://my.minecraft.net/en-us/profile/skin">https://my.minecraft.net/en-us/profile/skin</a></p>

## CHANGE YOUR SKIN - PC/MAC

### 1. SELECT MODEL

First choose the shape of your Minecraft character by picking one of two models.



### 2. UPLOAD CUSTOM SKIN

After you've created your custom skin, upload it here:

Drop your skin file here, or:

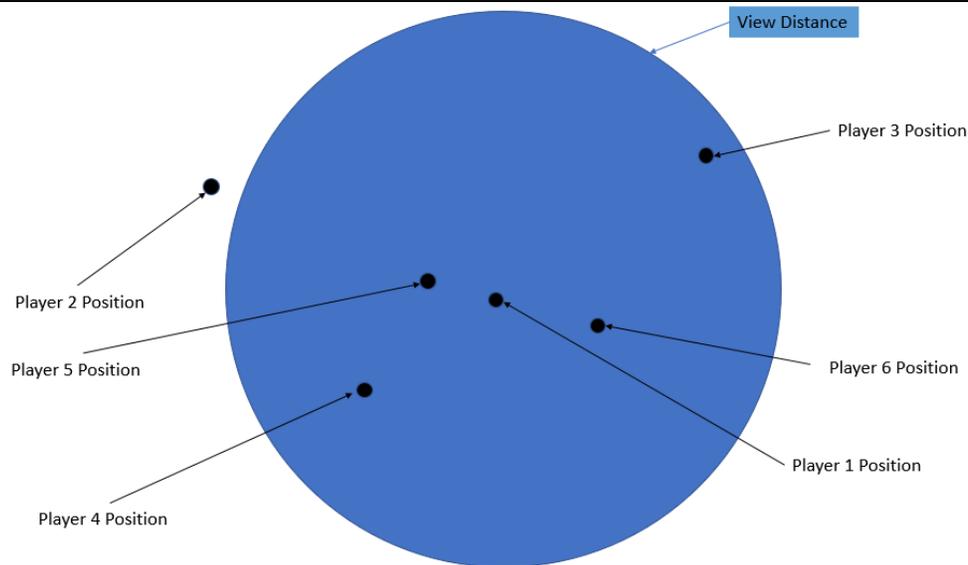
SELECT A FILE

**Note!** This changes your skin on the Minecraft: Java Edition. You can change your skin in the other versions of Minecraft by accessing menus in-game. For more information, check [this handy help article](#).

Do you wonder HOW you give your character a unique texture? Create your own skin or find ready-made skins on the internet! Check out [this help article](#) for more information.

Thus, Microsoft Minecraft Products perform customizing, using a processor of a client device, an avatar in response to input by the first user.

<p>receiving, by the client device, position information associated with fewer than all of the other user avatars in an interaction room of the virtual space, from a server process,</p>	<p>The use of Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included running software programming for a first user's client device to perform receiving from a Minecraft server process, the position information associated with less than all of the other users' avatars (namely programmed software, including but not limited to use of the "View Distance" option on Minecraft servers). The use of Microsoft Minecraft Products provided the user's client device with the ability to, timing and manner for, and benefit of performing this "receiving" step.</p> <p>The Microsoft Minecraft Products define a server property called "view distance" that limits the data, including the positions of other avatars, sent to client devices from the server process. The worlds on the server in Microsoft Minecraft Products can include various rooms where the avatars may interact.</p> <p><i>See:</i></p> <p><a href="https://minecraft.gamepedia.com/Overworld">https://minecraft.gamepedia.com/Overworld</a> <a href="https://minecraft.gamepedia.com/The_Nether">https://minecraft.gamepedia.com/The_Nether</a> <a href="https://minecraft.gamepedia.com/The_End">https://minecraft.gamepedia.com/The_End</a></p> <p>For example, as depicted in the illustration below, enforcement of the view distance property allows positions for Player 3, Player 4, Player 5, and Player 6 to be sent to the client device for Player 1. The position for Player 2 would not be sent from the server to the client device for Player 1.</p>
---	--



“In Minecraft, view distance dictates how many chunks a player can see when looking in a certain direction. With this configured on the server, this will limit the amount of data (in a chunk radius) that is sent to every player. Increasing this number will enhance the distance that each player can see, though at the cost of RAM (memory). Keeping this value low will reduce the distance that a player can see, however will load less chunks around the player, and therefore decrease the RAM (memory) usage on the server.”

*See:*

<https://nodecraft.com/support/games/minecraft/configuring-view-distance-on-a-minecraft-server>

<https://minecraft.gamepedia.com/Server.properties>

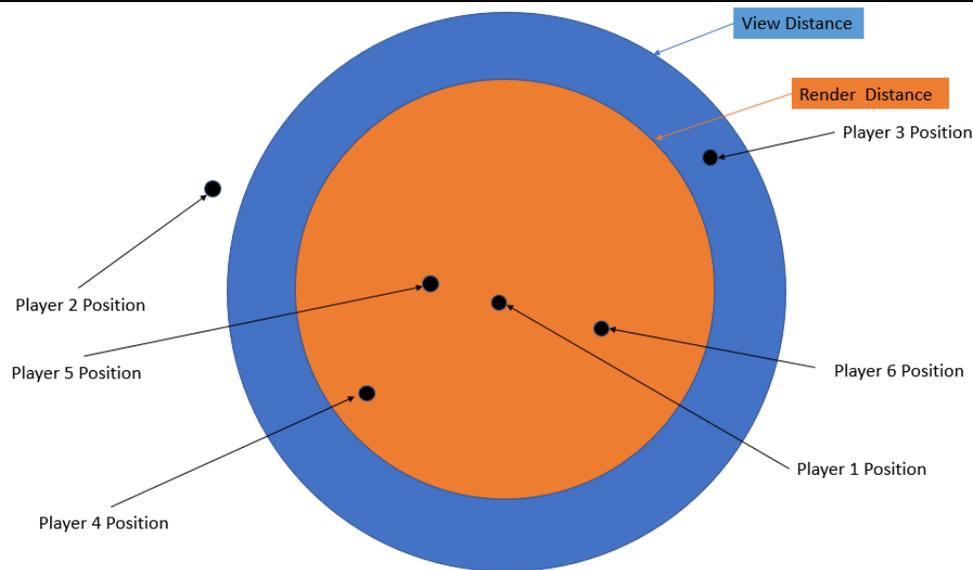
<https://shockbyte.com/billing/knowledgebase/134/Adjusting-View-Distance-on-Your-Minecraft-Server.html>

Thus, the Microsoft Minecraft Products perform receiving, by the client device, position information associated with fewer than all of the other user avatars in an interaction room of the virtual space, from a server process.

wherein the client device does not

The Microsoft Minecraft Products operate so that the client device does not receive position information of at least some avatars that fail to satisfy a participant condition imposed on avatars displayable on a client device display of

<p>receive position information of at least some avatars that fail to satisfy a participant condition imposed on avatars displayable on a client device display of the client device;</p>	<p>the client device.</p> <p>As noted above, the “view distance” prevents position information of at least some avatars from being sent to a client device. Positions of players that fail to be within the view distance from that player to the player associated with the client device are not transmitted. In addition, Microsoft’s Minecraft Products allow a user to choose several options, including an option to set a “render distance.”</p> <p>“The render distance controls how many chunks of the world are visible at once. The fewer chunks that are included, the faster each frame can be rendered, resulting in higher frames per second (FPS).”</p> <p><i>See:</i></p> <p><a href="https://minecraft.gamepedia.com/Options">https://minecraft.gamepedia.com/Options</a> <a href="https://minecraft.gamepedia.com/Debug_screen">https://minecraft.gamepedia.com/Debug_screen</a></p> <p>For example, as depicted in the illustration below, enforcement of the render distance property determines that the set of players (Player 4, Player 5, and Player 6) that may be displayable. Referring to the illustration below, player 2 fails to satisfy the “view distance” condition and the “render distance” condition set by the client, and its position is not received by the client. Player 3 would also not be displayable.</p>
---	---

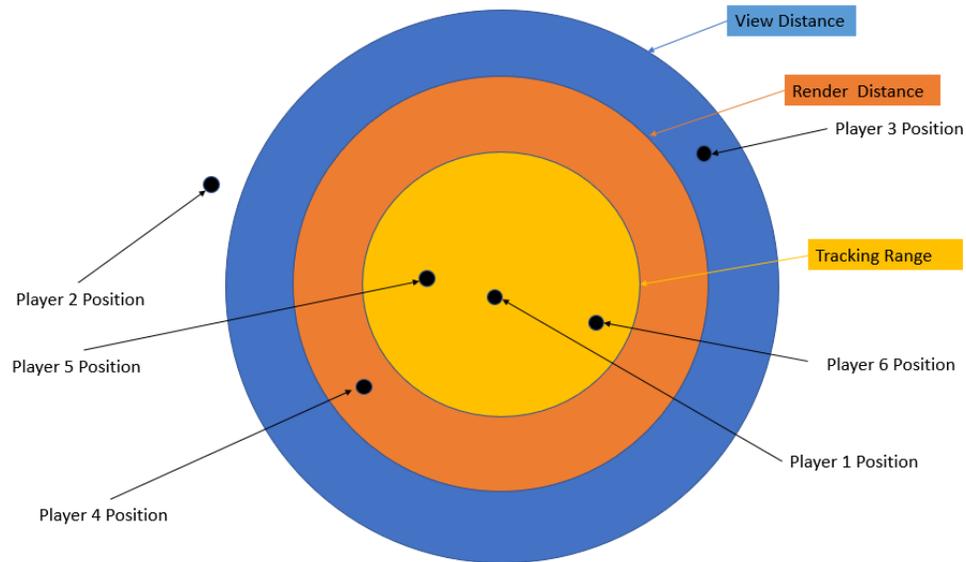


Thus, the Microsoft Minecraft Products perform the above-described “receiving” step such that a client device does not receive position information of at least some avatars that fail to satisfy a participant condition imposed on avatars displayable on a client device display of the client device.

determining, by the client device, a displayable set of the other user avatars associated with the client device display; and

Use of the Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included running software programming for a first user’s client process to perform determining a displayable set of the other user avatars associated with the client device display, and displaying, on the client device display, the displayable set of other user avatars associated with the client device display. The use of Microsoft Minecraft Products provided the user’s client device with the ability to, timing and manner for, and benefit of performing this “determining” step. This determining step includes the Minecraft Java Edition Product’s determining from the received positions of less than all of the other users’ avatars, a displayable set of avatars to be displayed using filtering and crowd control features in the Minecraft Java Edition Product including, but not limited to, the “render distance” and “entity tracking range” options. As shown below, the user can use these options to further determine which of other users’ avatars are to be displayed to the user, and rendering the limited set of other user’s avatars on the user’s display.

The Microsoft Minecraft Products also have other properties that affect whether other player avatars are displayed, including a “entity-tracking-range” for players. Players within the tracking range may be visible to the player:



“Description: Controls the range in blocks that entities will become "visible" or otherwise known as "tracked" to the client. Entities outside of this range will be invisible as they are not being rendered to preserve CPU usage and bandwidth. This is particularly useful for PVP servers, as turning down the player range will 'nerf' wallhacks and radar to some extent.”

*See:*

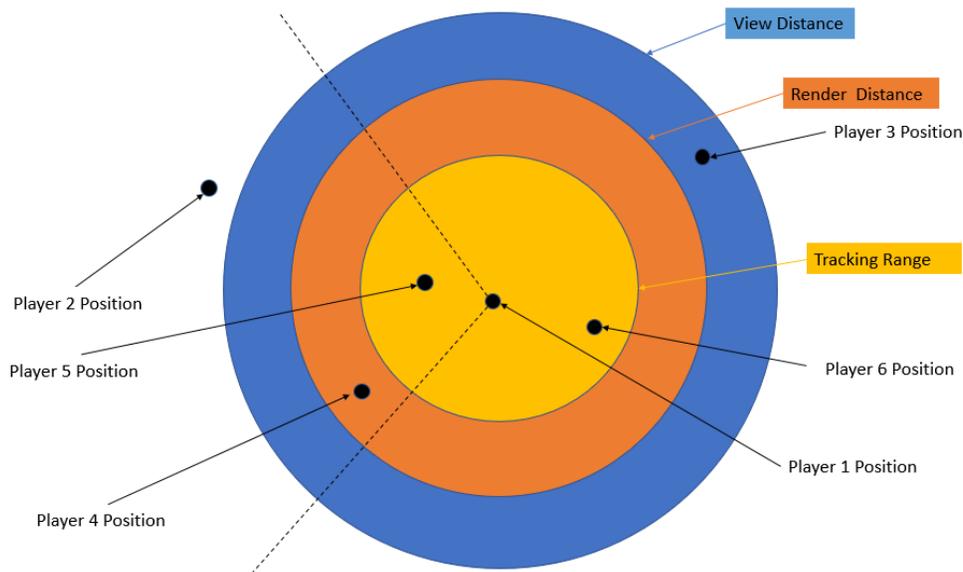
[https://www.reddit.com/r/admincraft/comments/1oiu7q/spigotyaml\\_question/](https://www.reddit.com/r/admincraft/comments/1oiu7q/spigotyaml_question/)

For example, as depicted in the illustration above, enforcement of the player entity-tracking-distance property determines the set of players (Player 5 and Player 6) who may be displayable. Players 2, 3, and 4 would not be displayable.

Also, Microsoft’s Minecraft Products monitor the orientation of the player and only will display players within the field of view (FOV) of the player.

<https://minecraft.gamepedia.com/Options> (FOV option described as “A value that controls how much of the game world is visible on the screen. Also known as Field of View.”)

For example, as depicted in the illustration below, a FOV is shown with dashed lines and assumes the eyes of player 1 are oriented to the left of the drawing. Enforcement of the player field of view (FOV) property allows Player 5 to be displayed to player 1. Players 2, 3, 4, and 6 would not be displayed.



Thus, the Microsoft Minecraft Products perform determining, by a client device, a displayable set of the other user avatars associated with the client device display.

displaying, on the client device display, the displayable set of the other user

Use of the Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included running software programming for a first user's client process to perform displaying, on the client device display, the displayable set of other user avatars associated with the client device display.

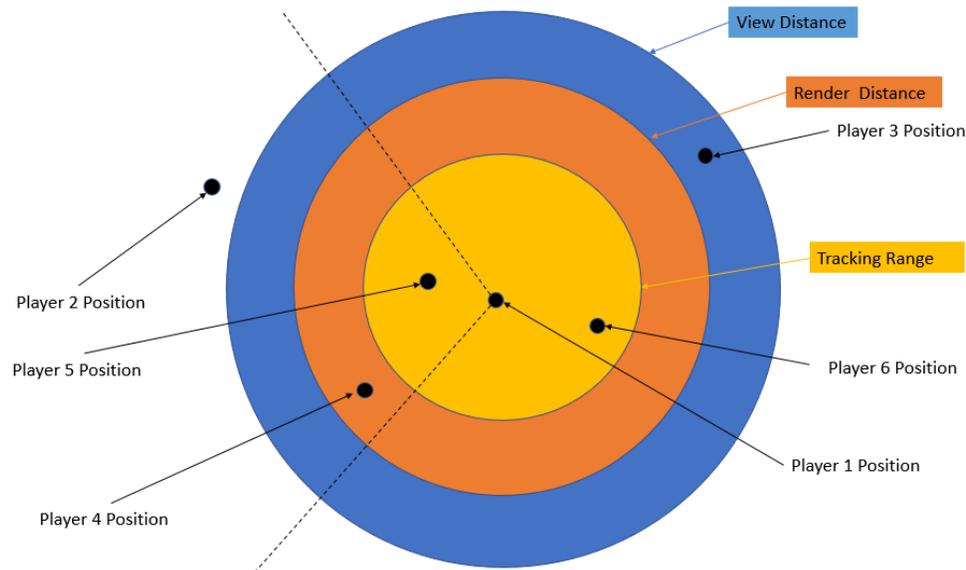
<p>avatars associated with the client device display.</p>	<p>When the various properties of Minecraft are enforced as described above regarding the other elements of claim 1, the remaining players are displayed on the display of the client device. In the example provided above, Player 1's display will display Player 5.</p> <p>And to summarize, the client device would not display the following players because of the following:  Player 2 – Position not sent to the client because player 2's position relative to player 1 was outside the view distance and not displayed because player 2's position relative to player 1 was outside the render distance.  Player 3 – Not displayed because player 3's position relative to player 1 was outside the render distance.  Player 4 – Not displayed because player 4's position relative to player 1 was outside the tracking range.  Player 6 - Not displayed because player 6's position relative to player 1 was outside the field of view of player's 1 current orientation.</p> <p>The display can have different perspectives, including first- and third-person views.</p> <p><i>See:</i></p> <p><a href="https://minecraft.gamepedia.com/Third-person_view">https://minecraft.gamepedia.com/Third-person_view</a></p> <p>Thus, the Microsoft Minecraft Products perform displaying, on the client device display, the displayable set of the other user avatars associated with the client device display.</p>
<p>2. The method according to claim 1, further comprising the step of:</p>	<p>The claim chart with respect to claim 1 above is hereby incorporated by reference.</p>
<p>monitoring an orientation of the first user avatar;</p>	<p>Use of the Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included running software programming for a first user's client device to perform monitoring an orientation of a user's avatar. Microsoft Minecraft Products provided the user's client device with the ability to, timing and manner for, and benefit of performing this "monitoring" step.</p> <p>As described above for claim 1, the orientation of player 1 is monitored to enforce the field of view (FOV) condition.</p>

Thus, the Microsoft Minecraft Products perform monitoring an orientation of the first user avatar.

wherein the step of determining comprises filtering the other user avatars based on the monitored orientation of the first user avatar.

The claim chart with respect to claim 1's "determining" step above is hereby incorporated by reference. Use of the Microsoft Minecraft Products included, as part of the "determining" step described above, running software programming for a first user's client device to perform filtering the other user avatars based on the monitored orientation of the first user avatar.

As further described above for claim 1, the orientation of player 1 is monitored to enforce the field of view (FOV) condition.



Thus, the step of determining, by the client device, a displayable set of the other user avatars associated with the client device display, comprises the Microsoft Minecraft Products performing filtering the other user avatars based on the monitored orientation of the first user avatar.

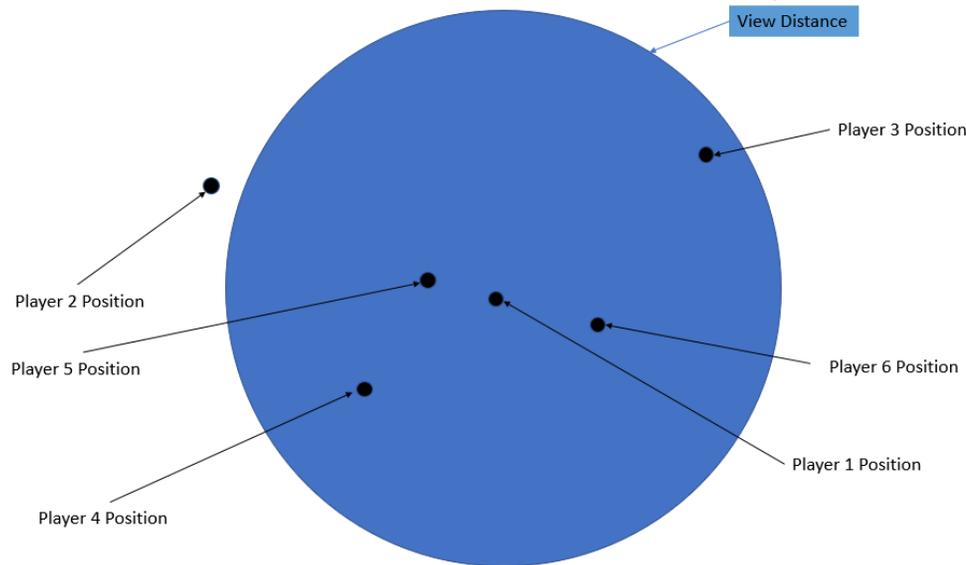
5. The method according to claim 1, further comprising receiving by the client device orientation information associated with fewer than all of the other user avatars, wherein the client device does not receive orientation information of at least some avatars of the other user avatars in the virtual space.

The claim chart with respect to claim 1 above is hereby incorporated by reference.

Use of the Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included running software programming for a first user's client device to perform receiving orientation information associated with fewer than all of the other user avatars, wherein the client device does not receive orientation information of at least some avatars of the other user avatars in the virtual space. Microsoft Minecraft Products provided the user's client device with the ability to, timing and manner for, and benefit of performing this "receiving" step.

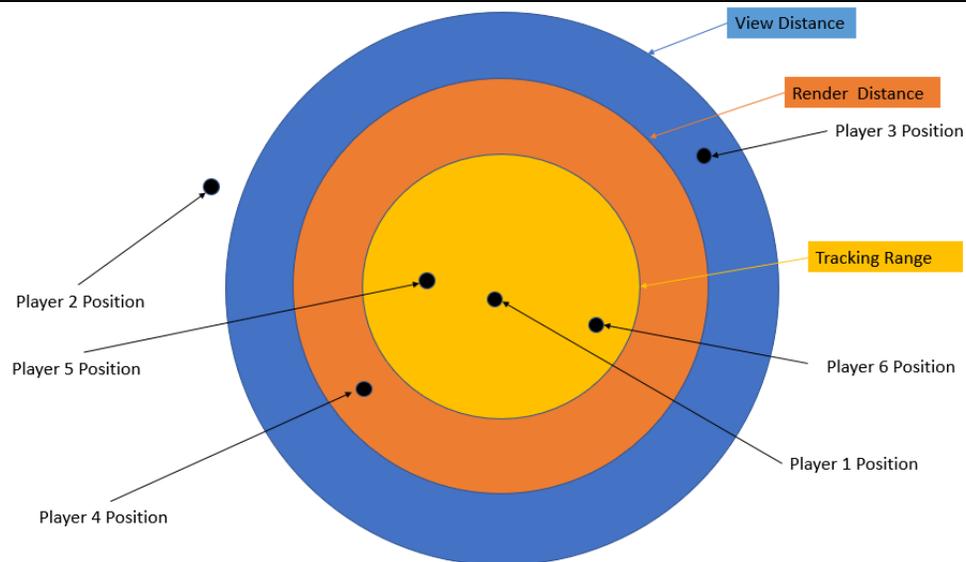
The Microsoft Minecraft Products define a server property called "view distance" that limits the data, including the orientations of other avatars, sent to client devices from the server process.

For example, as depicted in the illustration below, enforcement of the view distance property allows orientations for Player 3, Player 4, Player 5, and Player 6 to be sent to the client device for Player 1. The orientation for Player 2 would not be sent from the server to the client device for Player 1.



"In Minecraft, view distance dictates how many chunks a player can see when looking in a certain direction. With

	<p>this configured on the server, this will limit the amount of data (in a chunk radius) that is sent to every player. Increasing this number will enhance the distance that each player can see, though at the cost of RAM (memory). Keeping this value low will reduce the distance that a player can see, however will load less chunks around the player, and therefore decrease the RAM (memory) usage on the server.”</p> <p><i>See:</i></p> <p><a href="https://nodecraft.com/support/games/minecraft/configuring-view-distance-on-a-minecraft-server">https://nodecraft.com/support/games/minecraft/configuring-view-distance-on-a-minecraft-server</a>  <a href="https://minecraft.gamepedia.com/Server.properties">https://minecraft.gamepedia.com/Server.properties</a></p> <p>Thus, the Microsoft Minecraft Products perform receiving by the client device orientation information associated with fewer than all of the other user avatars, wherein the client device does not receive orientation information of at least some avatars of the other user avatars in the virtual space.</p>
<p>10. The method according to claim 1, wherein the step of determining comprises filtering the other user avatars based on a limit of the other user avatars that may be displayed on the client device display, the limit being set at the client device.</p>	<p>The claim chart with respect to claim 1 above, including the claim chart with respect to claim 1’s “determining” step above, is hereby incorporated by reference.</p> <p>Use of the Microsoft Minecraft Products, such as a Minecraft Java Edition Product, included, as part of the “determining” step described above, running software programming for a first user’s client device to perform filtering the other user avatars based on a limit of the other user avatars that may be displayed on the client device display, the limit being set at the client device including, but not limited to, the “entity tracking range” option. Microsoft Minecraft Products provided the user’s client device with the ability to, timing and manner for, and benefit of performing this “filtering” step.</p> <p>Client devices in the Microsoft Minecraft Products filter other user avatars based on a limit of the other user avatars that may be displayed on the client device display, the limit being set at the client device.</p> <p>The “entity-tracking-range” in Minecraft is a limit of the other players (specifically how far away the other players may be from the first player) to be displayed on the client device. The code in the client device sets the entity-tracking-distance. Players within the tracking range may be visible to the player:</p>



“Description: Controls the range in blocks that entities will become "visible" or otherwise known as "tracked" to the client. Entities outside of this range will be invisible as they are not being rendered to preserve CPU usage and bandwidth. This is particularly useful for PVP servers, as turning down the player range will 'nerf' wallhacks and radar to some extent.”

*See:*

[https://www.reddit.com/r/admincraft/comments/1oiu7q/spigotyaml\\_question/](https://www.reddit.com/r/admincraft/comments/1oiu7q/spigotyaml_question/)

For example, as depicted in the illustration above, enforcement of the player entity-tracking-distance property determines that the set of players (Player 5 and Player 6) who may be displayable. Players 3 and 4 would not be displayable.

Thus, the step of determining, by the client device, a displayable set of the other user avatars associated with the client device display, comprises the Microsoft Minecraft Products performing filtering the other user avatars based on a limit of the other user avatars that may be displayed on the client device display, the limit being set at the

	client device.
--	----------------