

Minecraft: Java Edition – 1.21.11

Minecraft: Java Edition – 1.21.11

Mounts of Mayhem

(Unofficial Changelog)



Releases

1.21.11

[25w41a](#) | [25w42a](#) | [25w43a](#) | [25w44a](#) | [25w45a](#) | [25w46a](#) | [1.21.11 Pre-Release 1](#) | [1.21.11 Pre-Release 2](#) | [1.21.11 Pre-Release 3](#)

Navigation

[Gameplay Changelog](#)

[Technical Changelog](#)

[Data Pack Changelog](#)

[Resource Pack Changelog](#)

[Minecraft Server Management Protocol Changelog](#)

[Bug Fixes](#)

[Credits](#)

Gameplay Changelog

Gameplay Changelog

Additions

New Mobs

Nautilus

- Is neutral and aquatic
 - Becomes hostile when attacked
 - Is automatically hostile towards pufferfish
- Naturally spawns underwater in all ocean biomes
- Can be tamed using pufferfish or a bucket of pufferfish
- When tamed, can be equipped with a saddle, which allows a player to mount it, and nautilus armor, which adds armor points to the mob
- When mounted, gives the Breath of the Nautilus effect to its rider
- Has a dash skill, similar to camels, which boosts it through water
 - When used to attack, deals 3 damage
- Can be bred using pufferfish or a bucket of pufferfish, resulting in a baby nautilus
- Can be tempted with and fed raw fish, which heals it and boosts the growth of baby nautiluses
- If provoked, attacks with its dash attack
- Has a 5% chance to drop 1 nautilus shell when killed by a player
- When on land, takes suffocation damage
- Has a roaming restriction of 32 blocks
 - If equipped with a saddle, has a roaming restriction of 16 blocks
- When being ridden, has its own inventory screen that allows for equipping nautilus armor onto the mob
- Cannot be put into a boat

Zombie Nautilus

- Variant of nautilus that is undead
 - Can be protected from burning in sunlight by having nautilus armor equipped in its body slot
- Rarely naturally spawns, always with a drowned rider equipped with a trident
- Has a coral variant that spawns instead of the normal variant in warm oceans
- Cannot be bred
- Drops 0-3 rotten flesh on death when killed by a player
 - Maximum number of drops is increased by 1 per level of the Looting enchantment
- Does not take suffocation damage when out of water
- When being ridden, has its own inventory screen that allows for equipping nautilus armor onto the mob

- Cannot be put into a boat
- Despawns like other monsters, but becomes persistent when interacted with by a player

Parched

- Variant of skeleton
- Replaces about 50% of skeleton spawns on the surface in deserts
- Shoots arrows of weakness
- Has a slower rate of fire, like bogged
- Does not burn when exposed to sunlight
- Is immune to the Weakness effect
- Has a 50% chance to drop 0-1 arrows of weakness when killed by a player
 - Chance is increased per level of the Looting enchantment

Camel Husk

- Monster variant of camel that is undead (but is still passive)
- Replaces about 10% of husk spawns
- Spawns with the following riders:
 - Husk (primary) – spawns equipped with an iron spear in its main hand
 - Also drops 0-1 rabbit feet when killed, in addition to its regular drops
 - Maximum number of drops is increased by 1 per level of the Looting enchantment
 - Parched (secondary)
- Cannot be bred
- Can be tempted and healed using rabbit feet
- Drops 2-3 rotten flesh on death
 - Maximum number of drops is increased by 1 per level of the Looting enchantment
- Despawns like other monsters, but becomes persistent when interacted with by a player

New Items

Nautilus Armor

- Has 5 mineral tiers: Copper, Golden, Iron, Diamond, Netherite
- Can naturally generate in loot chests in buried treasures, shipwrecks, and underwater ruins, with the following chances:
 - Copper Nautilus Armor – ~10.8%
 - Iron Nautilus Armor – ~5.4%
 - Golden Nautilus Armor – ~2.7%
 - Diamond Nautilus Armor – ~1.1%
- Netherite nautilus armor is crafted in a smithing table with the following recipe: Netherite Upgrade (x1) + Diamond Nautilus Armor (x1) + Netherite Ingot (x1) → Netherite Nautilus Armor (x1)
- Can be equipped onto a nautilus or zombie nautilus to give it armor points (and knockback resistance if netherite)

Spear

- Has 7 mineral tiers:
 - Wooden
 - Deals a base attack damage of 1
 - Has a base attack speed of 1.54
 - Stone
 - Deals a base attack damage of 2
 - Has a base attack speed of 1.33
 - Can naturally generate in loot chests in underwater ruins, with a ~28.5% chance in small ruins, and a ~26.3% chance in large ruins
 - Copper
 - Deals a base attack damage of 2
 - Has a base attack speed of 1.18
 - Can naturally generate in loot chests in village weaponsmiths, with a ~30.6% chance
 - Iron
 - Deals a base attack damage of 3
 - Has a base attack speed of 1.05
 - Can naturally generate in loot chests in the following structures:
 - Buried treasure – ~16.7% chance
 - Village weaponsmith – ~22.9% chance
 - Golden
 - Deals a base attack damage of 1
 - Has a base attack speed of 1.05
 - Diamond
 - Deals a base attack damage of 4
 - Has a base attack speed of 0.95
 - Can naturally generate in the loot chests in the following structures:
 - Treasure variant of bastion remnant, as one of the following:
 - Randomly enchanted, with 80-100% durability – ~15.2% chance
 - No modifiers – ~15.2% chance
 - End city, as level 20-39 enchanted – ~12.7% chance
 - Netherite
 - Deals a base attack damage of 5
 - Has a base attack speed of 0.87
- Can be crafted with the following recipe: Mineral Item (x1) + Stick (x2) → Spear (x1)
- Does not deal any damage if used to attack an entity closer than 2 blocks
- Has an extended maximum reach, up to 4.5 blocks
- Has the following attacks:
 - Jab (primary) – is a low-damage attack with knockback
 - Can only be activated when the attack cooldown is at full charge
 - Can hit multiple entities at once
 - Does not destroy blocks

- Charge (secondary) – activated through holding the secondary action button (right-click on mouse and keyboard) and deals damage based on the spear's mineral tier, the player's view angle, and the velocity of both the player and target; can additionally deal knockback and dismount entities depending on how long the attack is held
 - While the attack is held, it goes through the following stages before returning to the item's idle state:
 - Engaged – can deal damage and knockback, and dismount entities
 - Tired – can deal damage and knockback, but cannot dismount entities
 - Indicated by the item rotating to a vertical position and shaking
 - Disengaged – can only deal damage
 - Indicated by the item being lowered, pointing downwards
 - Damage dealt is added on top of the attacker's base attack damage (is 1 for players)
 - Goes through its attack stages faster with higher mineral tiers
 - To dismount entities, the attacker needs to have one of the following speeds, depending on the item's mineral tier:
 - Wooden – ≥ 14 blocks/second
 - Golden – ≥ 10 blocks/second
 - Stone – ≥ 10 blocks/second
 - Copper – ≥ 9 blocks/second
 - Iron – ≥ 8 blocks/second
 - Diamond – ≥ 7.5 blocks/second
 - Netherite – ≥ 7 blocks/second
 - To deal knockback, the attacker needs to have a speed of at least 5.1 blocks/second
 - To deal damage, the attacker needs to have a speed of at least 4.6 blocks/second
- Cannot perform critical hits or sprint-knockback attacks
- Zombies, husks, zombified piglins, and piglins can naturally spawn equipped with a spear, and will use the spear's charge attack
 - When a non-player mob uses the spear's charge attack, the spear's attack range is half of that when used by players, but can deal damage, dismount entities and deal knockback at lower speeds than players
 - Any other mob equipped with a spear will use the spear's jab attack instead
 - Zombies and husks will spawn with an iron spear, while zombified piglins and piglins will spawn with a golden spear
- Piglins will admire and prefer golden spears as their weapon

Netherite Horse Armor

- Is crafted in a smithing table with the following recipe: Netherite Upgrade (x1) + Diamond Horse Armor (x1) + Netherite Ingot (x1) → Netherite Horse Armor (x1)
- Gives 19 armor points, 3 armor toughness, and 1 knockback resistance when equipped in the body slot

New Enchantments

Lunge

- Can only be applied to spears
- When applied to a spear, when its jab attack is used, propels the attacker horizontally in their view direction
 - Does not require the attack to actually hit anything
 - The view angle needs to be perfectly leveled horizontally to achieve maximum lunge distance
- Deals significant exhaustion to the player (depending on the enchantment's level) and 1 durability damage when used
- Can only be used if the player has at least 6 hunger points
- Has 3 levels, which have the following effects:
 - I – distance traveled is 0.458 blocks/tick; deals 4 exhaustion per use (costs 1 hunger point when having 0 saturation)
 - II – distance traveled is 0.916 blocks/tick; deals 8 exhaustion per use (costs 2 hunger points when having 0 saturation)
 - III – distance traveled is 1.374 blocks/tick; deals 12 exhaustion per use (costs 3 hunger points when having 0 saturation)

New Status Effects

Breath of the Nautilus

- Given by nautilus and zombie nautilus while mounted on them
 - After dismounting, the effect lasts up to 2 seconds
- When active, pauses consumption of the oxygen bar while underwater
- Is an ambient effect (its HUD icon has a blue outline and does not pulse when having a duration less than 10 seconds)

New Advancements

Mob Kabob

- Granted when hitting 5 mobs at once with a spear's charge attack
- Is a child of Monster Hunter

New Options

Preset

- Sets Quality & Performance options in the Video Settings screen to certain values, according to the selected preset
- Can be set to one of the following:
 - Fast – designed for low-end devices

- Fancy – designed for mid to high-end devices
 - Fabulous! – designed for high-end devices
 - Custom – represents a custom configuration of options
- Located in the Quality & Performance section of the Video Settings screen
- Set to Fancy by default
 - For existing users, is set to Custom, to preserve users' current configuration of options

See-Through Leaves

- If enabled, gaps can be seen through leaves
- Located in the Quality & Performance section of the Video Settings screen
- Enabled by default

Show Vignette

- Toggles the in-game dark vignette when the player is in a dark area and the red vignette when the player is close to a world border
- Located in the Preferences section of the Video Settings screen
- Enabled by default

Weather Effect Radius

- Sets the radius of the area that weather effects (rain and snow) are visible from the player
- Can be set to a value from 3 to 10 blocks, in 1 block intervals
- Located in the Quality & Performance section of the Video Settings screen
- Has the following default values by graphics preset:
 - Fast – 5 blocks
 - Fancy, Fabulous! – 10 blocks

Improved Transparency

- Toggles an experimental approach that uses screen shaders for drawing weather, clouds, and particles behind translucent blocks and water
- If enabled, will significantly impact GPU performance
- Located in the Quality & Performance section of the Video Settings screen
- Disabled by default for the Fast and Fancy graphics presets
- Enabled by default for the Fabulous! graphics preset

Texture Filtering

- Controls the method used to sample textures when they are viewed at an angle or from a distance
- Can be set to one of the following values:
 - None
 - RGSS – (Rotated Grid Super Sampling) uses a shader-based filtering method that improves texture clarity at a moderate performance cost

- Anisotropic – uses a hardware-based filtering method that provides the best clarity
 - When set to this value, the Anisotropic Filtering option can be modified
- Located in the Quality & Performance section of the Video Settings screen
- Has the following default values by graphics preset:
 - Fast – None
 - Fancy – RGSS
 - Fabulous! – Anisotropic

Anisotropic Filtering

- Significantly improves how smooth textures look, but impacts performance and significantly impacts video memory usage
 - May not be supported on all hardware
- Can only be modified if the Texture Filtering option is set to Anisotropic
- Can be set to one of the following values: 2x, 4x, 8x
- Located in the Quality & Performance section of the Video Settings screen
- Is set to 4x by default for the Fabulous! graphics preset
- Is not enabled by default for other graphics presets

Chunk Fade

- Sets how long, in seconds, chunks should fade in when they're first rendered
- Can be set to a value from None to 2.00 seconds, in 0.05 second intervals
 - If set to None, chunks do not fade in when first rendered
- Located in the Preferences section of the Video Settings screen
- Set to 0.75 seconds by default

Changes

- Updated zombie horses:
 - Now spawn naturally, in savanna and plains biomes, in darkness, except in Peaceful difficulty
 - When spawned naturally, spawns with a zombie rider that is equipped with an iron spear
 - The zombie rider will drop 0-1 red mushrooms on death
 - Maximum number of drops is increased by 1 per level of the Looting enchantment
 - Now count towards the hostile mob cap, unless it is tamed
 - Increased their base health to 25
 - Can now be tamed, in the same way as regular horses
 - Now burn when exposed to sunlight and avoid areas with sunlight
 - Can now be equipped with horse armor, which will protect it from burning in sunlight
 - Can now be tempted with and fed red mushrooms, which will heal them
 - Now drop 2-3 rotten flesh on death, instead of 0-2

- No longer despawn in Peaceful difficulty
- Updated the texture of zombie horses, their saddle, and their spawn egg item
- Updated horses, zombie horses, mules, donkeys, and camels:
 - No longer sink in water while being ridden
 - No longer panic when being controlled by their rider
- Spawn eggs are now sorted in the creative inventory by the following categories: farm animals, mounts, pets, wild animals, aquatic, miscellaneous, golems, villagers, undead, spiders, monsters, illagers, Nether, End
- Updated sounds:
 - Zombie horses now have the following new sounds:
 - An angry sound that plays when bucking off a mounted player or when a player fails to interact with it
 - An eating sound when they are fed
 - Anvil destroy, land, place, and use sounds now play at a slightly lower volume
- Updated the title screen panorama to show features from the Mounts of Mayhem drop
- Increased the amount of armor points that netherite armor items give when equipped in the body slot (applicable only to mobs that have this equipment slot; not applicable to players) to 19, from 11
- Updated advancements:
 - How Did We Get Here? now requires also having the Breath of the Nautilus status effect to be granted
 - Best Friends Forever can now be granted from taming a nautilus, zombie nautilus, or zombie horse
 - The Parrots and the Bats can now be granted from breeding nautiluses
 - Two by Two now requires also breeding nautiluses to be granted
 - Monster Hunter can now be granted by killing a zombie nautilus, zombie horse, camel husk, or parched
 - Monsters Hunted now requires also killing a zombie nautilus, zombie horse, camel husk, and parched to be granted
 - Oh Shiny can now be granted by giving a hostile piglin a golden nautilus armor or golden spear
- Items can no longer play their bob animation when their durability changes, such as when losing durability from an attack, or being repaired by the Mending enchantment
- Cloud movement is now the same for all players on a server
- Updated UIs:
 - Updated the Key Binds screen:
 - There is now a new Debug section, containing the following new key binds:
 - Toggle Overlay – bound to F3 by default
 - Debug modifier key – bound to F3 by default
 - Clear chat – bound to D by default
 - Copy data – bound to I by default
 - Copy location – bound to C by default
 - Cycle spectator – bound to N by default

- Debug crash – bound to C by default
 - Debug options – bound to F6 by default
 - Dump dynamic textures – bound to S by default
 - Dump version info – bound to V by default
 - Game mode switcher – bound to F4 by default
 - Reload chunks – bound to A by default
 - Reload resource packs – bound to T by default
 - Show advanced tooltips – bound to H by default
 - Show chunk boundaries – bound to G by default
 - Show hitboxes – bound to B by default
 - Start/stop profiling – bound to L by default
 - Toggle lost focus pause – bound to P by default
 - Profiling chart – bound to 1 by default
 - FPS charts – bound to 2 by default
 - Network charts – bound to 3 by default
- The Miscellaneous section now contains the following new key binds:
 - Toggle GUI – bound to F1 by default
 - Toggle Spectator Shader Effects – bound to F4 by default
- Menu sliders will now "stick" to the closest available value when releasing them after dragging with a mouse
- When using the keyboard to change slider values, pressing the arrow keys will now move the slider to the next available value, instead of by one pixel
- Renamed the "In F3" buttons in the Debug Options screen to "In overlay"
- Updated the Video Settings screen:
 - Options are now split into the following sections: Display, Quality & Performance, Preferences
 - Removed the following options: View Bobbing, Glint Speed, Glint Strength, FOV Effects, Distortion Effects
 - Still available in the Accessibility Settings screen
- The following screens now have search boxes:
 - Single Biome Customization – filters available biomes
 - Language – filters available languages
- Updated the Show Music Toast option:
 - Renamed it to Music Toast
 - Can now be set to "Pause Menu", which only shows the music toast in the Game Menu screen
- There is now the following new splash text: "One does not simply walk to the Far Lands"
- Rebound the following actions from the "cmd" key on Mac devices to the "ctrl" key: the input modifier to drop all items, the input modifier to include all data when using pick block, the input modifier to change the current GUI scale in the Video Settings screen, stopping the recipe book from cycling through variations of items
- Updated fog:

- Underwater biome fog color and fog distance is now blended based on the player's position between biomes, similarly to regular biome fog and sky colors
 - Environmental fog in the Nether is no longer dependent on the player's render distance and is now consistently applied from 10 to 96 blocks away from the player
- Chunks now fade into view instead of popping into view
- Bats will no longer spawn more frequently and in higher level levels between the dates October 20 and November 3
- The color of the sky during sunrise and sunset is now affected by rain and thunder weather

Technical Changelog

Technical Changelog

Additions

New Server Properties

management-server-allowed-origins

- Configures allowed origins for `Sec-WebSocket-Protocol` authentication in the Minecraft Server Management Protocol
 - If set to an empty string, `Sec-WebSocket-Protocol` authentication is effectively disabled
- Type: `string`
- Default value: empty string

Changes

- There is now a new `ClientFps` JFR event
- Experience packets that do not change the player's experience progress or level are no longer sent by the server
- Updated debug renderers:
 - More debug renderers are now available in the Debug Settings screen
 - Improved some existing renderers
 - Significantly improved the performance of debug renderers
- All blocks in the world now have mipmaps applied, if the Mipmap Levels option is not set to OFF
 - As a result, blocks that previously did not have mipmaps (such as different vines, rails, foliage (except for leaves), etc.) should look better from higher distances
- Improved performance that is expected to result in higher client FPS for most players
- There is now support for `KQueue` on OSX to improve network connection performance

Data Pack Changelog

Data Pack Changelog

v94.0

Additions

Environment Attribute

- Provides a data-driven way to control a variety of visual and gameplay systems
- Each environment attribute controls a specific visual or gameplay effect, such as the color of the sky (`visual/sky_color`) or whether water can be placed at a given location (`gameplay/water_evaporates`)
- Environment attribute values can be provided by environment attribute sources from the following, in order of low to high priority: dimensions, biomes, timelines (experimental), weather
 - The "effective" value of an environment attribute (i.e. what will actually show in-game) will be some combination of the values provided by each source, according to its priority
 - Example: If the `overworld` dimension provides a sky color of green and the `plains` biome provides a sky color of red, the sky color will be red when the player is in a plains biome, while the sky will be green when the player is in any other biome in the Overworld dimension
 - When an environment attribute source provides an environment attribute, it can:
 - Override the value
 - Apply a modifier to a previous value
 - Environment attribute modifiers will apply a modifier to a value provided by a source with lower priority
 - Example: If the `overworld` dimension provides a water fog radius of 96.0 blocks and the `plains` biome modifies the water fog radius by multiplying it by 0.85, the water fog radius will be $96.0 * 0.85 = 81.6$ blocks when the player is in a plains biome, while the water fog radius will be 96.0 blocks when the player is in any other biome in the Overworld dimension
 - The kinds of modifiers depends on the type of environment attribute; the most basic modifier, supported by every environment attribute, is the `override` modifier, which purely behaves as an override of the preceding value
 - Is also the default modifier, if not specified
 - For some environment attributes, after modifiers have been applied, interpolation is then performed on values from a source, to allow for a smooth transition between two or more values (such as when moving between two biomes)

- Example: If the `plains` biome provides a sky color of red and the `desert` biome provides a sky color of yellow, then the sky color will gradually shift from red to yellow as the player moves from a plains biome to a desert biome
 - Other attributes, such as `water_evaporates`, represent discrete values and will not be smoothly transitioned (only the biome exactly at a subject position will be considered)
 - Smooth transitions between biomes are based on the biomes within an 8 block radius of the player's camera
 - Biomes that occupy a larger portion of that radius and are closer to the camera will have a stronger influence on the final interpolated value
- Environment attributes can be specified in a new `attributes` field in both dimension type and biome definitions, which maps environment attributes to their values
 - Type: `{ environment attribute id : <value> }`
 - Type of `value` is dependent on specified environment attribute id
 - In-line values will always be assumed to use the `override` modifier
 - Values can also be of type `object`, with the following fields:
 - (optional) `modifier` – the modifier to use
 - Accepted values depend on the value type of the environment attribute
 - `override` is always an accepted value
 - Default value: `override`
 - `argument` – the modifier argument
 - Type is dependent on the modifier
 - How the argument is used also depends on the modifier
 - Example of in-line values: `"attributes": {
 "minecraft:visual/fog_color": "#ffaa00",
 "minecraft:gameplay/water_evaporates": true
 }`
 - Example of value modifier: `"attributes": {
 "minecraft:visual/water_fog_end_distance": {
 "modifier": "multiply",
 "argument": 0.85
 }
 }`
- Common data types of environment attributes:
 - RGB color – can be one of the following:
 - Hex color RGB string, in the format `#rrggbb`
 - `float[3]` array, in the format `[r, g, b]`, with each value having range `[0, 1]`
 - `integer`, in packed RGB format
 - ARGB color – can be one of the following:

- Hex color ARGB string, in the format `#aarrggbb`
 - `float[4]` array, in the format `[a, r, g, b]`, with each value having range `[0, 1]`
 - integer, in packed ARGB format
- Particle options – a full definition of a particle, including any type-specific properties (like in the `particle` command)
 - Example: {


```

"minecraft:block_crumble",
"block_state": {
  "Name": "minecraft:dirt"
}
          
```
- Mob activity – an id from the built-in `activity` registry; can be one of the following values: `minecraft:core`, `minecraft:idle`, `minecraft:work`, `minecraft:play`, `minecraft:rest`, `minecraft:meet`, `minecraft:panic`, `minecraft:raid`, `minecraft:pre_raid`, `minecraft:hide`, `minecraft:fight`, `minecraft:celebrate`, `minecraft:admire_item`, `minecraft:avoid`, `minecraft:ride`, `minecraft:play_dead`, `minecraft:long_jump`, `minecraft:ram`, `minecraft:tongue`, `minecraft:swim`, `minecraft:lay_spawn`, `minecraft:sniff`, `minecraft:investigate`, `minecraft:roar`, `minecraft:emerge`, `minecraft:dig`
- Common modifiers of environment attributes:
 - Modifiers on boolean values: `override`, `and`, `nand`, `or`, `nor`, `xor`, `xnor`
 - Modifiers on float values: `override`, `add`, `subtract`, `multiply`, `minimum`, `maximum`
 - Modifiers on RGB color values:
 - `override`
 - Argument type: ARGB color
 - `add` – component-wise additive color blending
 - Argument type: RGB color
 - `subtract` – component-wise subtractive color blending
 - Argument type: RGB color
 - `multiply` – component-wise multiplicative color blending
 - Argument types:
 - RGB color
 - ARGB color – the alpha component is multiplied independently, just like the other color channels
 - `blend_to_gray` – modifies a color by taking its grayscale form, applying a brightness modifier, and mixing with this using some factor

- Calculation is the following: `gray = brightness * (0.3 * red + 0.59 * green + 0.11 * blue)`, `result = lerp(factor, subject, [gray, gray, gray])`
 - Argument type: `object`
 - Fields of argument:
 - `brightness` – the multiplier to apply to the grayscale value
 - Type: `float`
 - Range: `[0.0, 1.0]`
 - `factor` – the factor to mix with
 - Type: `float`
 - Range: `[0.0, 1.0]`
 - Modifiers on ARGB color values:
 - All modifiers on RGB color values
 - `alpha_blend` – traditional alpha blending that might be seen in image-editing software
 - Argument type: ARGB color
 - If the alpha component of the argument is set to 1, the modifier will behave as an override, with no blending
- Available environment attributes:
 - `visual/fog_color` – sets the color of fog (when the camera is not submerged in another substance, like water)
 - The final value is also affected by the time of day, weather, and status effects
 - Value type: RGB color
 - Supported modifiers: RGB color modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: `#000000`
 - Replaces the `effects.fog_color` field of biome definitions
 - `visual/fog_start_distance` – sets the distance, in blocks, from the camera at which fog starts to have an effect (when the camera is not submerged in another substance)
 - If set to a negative value, the fog will start out with density as if it had started that many blocks behind the camera
 - The final value is also affected by weather
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: `0.0`

- `visual/fog_end_distance` – sets the distance, in blocks, from the camera at which fog reaches its maximum density (when the camera is not submerged in another substance)
 - The final value is also affected by weather
 - Value type: `float`
 - Value must be non-negative
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: `1024.0`
- `visual/water_fog_color` – sets the color of fog when the camera is submerged in water
 - The final value is also affected by the time of day, weather, and status effects
 - Value type: RGB color
 - Supported modifiers: RGB color modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Replaces the `effects.water_fog_color` field of biome definitions
- `visual/water_fog_start_distance` – sets the distance, in blocks, from the camera at which underwater fog starts to have an effect
 - If set to a negative value, the fog will start out with density as if it had started that many blocks behind the camera
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: `-8.0`
- `visual/water_fog_end_distance` – sets the distance, in blocks, from the camera at which underwater fog reaches its maximum density
 - The final value is also modified by how long the player has been underwater
 - Value type: `float`
 - Value must be non-negative
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Replaces the `has_closer_water_fog` biome tag
- `visual/sky_color` – sets the color of the sky
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - The final value is also affected by the time of day and weather
 - Value type: RGB color

- Supported modifiers: RGB color modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: #000000
 - Replaces the `effects.sky_color` field of biome definitions
- `visual/sky_fog_end_distance` – sets the distance, in blocks, from the camera at which fog that affects the sky reaches its maximum density (when the camera is not submerged in another substance)
 - The attribute's effects are restricted by the value of the Render Distance option
 - Value type: `float`
 - Value must be non-negative
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 512.0
- `visual/sunrise_sunset_color` – sets the color and intensity of the sunrise and sunset effects in the sky
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - If the color specified is fully transparent, no sunrise or sunset will be rendered
 - Value type: ARGB color
 - Supported modifiers: ARGB color modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: #00000000
- `visual/sun_angle` – sets the angle, in degrees, of the sun in the sky, clockwise from east to west (with 0 being directly up)
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 0.0
- `visual/moon_angle` – sets the angle, in degrees, of the moon in the sky, clockwise from east to west (with 0 being directly up)
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated

- Resolved at the camera's position
 - Default value: 0.0
- `visual/moon_phase` – sets the phase of the moon in the sky
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - Accepted values: `full_moon`, `waning_gibbous`, `third_quarter`, `waning_crescent`, `new_moon`, `waxing_crescent`, `first_quarter`, `waxing_gibbous`
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the camera's position
 - Default value: `full_moon`
- `visual/star_angle` – sets the angle, in degrees, of the star in the sky, clockwise from east to west (with 0 being directly up)
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 0.0
- `visual/star_brightness` – sets the brightness of stars in the sky
 - Is only applied if the dimension type specifies the `overworld` skybox rendering type
 - If set to 0.0, stars are fully hidden
 - The value 0.5 represents normal brightness
 - Value type: `float`
 - Value range: `[0.0, 1.0]`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 0.0
- `visual/sky_light_color` – sets the visual color of sky light
 - For blocks with a sky light level of 0, this attribute has no effect
 - Is multiplied by the value of the `visual/sky_light_factor` attribute to get the final sky light color
 - The value is passed into the `lightmap.fsh` shader as `SkyLightColor`
 - Value type: RGB color
 - Supported modifiers: RGB color modifiers
 - Is interpolated
 - Resolved at the camera's position

- Default value: #ffffff
- `visual/sky_light_factor` – sets the visual brightness of sky light
 - The value is passed into the `lightmap.fsh` shader as `SkyFactor`
 - Does not control the corresponding gameplay effect, is instead controlled by the `gameplay/sky_light_level` attribute
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 1.0
- `visual/cloud_color` – sets the color of the clouds
 - If the color specified is fully transparent, clouds are entirely disabled and happy ghosts will not regenerate faster when at cloud height
 - Value type: ARGB color
 - Supported modifiers: ARGB color modifiers
 - Is interpolated
 - Resolved at the camera's position for rendering, or at the position of a happy ghost for regeneration
 - Default value: #00000000
- `visual/cloud_fog_end_distance` – sets the distance, in blocks, from the camera at which fog that affects clouds reaches its maximum density (when the camera is not submerged in another substance)
 - The attribute's effects are restricted by the value of the Cloud Distance option
 - Value type: `float`
 - Value must be non-negative
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position
 - Default value: 2048.0
- `visual/cloud_height` – sets the height at which all clouds appear
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the camera's position for rendering, or at the position of a happy ghost for regeneration
 - Default value: 192.33
 - Replaces the `cloud_height` field of dimension type definitions
- `visual/default_dripstone_particle` – sets the default particle to be dripped from pointed dripstone blocks when no fluid is placed above them
 - Value type: Particle options
 - Supported modifiers: `override`
 - Is not interpolated

- Resolved at the position of the pointed dripstone block
- Default value:
 - { "type": "minecraft:dripping_dripstone_water" }
- Replaces the `ultrawarm` field of dimension type definitions
- `visual/ambient_particles` – controls how ambient particles are spawned around the camera
 - Value type: `object[]`
 - Fields of each value element:
 - `particle` – the particle to spawn
 - Type: particle definition
 - `probability` – the probability to spawn the particle in an empty space when randomly ticked
 - Type: `float`
 - Range: `[0.0, 1.0]`
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the camera's position
 - Default value: `[]`
 - Replaces the `effects.particle` field of biome definitions
- `audio/background_music` – controls how background music is played
 - Value type: `object`
 - Value fields:
 - (optional) `default` – the default background music to play
 - Type: `object`
 - Fields:
 - `sound` – the music track to play
 - Type: sound event
 - `min_delay` – the minimum delay, in ticks, between tracks
 - Type: `integer`
 - Must be non-negative
 - `max_delay` – the maximum delay, in ticks, between tracks
 - Type: `integer`
 - Must be non-negative
 - (optional) `replace_current_music` – whether this music track can replace whatever is currently playing
 - Type: `boolean`
 - Default value: `false`
 - (optional) `underwater` – background music to play while the player is underwater

- Has the same format as the `default` field
- (optional) `creative` – background music to play while the player is in Creative mode
 - Has the same format as the `default` field
- Supported modifiers: `override`
- Is not interpolated
- Resolved at the camera's position
- Default value: `{}`
- Replaces the `effects.music` field of biome definitions
- `audio/music_volume` – sets the volume at which music should play
 - Any music currently playing will fade over time to this volume
 - Value type: `float`
 - Value range: `[0.0,1.0]`
 - Supported modifiers: `float` modifiers
 - Is not interpolated
 - Resolved at the camera's position
 - Default value: `1.0`
 - Replaces the `effects.music_volume` field of biome definitions
- `audio/ambient_sounds` – controls how ambient sounds are played around the camera
 - Value type: `object`
 - Value fields:
 - (optional) `loop` – a sound to be continually looped
 - Type: `sound event`
 - (optional) `mood` – a sound that will be randomly played, based on surrounding darkness
 - Type: `object`
 - Fields:
 - `sound` – the sound to play
 - Type: `sound event`
 - `tick_delay` – the number of ticks between mood sounds, assuming a light level of 0
 - Type: `integer`
 - Must be non-negative
 - `block_search_extent` – the block radius in which light levels are sampled
 - Type: `integer`
 - Must be non-negative
 - `offset` – an additional distance offset to apply to sounds produced
 - Type: `double`

- (optional) `additions` – sounds that will be randomly played, in addition to other ambient sounds
 - Type: `object[]`
 - Fields of each element:
 - `sound` – the sound to play
 - Type: sound event
 - `tick_chance` – the probability within a tick to play the sound
 - Type: `float`
 - Range: `[0.0, 1.0]`
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the camera's position
 - Default value: `{ }`
 - Replaces the following fields of the `effects` field of biome definitions: `ambient_sound`, `mood_sound`, `additions_sound`
- `audio/firefly_bush_sounds` – if set to `true` and a firefly bush block is not below an opaque block, the block will produce idle sounds
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the firefly bush block
 - Default value: `false`
- `gameplay/sky_light_level` – sets the effective light level of the sky used by mechanics such as mob spawning or daylight detector blocks
 - Example: A block fully exposed to the sky will be considered to have a light level of exactly this value, while a block deep in a cave will not be affected at all
 - Does not control the corresponding visual effect, is instead controlled by the `visual/sky_light_factor` attribute
 - Cannot be specified for a biome, as it is resolved for a whole dimension
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Default value: `15.0`
- `gameplay/can_start_raid` – if set to `false`, a raid cannot be started by a player with the Raid Omen effect
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position that the raid would be started
 - Default value: `true`

- Replaces the `has_raids` field of dimension type definitions
- `gameplay/water_evaporates` – if set to `true`, water cannot be placed from a water bucket, melting ice, or a wet sponge drying out when placed, and mud will not convert to clay when there is pointed dripstone 2 blocks below the mud
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the water interaction
 - Default value: `false`
 - Replaces the `ultrawarm` field of dimension type definitions
- `gameplay/bed_rule` – controls how a bed can be used to sleep and set the player's respawn point
 - Value type: `object`
 - Value fields:
 - `can_sleep` – controls how the bed can be used to sleep
 - Accepted values:
 - `always` – the bed can always be used to sleep, if the bed is not obstructed and there are no monsters nearby
 - `when_dark` – the bed can only be used to sleep when the global skylight level is less than 4
 - `never` – the bed can never be used to sleep
 - `can_set_spawn` – controls how the bed can be used to set the player's respawn point
 - Has the same format as the `can_sleep` field
 - (optional) `explodes` – if set to `true`, the bed will explode when interacted with
 - Type: `boolean`
 - Default value: `false`
 - (optional) `error_message` – the message to show to the player if they are unable to sleep or set their respawn point
 - Type: `text component`
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the head position of the bed block
 - Default value: `{"can_sleep": "when_dark", "can_set_spawn": "always", "error_message": {"translate": "block.minecraft.bed.no_sleep"}}`
 - Replaces the `bed_works` field of dimension type definitions
- `gameplay/respawn_anchor_works` – sets whether respawn anchors can be used to set the player's respawn point

- If set to `false`, the respawn anchor will explode once charged
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the respawn anchor block
 - Default value: `false`
 - Replaces the `respawn_anchor_works` field of dimension type definitions
- `gameplay/nether_portal_spawns_piglin` – sets whether nether portal blocks can spawn zombified piglins
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the nether portal block
 - Default value: `false`
- `gameplay/fast_lava` – sets whether lava should spread faster and further, and have a stronger pushing force on entities when flowing
 - Cannot be specified for a biome, as it is resolved for a whole dimension
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Default value: `false`
 - Replaces the `ultrawarm` field of dimension type definitions
- `gameplay/increased_fire_burnout` – sets whether fire blocks burn out more rapidly than normal
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the burning fire block
 - Default value: `false`
 - Replaces the `increased_fire_burnout` biome tag
- `gameplay/piglins_zombify` – sets whether piglins and hoglins zombify
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the zombifying entity
 - Default value: `true`
 - Replaces the `piglin_safe` field of dimension type definitions
- `gameplay/snow_golem_melts` – sets whether a snow golem should continuously take damage
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers

- Is not interpolated
 - Resolved at the position of the snow golem entity
 - Default value: `false`
 - Replaces the `snow_golem_melts` biome tag
- `gameplay/eyeblossom_open` – controls the behavior of eyeblossom blocks
 - Accepted values:
 - `false` – open eyeblossom blocks will eventually close by random block ticks
 - `true` – closed eyeblossom blocks will eventually open by random block ticks
 - `default` – open and closed eyeblossom blocks will stay in their current state
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the position of the eyeblossom block
 - Default value: `default`
- `gameplay/turtle_egg_hatch_chance` – sets the chance that a turtle egg block will switch to its next hatching state when randomly ticked
 - Value type: `float`
 - Value range: `[0.0, 1.0]`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the position of the turtle egg block
 - Default value: `0.0`
- `gameplay/creaking_active` – controls the behavior of creaking heart blocks (and their corresponding creaking entity)
 - If set to `true`, creaking heart blocks will become active
 - If set to `false`, creaking heart blocks will enter their dormant state
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the creaking heart block
 - Default value: `false`
- `gameplay/surface_slime_spawn_chance` – sets the additional chance rolled when a natural slime entity spawn attempt occurs in a biome contained in the `allows_surface_slime_spawns` tag
 - Value type: `float`
 - Value range: `[0.0, 1.0]`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the position of the slime spawn attempt
 - Default value: `0.0`

- `gameplay/cat_waking_up_gift_chance` – sets the chance that a cat entity lying on the bed block that its player owner is also lying on will drop a gift (from the `gameplay/cat_morning_gift` loot table) when the player wakes up
 - Value type: `float`
 - Supported modifiers: `float` modifiers
 - Is interpolated
 - Resolved at the position of the cat entity
 - Default value: `0.0`
- `gameplay/bees_stay_in_hive` – if set to `true`, bee entities will try to navigate to their hives and will not exit unless the bee hive block is broken or next to a fire block
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the bee entity or the bee hive block position
 - Default value: `false`
- `gameplay/monsters_burn` – if set to `true`, monster entities will burn while exposed to the sky
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the monster entity
 - Default value: `false`
- `gameplay/can_pillager_patrol_spawn` – if set to `true`, pillager patrols can spawn, if the global sky light level is greater than 11
 - Value type: `boolean`
 - Supported modifiers: `boolean` modifiers
 - Is not interpolated
 - Resolved at the position of the patrol spawn attempt
 - Default value: `true`
 - Replaces the `without_patrol_spawns` biome tag
- `gameplay/villager_activity` – controls the default AI activity for adult villager entities
 - Not all activities are supported by adult villager AI – if an unsupported one is chosen, the adult villager entity will do nothing
 - Supported activities: `core`, `hide`, `idle`, `meet`, `panic`, `pre_raid`, `raid`, `rest`, `work`
 - Value type: `mob activity`
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the position of the adult villager entity

- Default value: `minecraft:idle`
- `gameplay/baby_villager_activity` – controls the default AI activity for baby villager entities
 - Not all activities are supported by baby villager AI – if an unsupported one is chosen, the baby villager entity will do nothing
 - Supported activities: `core`, `hide`, `idle`, `meet`, `panic`, `play`, `pre_raid`, `raid`, `rest`
 - Value type: mob activity
 - Supported modifiers: `override`
 - Is not interpolated
 - Resolved at the position of the baby villager entity
 - Default value: `minecraft:idle`

Slot Source

- Allows for the grouping and filtering of inventory slots, in loot contexts
- A slot source definition can be specified as the following types:
 - `object` – fields:
 - `type` – slot source type
 - Type: `string`
 - Additional fields are specified depending on the slot source type
 - `slot source[]` – a list of slot sources that will have their resulting selections merged into one (see the `group` slot source type)
- Available slot source types:
 - `empty` – empty selection containing no slots
 - `group` – merges several slot sources into one, with the resulting selection containing all slots from each provided slot source
 - If a slot is included in more than one slot source, it will be repeated in the resulting slot source (example: `[a,b] + [c,a] → [a,b,c,a]`)
 - The following additional fields are specified for the slot source with this type:
 - `terms` – slot sources to join together
 - Type: `slot source[]`
 - Can alternatively be specified in-line as a list of slot sources
 - `slot_range` – selects slots within a slot range from the inventory of an entity or block entity
 - Mirrors the behavior of the `from` parameter of the `item` command
 - The following additional fields are specified for the slot source with this type:
 - `source` – the entity or block entity from which the slots will be sourced, from loot context
 - Accepted values: `block_entity`, `this`, `attacking_entity`, `last_damage_player`,

- slots – the slot range
 - Type: string
 - Is in the following formats: <slot_type>, <slot_type>.<slot_number>
 - Examples: armor.chest, container.*

- `contents` – selects all non-empty slots from a specified inventory component of one or more items
 - If no item is stored in the inventory, then the resulting selection will be empty
 - The following additional fields are specified for the slot source with this type:
 - `component` – the inventory component of the targeted items to target
 - Accepted values: `minecraft:bundle_contents`, `minecraft:charged_projectiles`, `minecraft:container`
 - `slot_source` – a slot source that contains slots with the item(s) to target
 - If the slot source includes more than one item with the specified inventory component, then the resulting selections will be merged identically (like the `group slot source` type)
 - Example: `Bundle [a,b] + Shulker Box [c,d] → [a,b,c,d]`
 - Type: slot source
- `filtered` – applies a filter to the selected slots, excluding any non-matching slots from the resulting selection
 - The following additional fields are specified for the slot source with this type:
 - `item_filter` – an item predicate to match against the items in each slot
 - Type: item predicate
 - `slot_source` – the slot source to filter
 - Type: slot source
- `limit_slots` – limits the number of slots provided, with the resulting selection containing at most a specified number of slots
 - Any slots bringing the number of slots above the specified limit will be excluded, in order of inclusion
 - Example: `[a,b,c,d] → [a,b,c]`, if the limit is set to 3
 - The following additional fields are specified for the slot source with this type:

- `limit` – the maximum number of slots to include in the resulting selection
 - Type: integer
 - `slot_source` – the slot source to limit
 - Type: slot source
- Example of a slot source that selects every slot with more than 16 items from the hotbar and armor slots of `this` entity in the loot context:

```
{
  "type": "minecraft:filtered",
  "item_filter": {
    "count": {
      "min": 16
    }
  },
  "slot_source": [
    {
      "type": "minecraft:slot_range",
      "source": "this",
      "slots": "hotbar.*"
    },
    {
      "type": "minecraft:slot_range",
      "source": "this",
      "slots": "armor.*"
    }
  ]
}
```

New Commands

stopwatch

- Creates a stopwatch that keeps track of real time
 - Does not depend on game ticks
 - Runs while the server/world is running, even when paused
 - Will eventually replace the usage of world borders to keep track of real time in a world
- Syntax:
 - `stopwatch create <id>` – creates a new stopwatch
 - Parameters:
 - `id` – the id to give to the new stopwatch
 - Type: namespaced stopwatch id
 - `stopwatch query <id> [<scale>]` – queries a given stopwatch, returning the queried value, optionally scaled
 - Parameters:
 - `id` – the id of the stopwatch to query

- Type: namespaced stopwatch id
- `scale` – scales and truncates the queried value
 - Type: integer
 - Default value: 1
- Example: `stopwatch query foo:bar 20` – returns the elapsed time in ticks (seconds scaled by 20)
- `stopwatch restart <id>` – restarts the elapsed time of a given stopwatch
 - Parameters:
 - `id` – the id of the stopwatch to restart
 - Type: namespaced stopwatch id
- `stopwatch remove <id>` – removes a given stopwatch
 - Parameters:
 - `id` – the id of the stopwatch to remove
 - Type: namespaced stopwatch id

New Data Components

use_effects

- Controls how the player behaves when using an item
- Type: object
- Fields:
 - (optional) `can_sprint` – whether the player can sprint while using the item
 - Type: boolean
 - Default value: false
 - (optional) `speed_multiplier` – the speed multiplier applied to the player while using the item
 - Type: float
 - Range: [0.0, 1.0]
 - Default value: 0.2
- Example: `use_effects={can_sprint:true,speed_multiplier:1.0}`

minimum_attack_charge

- Sets the minimum attack charge on the attack indicator required to attack with an item
 - If set to 0.0, no attack charge is required to attack with the item
 - If set to 1.0, a full attack charge is required to attack with the item
- Type: float
- Range: [0.0, 1.0]

damage_type

- Specifies the type of damage dealt by an item
- Type: damage type id
- Example: `damage_type='minecraft:spear'`

kinetic_weapon

- Enables a charge-type attack when using an item (primarily for spears), where, while being used, the damage is dealt along a ray every tick based on the relative speed of the attacker and target entities
 - Damage dealt is calculated according to the following formula:
`floor(relative_speed * velocity_multiplier)`, where
`relative_speed` is the difference of speed vectors of the attacker and target as projected onto the axis of the attacker's view vector
 - Any additional damage dealt from enchantments or attribute modifiers is added after the initial damage calculation
- Type: `object`
- Fields:
 - (optional) `delay_ticks` – the time, in ticks, required before the weapon is effective in attacking
 - Type: `integer`
 - Default value: `0`
 - `dismount_conditions` – the condition to apply the dismount effect of the attack
 - Type: `object`
 - `knockback_conditions` – the condition to apply the knockback effect of the attack
 - Type: `object`
 - `damage_conditions` – the condition to apply the damage effect of the attack
 - Type: `object`
 - (optional) `forward_movement` – the distance the item moves out of the player's hand during its attack animation
 - Type: `float`
 - Default value: `0.0`
 - (optional) `damage_multiplier` – the multiplier for the final damage from the relative speed
 - Type: `float`
 - Default value: `1.0`
 - (optional) `sound` – a sound event to play when the weapon is engaged
 - Type: `sound event`
 - (optional) `hit_sound` – a sound event to play when the attack hits an entity
 - Type: `sound event`
- Fields of the `dismount_conditions`, `knockback_conditions`, and `damage_conditions` fields:
 - `max_duration_ticks` – the number of ticks after which the condition is no longer checked
 - Starts after `delay_ticks` has elapsed
 - Type: `integer`

- (optional) `min_speed` – the minimum speed of the attacker, in blocks per second, along the direction that the attacker is looking
 - Type: `float`
 - Default value: `0.0`
 - (optional) `min_relative_speed` – the minimum relative speed between the attacker and target, in blocks per second, along the direction that the attacker is looking
 - Type: `float`
 - Default value: `0.0`
- Example:


```
kinetic_weapon={max_reach:5.0,forward_movement:1.0,delay_ticks:20,damage_conditions:{max_duration_ticks:60},knockback_conditions:{max_duration_ticks:40},dismount_conditions:{max_duration_ticks:20}}
```

piercing_weapon

- Enables a quick attack for an item (primarily for spears) that damages multiple entities along a ray
- Type: `object`
- Fields:
 - (optional) `deals_knockback` – whether the attack deals knockback
 - Type: `boolean`
 - Default value: `true`
 - (optional) `dismounts` – whether the attack dismounts the target
 - Type: `boolean`
 - Default value: `false`
 - (optional) `sound` – a sound event to play when attacking
 - Type: `sound event`
 - (optional) `hit_sound` – a sound event to play when the attack hits an entity
 - Type: `sound event`
- Example: `piercing_weapon={min_reach:0,max_reach:4,dismounts:true}`

swing_animation

- Specifies how the item swing animation is played when attacking or interacting using an item
- Type: `object`
- Fields:
 - (optional) `type` – the type of animation to play
 - Accepted values: `none`, `whack`, `stab`
 - Default value: `whack`
 - (optional) `duration` – the duration of the animation, in ticks
 - Type: `integer`
 - Default value: `6`

- Example: `swing_animation={type:'stab',duration:20}`

attack_range

- Enables a custom attack range when using the item for melee attacks
- Overrides the normal entity interaction range for a player
- Modifies the distance check for a mob's regular melee attack range (based on the distance of the target hitbox to the mob's hitbox) by adjusting the minimum and maximum padding for the attack hitbox
 - If the item also has the `kinetic_weapon` component, the mob's attack range will be applied in the same way that it applies to players
- Type: `object`
- Fields:
 - (optional) `min_range` – the minimum distance, in blocks, from the attacker to the target to be considered a valid attack
 - Type: `float`
 - Range: `[0.0, 64.0]`
 - Default value: `0.0`
 - (optional) `max_range` – the maximum distance, in blocks, from the attacker to the target to be considered a valid attack
 - Type: `float`
 - Range: `[0.0, 64.0]`
 - Default value: `3.0`
 - (optional) `hitbox_margin` – the margin applied to the target bounding box when checking for a valid hitbox collision
 - Type: `float`
 - Range: `[0.0, 1.0]`
 - Default value: `0.3`
 - (optional) `mob_factor` – the multiplier applied to the `min_range` and `max_range` fields when checking for valid distance when the item is used by a non-player mob
 - Type: `float`
 - Range: `[0.0, 2.0]`
 - Default value: `1.0`

New Damage Types

spear

- Dealt by attacks using a spear

New Enchantment Effect Components

post_piercing_attack

- Enchantment effects are applied after a piercing attack with the item

New Enchantment Entity Effect Types

apply_impulse

- Applies an impulse to the targeted entity
- The following additional fields are specified for the enchantment entity effect with this type:
 - `direction` – applies this vector as local coordinates (the same used by the `tp @s ^ ^ ^` command) onto the entity look vector
 - `coordinate_scale` – scales the resulting vector from `direction` by this vector on each axis in world space (X, Y, and Z)
 - `magnitude` – scales the resulting vector from `coordinate_scale` by this value
 - Type: level-based value

apply_exhaustion

- Applies exhaustion to the targeted entity
- Is effective only on players
- The following additional fields are specified for the enchantment entity effect with this type:
 - `amount` – the amount of exhaustion to apply
 - Type: level-based value

New Level-Based Value Types

exponent

- Raises a specified base to a specified power level
- The following additional fields are specified for a level-based value with this type:
 - `base` – the base of the exponent
 - Type: level-based value
 - `power` – the power of the exponent
 - Type: level-based value

New Loot Functions

discard

- Replaces the item stack with an empty one

New Game Rules

fire_spread_radius_around_player

- Controls the maximum distance, in blocks, that fire can spread around a player
 - Replaces the previous `doFireTick` and `allowFireTicksAwayFromPlayer` game rules
 - If set to 0, fire spreading is disabled
 - If set to -1, fire can spread without players around
- Value type: `integer`

- Value has a minimum of -1

New Loot Pool Entry Types

slots

- Is a singleton entry type
- Provides the items contained within specified slots to the loot table
- The following additional fields are specified for the singleton loot pool entry with this type:
 - `slot_source` – slot source describing where the items are located
 - Type: `slot source`

New Advancement Triggers

spear_mobs

- Triggers when a player hits a specified amount of entities with a kinetic weapon in a single use
- The following additional fields are specified for the `conditions` field of the advancement definition with this trigger:
 - (optional) `count` – the number of entities that are required to be hit in a single use of the kinetic weapon
 - Type: `integer`

New Tags

Timeline Tags

in_end

- Timelines in this tag are active in the End dimension
- Contains the following values: `#minecraft:universal`

in_nether

- Timelines in this tag are active in the Nether dimension
- Contains the following values: `#minecraft:universal`

in_overworld

- Timelines in this tag are active in the Overworld dimension
- Contains the following values: `#minecraft:universal,minecraft:day,minecraft:moon,minecraft:early_game`

universal

- Timelines in this tag are active in every dimension
- Contains the following values: `minecraft:villager_schedule`

New Block Tags

can_glide_through

- Climbable blocks in this tag can be glided through without stopping

- Contains the following values: `minecraft:vine`, `minecraft:twisting_vines`, `minecraft:twisting_vines_plant`, `minecraft:weeping_vines`, `minecraft:weeping_vines_plant`, `#minecraft:cave_vines`

New Entity Type Tags

burn_in_daylight

- Entity types in this tag will burn when exposed to daylight
- Contains the following values: `minecraft:skeleton`, `minecraft:stray`, `minecraft:wither_skeleton`, `minecraft:bogged`, `minecraft:zombie`, `minecraft:zombie_horse`, `minecraft:zombie_villager`, `minecraft:drowned`, `minecraft:zombie_nautilus`, `minecraft:phantom`

can_wear_nautilus_armor

- Entity types in this tag can equip nautilus armor
- Contains the following values: `minecraft:nautilus`, `minecraft:zombie_nautilus`

nautilus_hostiles

- Nautiluses will be hostile to entity types in this tag
- Contains the following values: `minecraft:pufferfish`

can_float_while_ridden

- Entity types in this tag can float in water while being ridden
- Contains the following values: `minecraft:horse`, `minecraft:zombie_horse`, `minecraft:mule`, `minecraft:donkey`, `minecraft:camel`, `minecraft:camel_husk`

New Item Tags

enchantable/lunge

- Items in this tag can be enchanted with the Lunge enchantment
- Contains the following values: `#minecraft:spears`

enchantable/melee_weapon

- Contains the following values: `#minecraft:swords`, `#minecraft:spears`

nautilus_bucket_food

- Contains the following values: `minecraft:pufferfish_bucket`, `minecraft:cod_bucket`, `minecraft:salmon_bucket`, `minecraft:tropical_fish_bucket`

nautilus_food

- Items in this tag can be used to feed a tamed nautilus or zombie nautilus

- Contains the following values: `#minecraft:fishes`,
`#minecraft:nautilus_bucket_food`

nautilus_taming_items

- Items in this tag can be used to tame a nautilus or zombie nautilus
- Contains the following values: `minecraft:pufferfish_bucket`,
`minecraft:pufferfish`

spears

- Contains the following values: `minecraft:diamond_spear`,
`minecraft:stone_spear`, `minecraft:golden_spear`,
`minecraft:netherite_spear`, `minecraft:wooden_spear`,
`minecraft:iron_spear`, `minecraft:copper_spear`

zombie_horse_food

- Items in this tag can be used to feed a tamed zombie horse
- Contains the following values: `minecraft:red_mushroom`

camel_husk_food

- Items in this tag can be used to feed a camel husk
- Contains the following values: `minecraft:rabbit_foot`

New Biome Tags

spawns_coral_variant_zombie_nautilus

- The coral variant of zombie nautilus can naturally spawn in biomes in this tag
- Contains the following values: `minecraft:warm_ocean`

Experimental

Zombie Nautilus Variant

- Defines variants of the zombie nautilus entity
- Defined in the `zombie_nautilus_variant` directory
- A zombie nautilus variant definition has the following fields:
 - `model` – the model to use
 - Accepted values: `normal`, `warm`
 - `asset_id` – the texture to use
 - Type: namespaced id
 - Resolves to the path `assets/<namespace>/textures/<id>.png`
 - `spawn_conditions` – the spawn conditions for the variant
 - Type: uniform variant selection rule

Timeline

- Specifies game behavior based on absolute day time (that which is set by the `time set` command), and can optionally specify a period over which it is repeated
- Is able to modify game behavior or visuals by acting as an environment attribute source to apply environment attribute modifiers
 - Is processed after dimension and biome environment attribute sources
- Defined in the `timeline` directory
- By default, has no effect, unless it is specified by a dimension type
- Can define environment attribute tracks, which specify how a modifier for a particular environment attribute changes over time, by defining keyframes
 - Depending on the attribute, interpolation may be used between each keyframe; if interpolation is not used, the previous keyframe value will always be selected
 - Unlike biome environment attribute sources, if a timeline uses a modifier instead of an override, interpolation is applied to the modifier arguments, instead of the final modified values
 - Examples:
 - A timeline with a period of 24000 might be set as the following:
 - At time = 0, the timeline provides `sky_color = #ff0000` (red)
 - At time = 1000, the timeline provides `sky_color = #ff0000` (red)
 - At time = 6000, the timeline provides `sky_color = #ff00ff` (magenta)
 - Between time = 0 and time = 1000, the sky color will be red
 - Between time = 1000 and time = 6000, the sky color will shift from red to magenta
 - From time = 6000 all the way until the timeline repeats and reaches time = 0 again, the sky color will slowly shift back to red
- A timeline definition has the following fields:
 - (optional) `period_ticks` – the duration, in ticks, over which the timeline will repeat
 - If not specified, the timeline will not repeat
 - Type: `integer`
 - (optional) `tracks` – map between environment attribute ids and environment attribute tracks
 - Type: { environment attribute id : environment attribute track }
- An environment attribute track definition has the following fields:
 - (optional) `ease` – used to ease the interpolation of the value between keyframes
 - Type: easing type
 - If the target attribute does not support interpolation, the easing mode will have no effect
 - Default value: `linear`
 - `keyframes` – keyframes of the track

- Type: `object[]`
- Fields of each element:
 - `ticks` – defines the tick (within the period) at which this keyframe's value will be active
 - Type: `integer`
 - Has a minimum value of 0
 - If the `period_ticks` field is specified in the timeline definition, then the maximum value is the value of that field
 - `value` – the modifier argument
 - How this value is used is dependent on the specified modifier
 - Type is dependent on the specified modifier
 - If no modifier is specified, or set to `override`, then the type of this field is the same as the environment attribute specified by the key in the parent map
- Must be ordered by each element's `ticks` field, in ascending order
- At most two elements can have the same value for the `ticks` field, which creates an immediate transition between the corresponding values
- (optional) `modifier` – the modifier to apply
 - Accepted values depend on the value type of the environment attribute specified by the key in the parent map
 - `override` is always an accepted value
 - Default value: `override`
- An easing type controls the interpolation between two keyframes by applying a curve
 - Can be one of the following types:
 - `string`
 - Accepted values: `constant`, `linear`, `in_back`, `in_bounce`, `in_circ`, `in_cubic`, `in_elastic`, `in_expo`, `in_quad`, `in_quart`, `in_quint`, `in_sine`, `in_out_back`, `in_out_bounce`, `in_out_circ`, `in_out_cubic`, `in_out_elastic`, `in_out_expo`, `in_out_quad`, `in_out_quart`, `in_out_quint`, `in_out_sine`, `out_back`, `out_bounce`, `out_circ`, `out_cubic`, `out_elastic`, `out_expo`, `out_quad`, `out_quart`, `out_quint`, `out_sine`
 - If set to `constant`, the previous keyframe is always selected
 - `object`
 - Fields:
 - `cubic_bezier` – specifies cubic bezier easing
 - Type: `float[4]`
 - Elements are in the format `[x1, y1, x2, y2]`
 - `x1` – x-coordinate of the first control point

- Range: [0.0, 1.0]
 - y1 – y-coordinate of the first control point
 - x2 – x-coordinate of the second control point
 - Range: [0.0, 1.0]
 - y2 – y-coordinate of the second control point
- Easings can be visualized with the help of the following tools: easings.net, cubic-bezier.com
- Example of definition:
 - {


```

          "period_ticks": 24000,
          "tracks": {
            "minecraft:gameplay/cat_waking_up_gift_chance":
          {
            "ease": "constant",
            "modifier": "maximum",
            "keyframes": [
              { "ticks": 362, "value": 0.0 },
              { "ticks": 23667, "value": 0.7 }
            ]
          }
        }
      
```
 - The timeline has a period of 24000 and has an attribute track that modifies the `gameplay/cat_waking_up_gift_chance` environment attribute
 - Because the easing is set to `constant`, between ticks 362 and 23667, the value is 0.0, while between ticks 23667 and 362, the value is 0.7
 - Like other environment attribute sources, the specified value is used as the argument to the specified modifier, which is then applied on top of sources with a lower priority
 - In this case, the modifier is set to `maximum`, so this timeline will only ever increase the value of the `gameplay/cat_waking_up_gift_chance` environment attribute to the value specified in the track
 -

Changes

- Updated data components:
 - Updated the `animation` field of `consumable`:
 - Renamed the `spear` animation to `trident`
 - Now plays a new animation when set to `spear`

- Items with `intangible_projectile` now show information about the component in their inventory tooltip
- food is now specified on the following items: `cod_bucket`, `salmon_bucket`, `pufferfish_bucket`, `tropical_fish_bucket`
- `use_effects` now has a new optional `interact_vibrations` field, which determines whether using an item triggers the `item_interact_start` and `item_interact_finish` game events
 - Type: `boolean`
 - Default value: `true`
- Updated entity data:
 - There is now a new `anger_end_time` field, which contains the time that the entity's anger ends, in game ticks
 - Type: `long`
 - Renamed the `AngryAt` field to `angry_at`
 - Removed the `AngerTime` field
- Updated predicates:
 - Data component predicates (used in the `predicates` field of `block`, `item`, and entity predicates) can now test for the existence of any data component, by specifying the key-value pair `<data_component>: {}`
 - The predicate check passes as long as the data component is present, regardless of its actual value
 - Existing predicates for specific data components are unchanged
 - Examples:
 - `{written_book_content:{author:"foo"}}` – existing format for matching books
 - `{written_book_content:{}}` – passes if the `written_book_content` data component is present, regardless of its value
 - `{instrument:{}}` – passes if the `instrument` data component is present, regardless of its value
 - The `flags` field of an entity predicate now has the following new optional fields:
 - `is_in_water` – tests whether the entity is touching water or a bubble column
 - Type: `boolean`
 - `is_fall_flying` – tests whether the entity is gliding with an elytra
 - Type: `boolean`
- Updated commands:
 - `execute` now has the following new syntax: `execute if|unless stopwatch <id> <range>`, which executes if the elapsed time of a specified stopwatch falls within a specified range
 - The stopwatch value is measured in seconds and has an accuracy of up to a millisecond

- New parameters:
 - `id` – the id of the stopwatch
 - Type: namespaced stopwatch id
 - `range` – the range to check, in seconds
 - Type: float range
 - Example: `/execute if stopwatch foo:bar ..10 run say Stopwatch foo:bar has not reached 10 seconds yet`
- Item predicates in command syntax
 (`<item>[predicate~{...},component={...}]`) now accept empty predicates for any data component
 - Similarly to data component predicates, empty entries will only check for the existence of the specified data component
 - Existing shorter syntax for checking data component existence is unchanged
 - Example: `*[instrument]` and `*[instrument~{}]` are equivalent
- Resource suggestions now list results in any namespace, instead of just within `minecraft`
- Updated the `time` parameter of `worldborder`:
 - Now specifies the number of ticks by default, instead of number of seconds
 - Seconds can still be specified by adding the `s` suffix
 - Can now specify in-game days by adding the `d` suffix
- Commands specifying the `blocks` texture atlas for item textures now need to reference the new `items` texture atlas instead (due to item textures being moved to this atlas)
- Updated dimension type definitions:
 - Now has a new optional `timelines` field, which specifies which timelines are active in the dimension
 - Types: timeline id, timeline id[], timeline tag id
 - Replaced the `effects` field with the following new fields:
 - `skybox` – specifies the skybox rendering type to use
 - Accepted values:
 - `none` – no skybox rendering
 - Corresponds to the `minecraft:nether` value of the previous `effects` field
 - `overworld` – Overworld skybox rendering
 - Corresponds to the `minecraft:overworld` value of the previous `effects` field
 - The following environment attributes are applied only when this value is set: `visual/sky_color`, `visual/sun_angle`,

- visual/sunrise_sunset_color,
 - visual/moon_phase, visual/moon_angle,
 - visual/star_angle,
 - visual/star_brightness
 - end – End skybox rendering
 - Corresponds to the `minecraft:end` value of the previous `effects` field
 - Default value: `overworld`
- `cardinal_light` – the direction of cardinal lighting that affects blocks
 - Accepted values:
 - default
 - Corresponds to the `minecraft:overworld` and `minecraft:end` values of the previous `effects` field
 - nether
 - Corresponds to the `minecraft:nether` value of the previous `effects` field
 - Default value: `default`
- Replaced the `fixed_time` field with a new optional `has_fixed_time` field, which specifies whether behaviors affected by fixed time (and not controlled through environment attributes) occur or not
 - Type: `boolean`
 - Default value: `false`
- Removed the following fields:
 - `ultrawarm` – replaced by the `gameplay/water_evaporates`, `gameplay/fast_lava`, and `visual/default_dripstone_particle` environment attributes
 - `bed_works` – replaced by the `gameplay/bed_rule` environment attribute
 - `respawn_anchor_works` – replaced by the `gameplay/respawn_anchor_works` environment attribute
 - `cloud_height` – replaced by the `visual/cloud_height` environment attribute
 - `piglin_safe` – replaced by the `gameplay/piglins_zombify` environment attribute
 - `has_raids` – replaced by the `gameplay/can_start_raid` environment attribute
 - `natural` – replaced by the `gameplay/nether_portal_spawns_piglin`, `gameplay/eyeblossom_open`, and `gameplay/creaking_active` environment attributes
- World border interpolation is now dependent on game ticks instead of real time

- As a result, the world border will no longer move while the game is paused and will respect changes made by the `tick` command
- Replaced the `modifier` field of the `filtered` loot function type with the following new fields:
 - `on_pass` – functions to run when the `item_filter` predicate passes
 - Type: loot function
 - `on_fail` – functions to run when the `item_filter` predicate fails
- Updated game rules:
 - Are now registered in a dedicated registry
 - Renamed the following:
 - `announceAdvancements` → `show_advancement_messages`
 - `commandBlocksEnabled` → `command_blocks_work`
 - `commandModificationBlockLimit` → `max_block_modifications`
 - `disableElytraMovementCheck` → `elytra_movement_check`
 - `disablePlayerMovementCheck` → `player_movement_check`
 - `disableRaid` → `raids`
 - `doDaylightCycle` → `advance_time`
 - `doEntityDrops` → `entity_drops`
 - `doImmediateRespawn` → `immediate_respawn`
 - `doInsomnia` → `spawn_phantoms`
 - `doLimitedCrafting` → `limited_crafting`
 - `doMobLoot` → `mob_drops`
 - `doMobSpawning` → `spawn_mobs`
 - `doPatrolSpawning` → `spawn_patrols`
 - `doTileDrops` → `block_drops`
 - `doTraderSpawning` → `spawn_wandering_traders`
 - `doVinesSpread` → `spread_vines`
 - `doWardenSpawning` → `spawn_wardens`
 - `doWeatherCycle` → `advance_weather`
 - `maxCommandChainLength` → `max_command_sequence_length`
 - `maxCommandForkCount` → `max_command_forks`
 - `naturalRegeneration` → `natural_health_regeneration`
 - `snowAccumulationHeight` → `max_snow_accumulation_height`
 - `spawnRadius` → `respawn_radius`
 - `spawnerBlocksEnabled` → `spawner_blocks_work`
 - Renamed remaining game rules from the previous camel-case style to snake-case
 - The values of the following are now inverted, to match their new names: `elytra_movement_check`, `player_movement_check`, `raids`
 - The following now have the following limits on their values:
 - `max_block_modifications` – minimum value of 1

- `max_command_forks` – minimum value of 1
 - `max_command_sequence_length` – minimum value of 0
 - `max_entity_cramming` – minimum value of 0
 - `max_snow_accumulation_height` – range of [1, 8]
 - `players_nether_portal_creative_delay` – minimum value of 0
 - `players_nether_portal_default_delay` – minimum value of 0
 - `players_sleeping_percentage` – minimum value of 0
 - `random_tick_speed` – minimum value of 0
 - `respawn_radius` – minimum value of 0
- Removed the following: `doFireTick`, `allowFireTicksAwayFromPlayers`
 - Replaced by the new `fire_spread_radius_around_player` game rule
- Replaced the `bool_rules` and `int_rules` of game test environment definitions with type `game_rules` with a new `rules` field, which is a map of game rule resource locations to game rule values
 - Type: {game rule resource location : <value>}
 - Type of value is either integer or boolean, depending on the key specified
- Updated the `effect` field of biome definitions:
 - The type of the following fields is now RGB color, instead of integer: `water_color`, `foliage_color`, `dry_foliage_color`, `grass_color`, `grass_color_modifier`
 - Removed the following fields:
 - `fog_color` – replaced by the `visual/fog_color` environment attribute
 - `water_fog_color` – replaced by the `visual/water_fog_color` environment attribute
 - `sky_color` – replaced by the `visual/sky_color` environment attribute
 - `particle` – replaced by the `visual/ambient_particles` environment attribute
 - `ambient_sound`, `mood_sound`, `additions_sound` – replaced by the `audio/ambient_sounds` environment attribute
 - `music` – replaced by the `audio/background_music` environment attribute
 - `music_volume` – replaced by the `audio/music_volume` environment attribute
- Updated tags:
 - The `is_player_attack` and `no_knockback` damage type tags now contain the value `minecraft:spear`
 - Updated enchantment tags:
 - `non_treasure` now contains the value `minecraft:lunge`

- tooltip_order now contains the value `minecraft:lunge`, between `minecraft:frost_walker` and `minecraft:sharpness`
- Updated entity type tags:
 - aquatic now contains the following values: `minecraft:nautilus`, `minecraft:zombie_nautilus`
 - cannot_be_pushed_onto_boats now contains the following values: `minecraft:nautilus`, `minecraft:zombie_nautilus`
 - can_breathe_under_water now contains the value `minecraft:nautilus`
 - can_equip_saddle now contains the following values: `minecraft:camel_husk`, `minecraft:nautilus`, `minecraft:zombie_nautilus`
 - can_wear_horse_armor now contains the value `minecraft:zombie_horse`
 - not_scary_for_pufferfish now contains the following values: `minecraft:nautilus`, `minecraft:zombie_nautilus`
 - zombies now contains the following values `minecraft:camel_husk`, `minecraft:zombie_nautilus`
 - skeletons now contains the value `minecraft:parched`
- Updated item tags:
 - enchantable/durability now contains the value `#minecraft:spears`
 - Updated enchantable/fire_aspect:
 - Now contains the value `#minecraft:enchantable/melee_weapon`
 - No longer contains the value `#minecraft:enchantable/sword`
 - Updated enchantable/sharp_weapon:
 - Now contains the value `#minecraft:enchantable/melee_weapon`
 - No longer contains the value `#minecraft:swords`
 - Renamed `enchantable/sword` to `enchantable/sweeping`
 - piglin_loved now contains the following values: `minecraft:golden_nautilus_armor`, `minecraft:golden_spear`
 - piglin_preferred_weapons now contains the value `minecraft:golden_spear`
- Removed the following biome tags:
 - has_closer_water_fog – replaced by the `visual/water_fog_end_distance` environment attribute
 - increased_fire_burnout – replaced by the `gameplay/increased_fire_burnout` environment attribute

- `plays_underwater_music` – replaced by the `underwater` value field of the `audio/background_music` environment attribute
- `snow_golem_melts` – replaced by the `gameplay/snow_golem_melts` environment attribute
- `without_patrol_spawns` – replaced by the `gameplay/can_pillager_patrol_spawn` environment attribute

Resource Pack Changelog

Resource Pack Changelog

v75.0

Additions

New Atlases

celestials

- Includes sprites for objects rendered in the sky, such as the sun and moon
- Includes all sprites from a new `textures/environment/celestial` folder

items

- Contains item textures, which have been split from the existing `blocks` atlas
- Does not have mipmaps
- Can be used for block and item models

New Item Sprites

`nautilus_spawn_egg`

`zombie_nautilus_spawn_egg`

`copper_nautilus_armor`

`iron_nautilus_armor`

`golden_nautilus_armor`

`diamond_nautilus_armor`

`netherite_nautilus_armor`

`wooden_spear`

`wooden_spear_in_hand` – for the wooden spear when held in hand

`stone_spear`

`stone_spear_in_hand` – for the stone spear when held in hand

`copper_spear`

`copper_spear_in_hand` – for the copper spear when held in hand

`iron_spear`

`iron_spear_in_hand` – for the iron spear when held in hand

`golden_spear`

`golden_spear_in_hand` – for the golden spear when held in hand

`diamond_spear`

`diamond_spear_in_hand` – for the diamond spear when held in hand

`netherite_spear`

`netherite_spear_in_hand` – for the netherite spear when held in hand

`netherite_horse_armor`

`leather_horse_armor_overlay` – overlay layer split from `leather_horse_armor`

New Entity Textures

nautilus/nautilus
nautilus/nautilus_baby
nautilus/zombie_nautilus
equipment/nautilus_body/copper
equipment/nautilus_body/iron
equipment/nautilus_body/gold
equipment/nautilus_body/diamond
equipment/nautilus_body/netherite
equipment/nautilus_saddle/saddle
equipment/zombie_nautilus_coral
equipment/horse_body/netherite
equipment/horse_body/leather_overlay – overlay layer split from
equipment/horse_body/leather

New GUI Sprites

container/slot/nautilus_armor
container/slot/spear – for the spear icon used in the smithing table screen
container/inventory/effect_background – nine-sliced sprite
container/inventory/effect_background_ambient – nine-sliced sprite
container/nautilus – for the nautilus inventory screen
container/slot/nautilus_armor_inventory – for the nautilus armor icon used in the
nautilus inventory screen

New Status Effect Sprites

breath_of_the_nautilus

New Sound Events

item.spear.hit
item.spear.use
item.spear.attack
item.spear_wood.hit
item.spear_wood.use
item.spear_wood.attack
item.spear.lunge_1
item.spear.lunge_2
item.spear.lunge_3
entity.baby_nautilus.ambient
entity.baby_nautilus.death
entity.baby_nautilus.eat
entity.baby_nautilus.hurt
entity.baby_nautilus.swim
entity.baby_nautilus.ambient_land

entity.baby_nautilus.death_land
entity.baby_nautilus.hurt_land
entity.nautilus.ambient
entity.nautilus.ambient_land
entity.nautilus.death
entity.nautilus.death_land
entity.nautilus.dash
entity.nautilus.dash_land
entity.nautilus.dash_ready
entity.nautilus.dash_ready_land
entity.nautilus.eat
entity.nautilus.hurt
entity.nautilus.hurt_land
entity.nautilus.swim
entity.nautilus.riding
entity.zombie_nautilus.ambient
entity.zombie_nautilus.ambient_land
entity.zombie_nautilus.death
entity.zombie_nautilus.death_land
entity.zombie_nautilus.dash
entity.zombie_nautilus.dash_land
entity.zombie_nautilus.dash_ready
entity.zombie_nautilus.dash_ready_land
entity.zombie_nautilus.eat
entity.zombie_nautilus.hurt
entity.zombie_nautilus.hurt_land
entity.zombie_nautilus.swim
item.nautilus_armor.equip
item.nautilus_armor.unequip
item.nautilus_saddle_equip
entity.zombie_horse.angry
entity.zombie_horse.eat
entity.parched.ambient
entity.parched.death
entity.parched.hurt
entity.parched.step
entity.camel_husk.ambient
entity.camel_husk.dash
entity.camel_husk.dash_ready
entity.camel_husk.death
entity.camel_husk.eat
entity.camel_husk.hurt
entity.camel_husk.saddle
entity.camel_husk.sit
entity.camel_husk.stand

```
entity.camel_husk.step
entity.camel_husk.step_sand
entity.parrot.imitate.camel_husk
entity.parrot.imitate.parched
entity.parrot.imitate.zombie_horse
entity.parrot.imitate.zombie_nautilus
```

New Item Models

`spear_in_hand` – model for the spear item when held in hand

New Shaders

New Core Shaders

`animate_sprite*`

- Consists of the following shaders:
 - `animate_sprite.vsh`
 - `animate_sprite_blit.fsh`
 - `animate_sprite_interpolate.fsh`
- Perform GPU-based sprite animations
 - Previously, sprite animations were performed per-tick on the CPU
 - For regular frame-based animations, `animate_sprite.vsh` and `animate_sprite_blit.fsh` are used to perform the draw
 - For interpolated animations, `animate_sprite.vsh` and `animate_sprite_interpolate.fsh` are used to perform the draw
- Use a new `SpriteAnimationInfo` uniform, which contains information on where the sprite should be drawn to, within the greater texture atlas

Changes

- Updated Unifont to version 17.0.01
- Updated textures:
 - The `texture` field of a `.mcmeta` file now has the following new fields:
 - (optional) `mipmap_strategy` – sets the method of mipmapping of the texture
 - Accepted values:
 - `mean` – averages both color and alpha between groups of 4 pixels for the current mipmap level to generate the next mipmap level pixel
 - Is the default strategy for most solid or translucent full block textures

- `dark_cutout` – similar to `mean`, but blends colors in a way that makes pixels bordering cutout pixels darker, simulating the darker interior of a block
 - Is the default strategy for leaves and mangrove roots textures
 - `cutout` – always generates a mipmap based on the original texture instead of the previous mipmap
 - Used for most cutout blocks that became mipmapped as of this version
 - `strict_cutout` – is a modification of `cutout` that uses a stricter alpha cutoff value, leading to the textures using this value disappearing at high mipmap values
 - Is used for flowers and similar block textures to avoid artifacts
 - `auto` – automatically picks the `mean` strategy for textures that do not contain fully transparent pixels, while picking `cutout` for those that do
- Default value: `auto`
- (optional) `alpha_cutoff_bias` – controls the alpha bias for cutout textures
 - Is only used for lower mipmap levels of cutout textures
 - Does not change the alpha of the first mipmap level, and has no effect on other textures
 - If set to a value greater than `1.0` or lower than `-1.0`, the texture will be fully opaque or fully transparent, respectively
 - Should be set to a higher value for textures that can become too transparent/thin at a distance, or to a lower value for textures that become too "opaque"
 - Example: The `kelp` texture uses a value of `0.1` to prevent it from becoming fully transparent at a distance
 - Type: `float`
 - Default value: `0.0`
- Glass, glass panes, beacons, and redstone dust now support translucent textures
- Moved the following to the new `textures/environment/celestial` folder, to the following relative paths:
 - `textures/environment/sun` → `sun`
 - `textures/environment/end_flash` → `end_flash`
- Split `textures/environment/moon_phases` into the following textures in the new `textures/environment/celestial` folder, to the following relative paths:
 - `moon/full_moon`
 - `moon/waning_gibbous`

- moon/third_quarter
 - moon/waning_crescent
 - moon/new_moon
 - moon/waxing_crescent
 - moon/first_quarter
 - moon/waxing_gibbous
- Split the item textures from the `blocks` atlas to the new `items` atlas
 - As a result, item models can now only use textures from either the `blocks` or `items` atlas (cannot be mixed) and block models can no longer use item textures (as all textures must be from the `blocks` atlas)
- A warning is now printed if any defined sprites in any atlases share a name
- Beacons no longer support translucent textures
- Textures used for still water and lava are now controlled by the `block/water_still` and `block/lava_still` textures, respectively
- Removed the following GUI sprites:
 - `container/inventory/effect_background_large`,
 - `container/inventory/effect_background_small`
- Updated models:
 - Item model definitions now have a new optional `swap_animation_scale` field, which indicates how fast the item model moves up and down when swapped in the hotbar with another item
 - Setting this to a higher value allows items that take more of the screen space to fully bob down before being swapped into the next item model
 - Type: `float`
 - Default value: `1.0`
 - Updated the `elements[].rotation` field of model definitions:
 - Now has the following new optional fields to rotate the element on more than one axis:
 - `x` – the rotation around the X axis, in degrees
 - Type: `float`
 - Default value: `0.0`
 - `y` – the rotation around the Y axis, in degrees
 - Type: `float`
 - Default value: `0.0`
 - `z` – the rotation around the Z axis, in degrees
 - Type: `float`
 - Default value: `0.0`
 - The element will be rotated around axes in the following order: X axis, Y axis, Z axis
 - The existing `axis` and `angle` fields can still be used to rotate the element on only one axis (if both fields are present, then the `x`, `y`, and `z` fields will be ignored)

- The value of the `angle` field is no longer restricted to the range `[-45.0, 45.0]`
- Updated shaders:
 - The `block` vertex and fragment shaders now copy the `terrain` vertex and fragment shaders and now handle ad-hoc blocks (e.g. blocks held by entities)
 - Updated uniforms:
 - There is now a new `ChunkSection` uniform, which is used by `terrain.vsh`
 - Replaces the previous `DynamicTransforms` uniform
 - `Globals` now contains camera coordinates
- Values of the `variants` field of block state dispatch definitions now have a new optional `z` field, which rotates the block model around the Z axis
 - Rotation around the Z axis is applied after the X and Y axes
 - Has the same format as the existing `x` and `y` fields

Minecraft Server Management Protocol Changelog

Minecraft Server Management Protocol

Changelog

v2.0.0

Additions

New Notifications

server/activity

- Indicates that a network connection to the server has been initiated
- Is rate-limited to 1 notification per 30 seconds

Changes

- There is now support for authentication from web browsers, which is performed by passing the token in the `Sec-WebSocket-Protocol` header when opening the WebSocket connection
 - Requests authenticating this way are subject to `Origin` header checks, which require configuration of allowed origins on the server-side using the new `management-server-allowed-origins` server property
 - **Example:** `Sec-WebSocket-Protocol: minecraft-v1, <token>`
- Updated the `typed_game_rule` and `untyped_game_rule` schemas:
 - The key is now of type game rule resource location
 - Changed the type of the `value` field to either `boolean` or `integer`, from `string`

Bug Fixes

Bug Fixes

- Fixed 97 [Mojira](#) issues:
 - [MC-2791](#) – (from version Minecraft 1.4.2) The player model in the inventory screen renders in the wrong orientation when it's not standing up straight
 - [MC-22882](#) – (from version Minecraft 1.6.2) Ctrl + Q doesn't work on Mac
 - [MC-53491](#) – (from version Minecraft 14w17a) The world border can change in size when the game is paused
 - [MC-54989](#) – (from version Minecraft 1.7.9) Sliders can be moved to positions between available values
 - [MC-73186](#) – (from version Minecraft 1.8.1-pre2) Gaps between the faces of item models and complex block models
 - [MC-80476](#) – (from version Minecraft 1.8.4) On macOS, the "drop item stack" shortcut conflicts with the "quit Minecraft" shortcut
 - [MC-114265](#) – (from version Minecraft 1.11.2) Mipmaps are too dark around transparent edges in textures (e.g. side of grass)
 - [MC-146862](#) – (from version Minecraft 19w13a) Long effect names cause text to appear outside the effect box
 - [MC-147718](#) – (from version Minecraft 1.12.1) F1, F3, F3+[char] combos cannot be rebound
 - [MC-149630](#) – (from version Minecraft 1.14) Some particles have very thin, vertical lines that flash in and out around the particles
 - [MC-159275](#) – (from version 1.14.4) Map player markers on maps of the nether don't rotate when doDaylightCycle is disabled
 - [MC-162573](#) – (from version 1.14.4) A white outline is rendered on composters' edges when viewed from far away
 - [MC-171688](#) – (from version 1.15.2) Invisibility status of LivingEntity is not updated when ActiveEffects are modified directly
 - [MC-199467](#) – (from version 1.16.2) Certain entity animations stop after they've existed in world for too long
 - [MC-195505](#) – (from version 1.16.1) Short grass is unaffected by mipmap levels
 - [MC-234358](#) – (from version 1.17.1) Moiré patterns / aliasing on certain objects when viewed from a distance
 - [MC-237158](#) – (from version 21w37a) Magma blocks can generate on the ceilings of caves below aquifers
 - [MC-238715](#) – (from version 1.17.1) The "minecraft:block.chest.locked" sound cannot be heard by other players when attempting to open locked blocks
 - [MC-241862](#) – (from version 1.17.1) It sometimes requires multiple keyboard inputs to change values on sliders
 - [MC-245895](#) – (from version 1.18.1) View Bobbing stops working after long elytra flight
 - [MC-248499](#) – (from version 1.18.1) Potion UIs displayed within the inventory don't have cyan outlines if the effects are granted by beacons or conduits

- [MC-263562](#) – (from version 1.20.1) World types in Realms backup info screen are untranslatable
- [MC-264151](#) – (from version 1.20.1) Glass blocks do not use mipmapping, but glass panes do
- [MC-266425](#) – (from version 23w44a) Recipes for new waxed copper blocks are not grouped
- [MC-267364](#) – (from version 1.20.4) Teleporting in the air is considered flying by server
- [MC-269295](#) – (from version 1.20.4) Jump bar progress renders unused pixels from dynamic texture atlas
- [MC-271729](#) – (from version 23w51a) Armadillo scute drops are not affected by the mob_drops game rule
- [MC-271938](#) – (from version 24w19a) Mace smash attack can push players in creative mode that are flying
- [MC-271941](#) – (from version 1.20.6) Music discs can sometimes spawn inside jukeboxes when ejected from them
- [MC-276382](#) – (from version 1.21.1) Leather horse armor leggings & helmet parts are untextured
- [MC-276445](#) – (from version 1.21.1) Highlighted text within the anvil and creative inventory interfaces renders blue making it difficult to read
- [MC-278742](#) – (from version 1.21.4) Creakings cannot be ridden by any entities using commands
- [MC-279076](#) – (from version 1.21.4) Certain mobs will always retarget the player upon reloading the chunks, regardless of distance or gamemode
- [MC-295949](#) – (from version 1.21.5) Flying through vines with an elytra cancels the flight
- [MC-296952](#) – (from version 1.21.5) F4 key to toggle shaders cannot be rebound
- [MC-297328](#) – (from version 1.21.5) Unloading and reloading an area causes invisible entities to be visible until you get close enough.
- [MC-297367](#) – (from version 1.21.5) Checkbox filled status isn't narrated
- [MC-298405](#) – (from version 1.21.6 Pre-Release 1) Text components in the "label" of input controls in dialogs don't support "hover_event"
- [MC-298767](#) – (from version 1.21.5) Piercing arrows phase through mobs that are in the same block
- [MC-298915](#) – (from version 1.21.6) Multishot crossbows loaded in survival or adventure mode show their secondary charged projectiles on a separate line in the tooltip
- [MC-298942](#) – (from version 1.21.6) Character body moves instead of the head when riding a happy ghast
- [MC-299136](#) – (from version 1.21.7 Release Candidate 1) Hyper Potions (Ian Tsuchiura) is not mentioned in the credits and splash texts
- [MC-299196](#) – (from version 1.21.6) Waypoints fade out when an advancement is granted

- [MC-299876](#) – (from version 1.21.8) Labels for booleans within dialogs are rendered in a slightly different shade of white than normal
- [MC-300642](#) – (from version 1.21.8) When texture atlases are large, gaps are rendered between blocks that have a low resolution
- [MC-300979](#) – (from version 25w33a) The movement and duration messages in the demo introduction screen now have a text shadow
- [MC-301271](#) – (from version 25w34b) Object text components do not render in some places unless there are text glyphs on the same line
- [MC-301311](#) – (from version 25w34b) The “Transfer Now” button no longer renders highlighted when the java realms information box is selected
- [MC-301424](#) – (from version 25w34b) entity_data item component detection is broken in resource packs
- [MC-301516](#) – (from version 25w35a) The mouse cursor doesn’t change to the hand shape when hovering over tabs in the “Create New World” menu and similar screens
- [MC-301517](#) – (from version 25w35a) The mouse cursor doesn’t change to the hand shape when hovering over arrows in the singleplayer, multiplayer, resource packs, and data packs menus
- [MC-301518](#) – (from version 25w35a) The mouse cursor doesn’t change to the hand shape when hovering over checkboxes
- [MC-301520](#) – (from version 25w35a) The mouse cursor doesn’t change to the hand shape when hovering over the difficulty lock button
- [MC-301527](#) – (from version 25w35a) The mouse cursor doesn’t change to the resize shape when scrolling in the advancements menu
- [MC-301557](#) – (from version 25w35a) The shading direction of the bolts on single and large copper chests does not match
- [MC-301595](#) – (from version 1.21.8) Music toasts briefly appear in the pause menu after a song has finished
- [MC-301632](#) – (from version 1.21.8) Comparators measuring jukeboxes do not update when a music disc is taken out of a jukebox if the music disc has already finished its song
- [MC-301754](#) – (from version 25w36a) The hand animation plays when right-clicking shelves with nothing in your hand
- [MC-301763](#) – (from version 25w36a) The outline of the “Invited” element within the realms “Players” tab renders above other tabs when selected and when a scroll bar is present
- [MC-301805](#) – (from version 25w36b) Selected item outline on statistics screen renders outside of scissor area
- [MC-301879](#) – (from version 25w36b) Double weathered and waxed weathered copper chests' bottom texture looks off compared to the other variants
- [MC-301988](#) – (from version 1.21.8) Buttons in the telemetry screen are not aligned properly in some languages
- [MC-302030](#) – (from version 25w37a) Setting commandBlocksEnabled to false prevents command blocks from being edited

- [MC-302071](#) – (from version 25w37a) Single and double copper chests have inconsistent side textures
- [MC-302111](#) – (from version 1.21.9 Pre-Release 1) Elements within the resource pack and data pack menus are not selected in order when using the TAB key
- [MC-302184](#) – (from version 1.21.9 Pre-Release 2) Copper golems in cave/void air never turn into statues
- [MC-302209](#) – (from version 1.21.9 Pre-Release 2) Debug renderers stop working when changing dimensions
- [MC-302246](#) – (from version 1.21.9 Pre-Release 2) Sprite object component does not render in server list depending on component tree
- [MC-302254](#) – (from version 1.21.9 Pre-Release 2) Cursor changes do not work for command suggestions
- [MC-302288](#) – (from version 1.21.9 Pre-Release 4) All shelves have the oak tan map color
- [MC-302325](#) – (from version 1.21.9 Release Candidate 1) When the player toggles a debug renderer using its hotkey (F3+B, F3+G) while in the debug options screen, the corresponding entry is not updated
- [MC-302338](#) – (from version 1.21.9 Release Candidate 1) The narrator button in the Accessibility Settings menu doesn't update upon pressing Ctrl+B
- [MC-302362](#) – (from version 1.21.9 Release Candidate 1) Clicking on "Singleplayer" or "Multiplayer" in the main menu then immediately clicking on a world or server joins it even when not clicking the play button
- [MC-302409](#) – (from version 1.21.9) Lightning that strikes waxed lightning rods do not deoxidize nearby unwaxed copper blocks
- [MC-302469](#) – (from version 1.21.9) Creepers at the player's exact position with an explosion radius of 0 set the player's velocity to NaN
- [MC-302477](#) – (from version 1.21.9) Loot tables stopped supporting SNBT as entity data
- [MC-302482](#) – (from version 1.21.9) Resource and data pack names can overlap selection boxes when the scroll bar is present
- [MC-302549](#) – (from version 1.21.9) Server main thread deadlock during respawn position resolution when the entire world spawn radius is filled with fluids
- [MC-302601](#) – (from version 1.21.9) A white outline is rendered on anvils' top texture when viewed from far away
- [MC-302695](#) – (from version 1.21.10) Can use the game mode switcher while the game is paused, and doing so unpauses the game
- [MC-302816](#) – (from version 1.21.10) When running NBT to SNBT conversion, the program waits for 1 minute after completion
- [MC-302854](#) – (from version 1.21.10) Glass is less transparent when viewed from far away
- [MC-303061](#) – (from version 1.21.10) The "minecraft:block.chest.locked" sound plays at the player position, not the chest's
- [MC-303381](#) – (from version 1.21.10) Game crashes when serializing a world spawn point with yaw or pitch values outside of rotation bounds

- [MC-303382](#) – (from version 1.21.10) Player spawn points with rotation values outside of bounds aren't serialized
- [MC-303395](#) – (from version 1.21.10) You cannot double-click on server regions within the realms select region menu to select them
- [MC-303396](#) – (from version 1.21.10) The search fields in the resource and data pack menus aren't automatically selected when opening the menus via keyboard navigation
- [MC-303755](#) – (from version 1.21.10) The mouse cursor doesn't change to the resize shape when hovering over the scroll bar in the villager and wandering trader trade UI
- [MC-303757](#) – (from version 1.21.10) The mouse cursor doesn't change to the proper shape when hovering over enchantments in the enchanting table UI
- [MC-303758](#) – (from version 1.21.10) The mouse cursor doesn't change to the hand shape when hovering over various elements in the recipe book
- [MC-303761](#) – (from version 1.21.10) The mouse cursor doesn't change to the proper shape when hovering over the output buttons or the scroll bar in the stonecutter UI
- [MC-303762](#) – (from version 1.21.10) The mouse cursor doesn't change to the proper shape when hovering over the output buttons or the scroll bar in the loom UI
- [MC-303763](#) – (from version 1.21.10) The mouse cursor doesn't change to the hand shape when hovering over the crafting slots in the crafter UI
- [MC-303764](#) – (from version 1.21.10) The mouse cursor doesn't change to the proper shape when hovering over the tabs or the scroll bar in the Creative mode inventory
- [MC-303765](#) – (from version 1.21.10) The mouse cursor doesn't change to the hand shape when hovering over the tabs in the advancements menu
- [MC-303799](#) – (from version 1.21.10) There is no 'Search...' text in the search field in the world selection screen
- [MC-303891](#) – (from version 1.21.10) Untamed zombie horses and skeleton horses cannot have saddles placed on their saddle slot using /item

Credits

Credits

[Minecraft Official Site](#)

[Mojira](#)

[Minecraft Wiki](#)