

Terminology

Some basic explainers for some of the terms used, listed alphabetically.

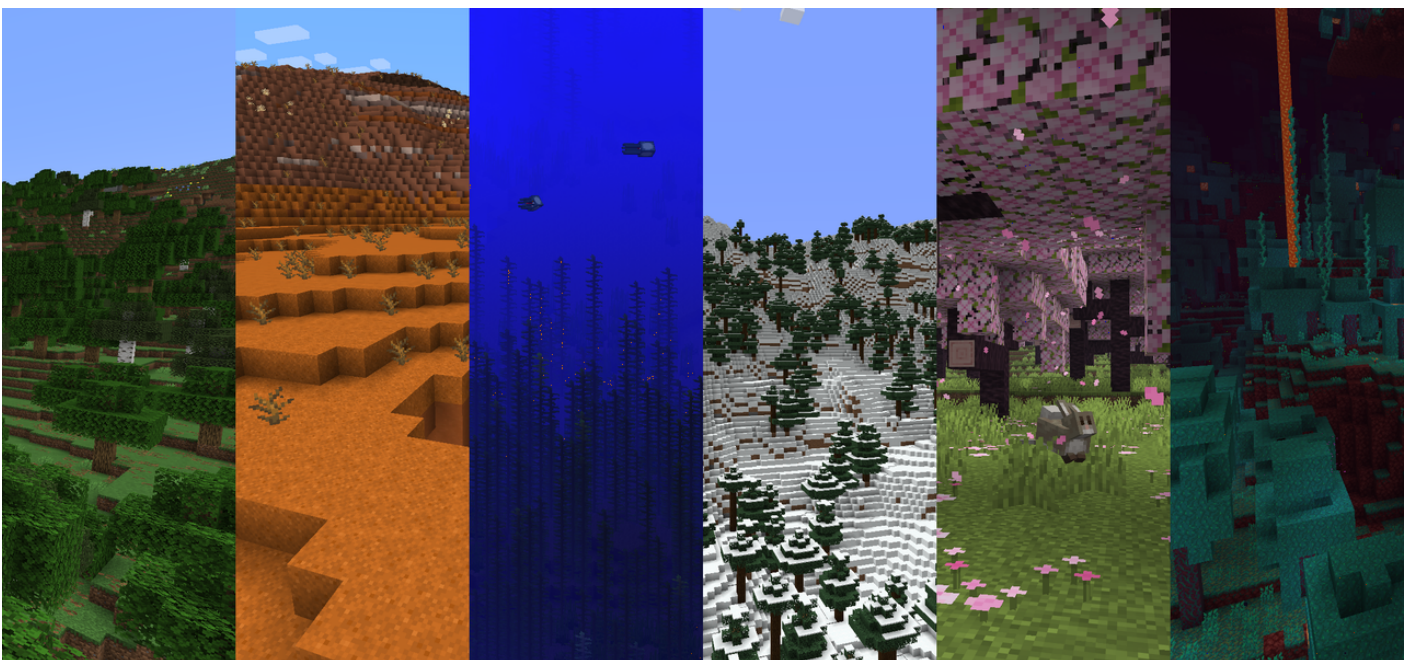
Architectural Terms

[A glossary of Minecraft architectural terms](#) that might be useful to know. If you've taken art history classes or played any building games before, you're probably familiar with a lot of these. Here's [a more extensive list](#) on Wikipedia as well.

Biome

Different regions in a world generated by Minecraft. Each region has its own unique terrain shapes, block materials, mob spawns, weather patterns, atmosphere/skybox colors, grass and tree colors, and more. Some things are only found in specific biomes, so be sure to explore the world and try to find them. There's around 60 biomes in version 1.21.1, (plus some more added by Cobblemon which trigger [Pokémon spawns](#)). In a recent update to Cobblemon, they changed the spawn behavior to prioritize family groups over individual spawns - it's very sweet to see roving packs of 'mons going about their business.

There's a logic to what biomes generate where, their size and frequency, and which biomes tend to spawn together. Categories of biomes in Minecraft include offshore biomes, highland biomes, woodland biomes, wetland biomes, flatland biomes, arid biomes, cave biomes, and Nether biomes. Villages and other predetermined structures can generate in different biomes.



Some example biomes from the Minecraft wiki. From left to right, forest, badlands, deep ocean, grove, cherry grove, and warped forest. There's a lot of diversity to be seen, just like in the real world, and this also extends to cave biomes and Nether biomes.

There are mods to add a large variety of unique biomes, but it's better to have those in place from the start rather than add them to an already generated world.

Chunk

A chunk is a 16x16 segment of the world. The game divides maps into chunks to keep things manageable. Chunks extend from the bottom void of the world all the way up to the vertical build limit of 384 blocks. Chunks load in around the player as needed as you explore the world. Chunk generation is largely dependent on the map seed, but can generate with slight variations under some conditions. You can read more about the technical side of chunk loading [here](#).

Client Side / Server Side

Client Side is whatever the player installs on their computer - some mods like font replacers are usually client side only, meaning only you would see the changes (your computer!). Server Side mods usually need to be installed on both Server and Clients. Mod hosting sites like Modrinth or Curseforge will say whether a mod is intended for Servers or Clients.

Game Modes

Different gameplay modes you can switch between. Switching modes is usually restricted to players with admin control, but we want everyone to pick how they play in our server. You can use [slash commands](#) to switch modes, or you can hit F4 to quickly switch between Creative and Survival.

Creative Mode - Creative Mode gives you access to all the blocks available in the game including spawn eggs for passive and hostile mobs (like a zombie or a sea turtle). We're playing in Peaceful mode so hostile mobs probably won't actually spawn. Every item or block you pick is effectively unlimited and this mode removes pretty much all challenge from the game. The only limit is your imagination (and server permissions).

It's very easy to build things in Creative Mode, but equally easy to break things by accident, especially chests or other containers. If you pop a chest, all the contents will spill out - do your best to put things back if this happens! If you're using the [Effortless Building](#) mod (bound to left-alt by default), you can very quickly place or replace blocks in huge chunks, and can use ctrl-z or ctrl-y to undo or redo anything placed in this mode.

Survival Mode - The default Minecraft experience, minus the danger. There is fall damage but it's basically negligible unless you fall from really high up (like into a ravine or something). You can die but there's no actual loss and no need to worry about losing items or experience on death.

Block placement is as expected, you will need to have specific blocks in your inventory to build with (either legitimate or through "cheating"), and tool strength and type matters for clearing things out of the way - eg, iron tools for mining ores, shovels or axes for dirt and trees. You can't accidentally break things just by misclicking.

Spectator Mode - I think you can just follow another player and see what they're doing

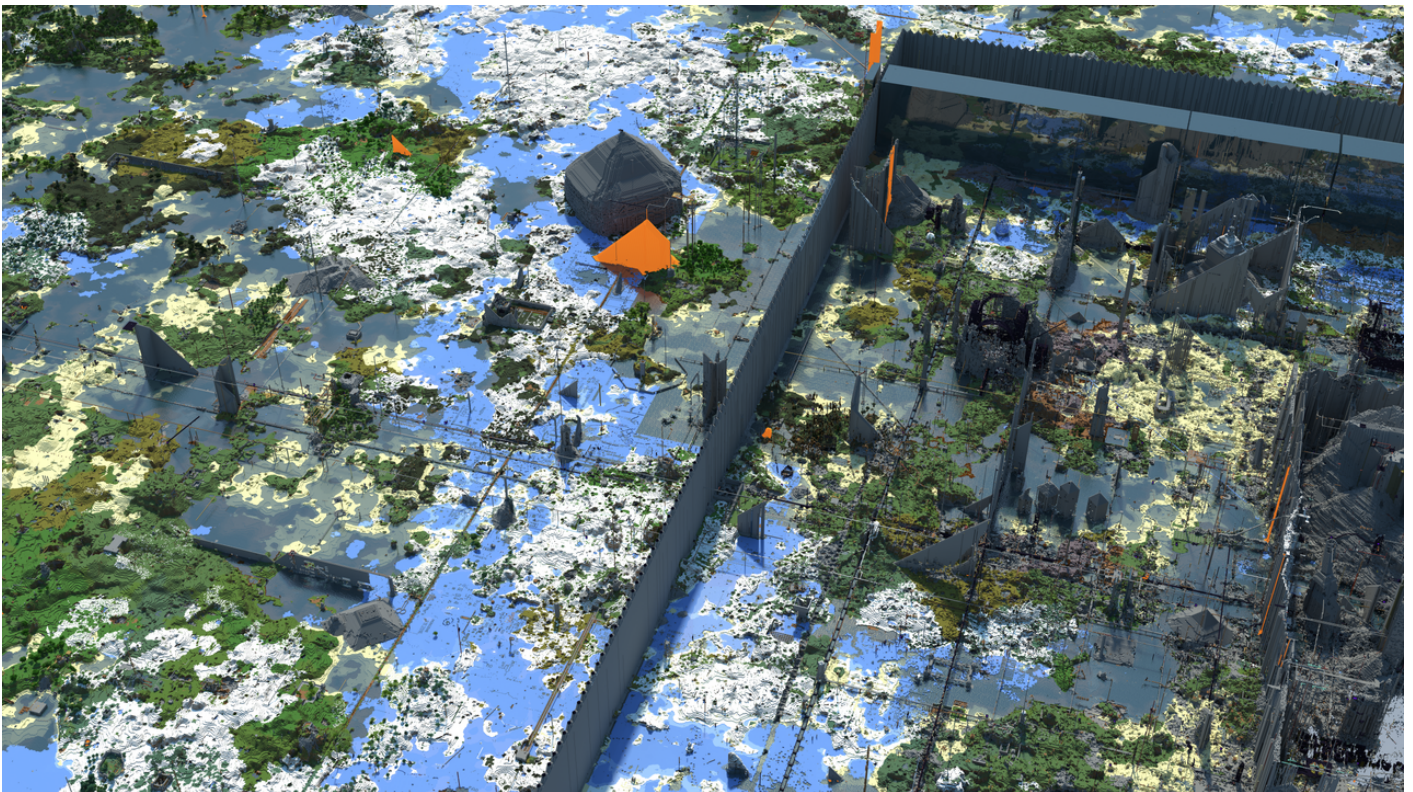
Adventure Mode - Not sure what this is about, not applicable for us. In Adventure Mode, you can't break or place blocks, or do anything that could be considered griefing. Seems cool for puzzle based maps, like an escape room sort of thing, or other multiplayer challenges.

Griefing

Intentionally antagonistic behavior in a multiplayer game. In a Minecraft context, this means things like stealing items from other players, breaking people's stuff or fucking around with preexisting buildings, intentional friendly-fire, abusing slash commands (like spamming the time-change commands) or otherwise being a dick. This is a much bigger deal in games where there's permadeath or actual consequences for dying, like losing your inventory.

We have a lot of bulk editing mods in this server meaning it's much easier to accidentally fuck up other people's stuff without intending to do so. We're currently looking into ways to protect areas to prevent any catastrophic losses of builds.

Below is an example of griefing in Minecraft from Wikipedia; the terrain looks incredibly fucked up.



Mods

Community made, unofficial modifications to a game. Anything that modifies the Java code from its vanilla state that is more complex than texture or audio swaps. Mods can be as simple as adding new block types (like furniture or new ores) and as complex as adding realistic physics to the game or porting the entirety of the Pokemon series and its game mechanics. Everyone needs to be running the same mods with matching versions or things get weird - like the game failing to launch, or blocks failing to place - usually the game will warn you about mod mismatches and stop you from joining the server until the conflict is resolved.

Optimization mods are mods intended to improve performance to keep the game running smoothly for everyone, such as reducing the amount of ram needed to run the server, limiting draw distance or simulation distance for things far away from active players, or fixing bugs from the base game.

On the [Overview](#) page, there's a short list of some of the mods we're running. You can view the full list on the Mods tab in Prism Launcher, or in game via settings.

Modloader

Modloaders are the software responsible for interfacing between the mods and Minecraft that allows mods to integrate with the game. There's many kinds of loaders with different capabilities. The loader we're using is Neoforge, a successor (and fork of) Forge from the original team minus the head dude ([tl;dr, he sucks and is hard to work with](#)), and mods built for Forge should also work on Neoforge. Besides Neoforge, there is also Forge, Fabric, and Quilt, plus some other ones used for older versions of Minecraft, and there are also mods to bridge compatibility between different loaders

If there are any mods you'd like to suggest for the server, make sure they're built for version 1.21.1 & Neoforge.

Modpack

A Modpack is a zip containing the mods as well as info about which version of Minecraft and Modloader to install. Installed via drag and drop with a launcher like Prism Launcher. Modpacks are a curated collection of mods intended to be used together, and contain all necessary dependencies needed to play.

Resource Pack

Resource Packs allow changes to textures, sounds, splash screens, fonts, translations, etc without any code modification. Resource Packs are also in .zip format and installed via drag and drop, or installed directly through a launcher. Previously referred to as *texture packs*, renamed when more asset types became supported. Resource Packs are entirely client side and can be freely changed by the players on a server. Resource Packs aren't considered Mod mods since they don't modify game code.

Shader Pack

Shader Packs are mods that enhance the visuals of the game. They can add things like volumetric clouds, waves and caustics in the water, bloomy lighting & raytraced lighting effects, and more. Shader packs are client side so every player might have a completely different visual experience within the same server. Some shaders are more like a vanilla+ experience, while some make it feel like totally different game, similar to playing a game with ReShade.

There's a range of options available for shaders, from cartoonish or fantasy, to hyper realism, and some are specifically tuned for low-end computers with onboard graphics. Some example shaders to check out: [Aurora Fantasy](#), [Complementary-Unbound](#), [Complementary-Reimagined](#), [Rethinking Voxels](#), [BSL Shaders](#), and [Sildur's Vibrant Shaders](#).

We picked [Solas Shader](#) to be the default for the server because it works well with the mod loadout at a modest performance cost. You're free to use whatever shaders you want if you're not into this look. You can also adjust parameters to improve performance or visuals as needed.

Skin

The appearance of your player model! These are connected to your Minecraft account and can be changed through Prism Launcher on the accounts tab. You can browse a site like [Skindex](#) or make your own skin with a template or a browser based editor [like this one](#). You can make your character look like whatever you want, like [Dog in Business Suit](#), [Hatsune Miku](#), or [Baked Beans Where There Shouldn't Be Beans](#). Skin textures are 64x64 pixels, saved in png format, and support transparency.

Custom skins go into a folder in your Instance directory and can be viewed and swapped between in Prism Launcher. There's also capes but those appear to be limited rewards for people who attended official Minecraft events, and custom capes require a mod to display.

Slash Commands

Commands that can manipulate the game state, change your position in the world, modify your inventory, and more; also known as console commands in other games. Read more about specific commands [here!](#)

Vanilla

An unmodified *Minecraft* game, client or server. This term can be used for any piece of software.