

Found a bug, need a feature or help with implementation. Contact at:
fx.valley.contact@gmail.com or join Discord <https://discord.gg/3ssjcBcgpu>

Installation & Setup

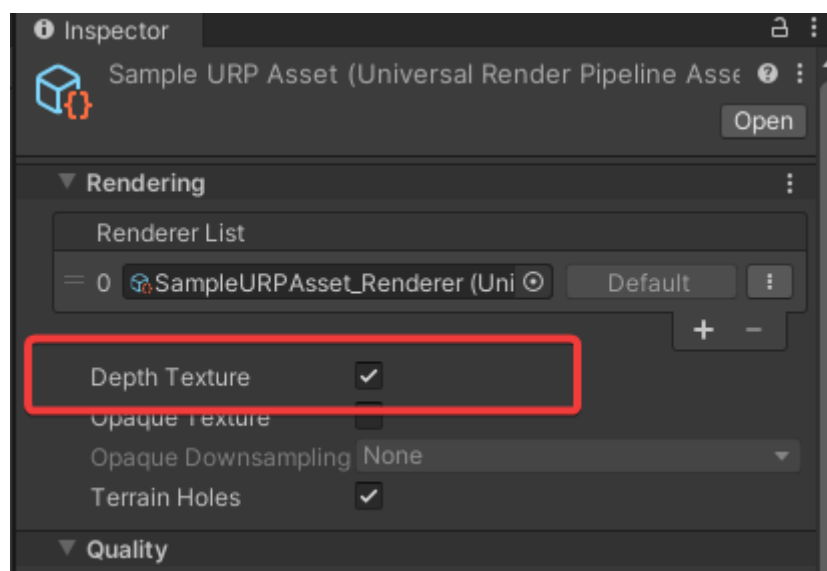
v1.0.11

1. BuiltIn Render Pipeline

- After importing you will need to install the Built In RP package
FXV/VolumeFog/InstallBuiltIn.unpackage that contains all required assets.

2. Universal Render Pipeline

- After importing you will need to install the URP package
FXV/VolumeFog/InstallURP.unpackage that contains all required assets.
- For all features to work properly depth texture needs to be enabled in *Universal Render Pipeline Asset*:



3. Updates

When updating assets to the new version remember to unpack specific Render Pipeline package once again.

If after update you are getting pink shaders - do Right click -> Reimport on all the shader files.

Troubleshooting

If after proper initialization and setup you are experiencing any problems, please contact support by [email](#) or via [discord](#) channel providing as much information as you can:

- Describe your problem, provide screenshots or video if possible
- Unity version (2021.3.31f1+ is supported)
- Platform (PC, Android, iOS, console etc.)
- is it VR project
- Graphics Api (DirectX (version), OpenGL (version) OpenGL ES (version), Vulkan, Metal etc.)
- Render pipeline (BuiltIn, URP, HDRP)
- Render path (Forward, Forward+, Deferred)
- Camera mode - orthographic, perspective
- Where does problem occur - edit mode, play mode, deployed build
- GPU or console/mobile device model
- does the problem also occur when asset is imported to empty Unity project
- errors/warnings from the console if there are any related to the asset
- editor/player log file if the problem is related to deploying and running a build - logs location are described here (Editor.log, Player.log):
<https://docs.unity3d.com/Manual/log-files.html>

Package contents

- Each render pipeline has its own demo scenes:

BuiltIn:

FXV/VolumeFog/Demo/Demo*.unity

URP:

FXV/VolumeFog/URP/Demo/Demo*_URP.unity

Scene named DemoShowcase contains all fog types placed on one scene for quick preview, while scenes Demo1, Demo2, Demo3 show some interesting use case scenarios.

- Fog functionality is handled by two scripts:

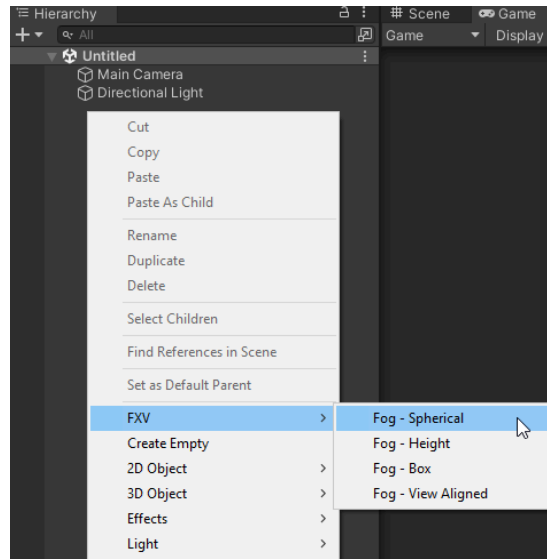
FXV/VolumeFog/Scripts/**VolumeFog**.cs - handles all the functionality and rendering of single fog volume.

FXV/VolumeFog/Scripts/**VolumeFogGroup**.cs - might be added as parent of multiple VolumeFog objects to control their parameters as a group.

- There are shaders for Lit and Unlit rendering for each fog type, and both render pipelines. These shaders are handled automatically by **VolumeFog** script, there is no need to manually setup materials.

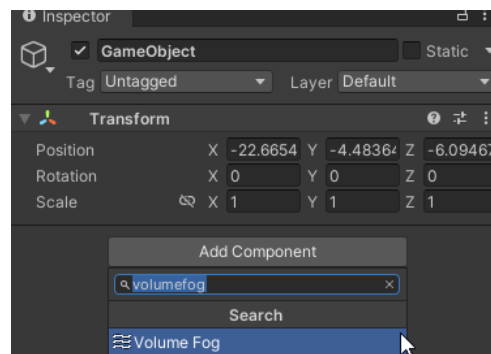
Creating fog volumes

1. The simplest method to create new fog volume is to Right Click in hierarchy panel - select FXV -> Fog **type**



This is limited to a few fog types but fog type can later easily be changed in the **VolumeFog** component inspector.

2. Second method is to add **VolumeFog** to an empty game object. Everything will be initialized by the script.



Resizing fog volumes

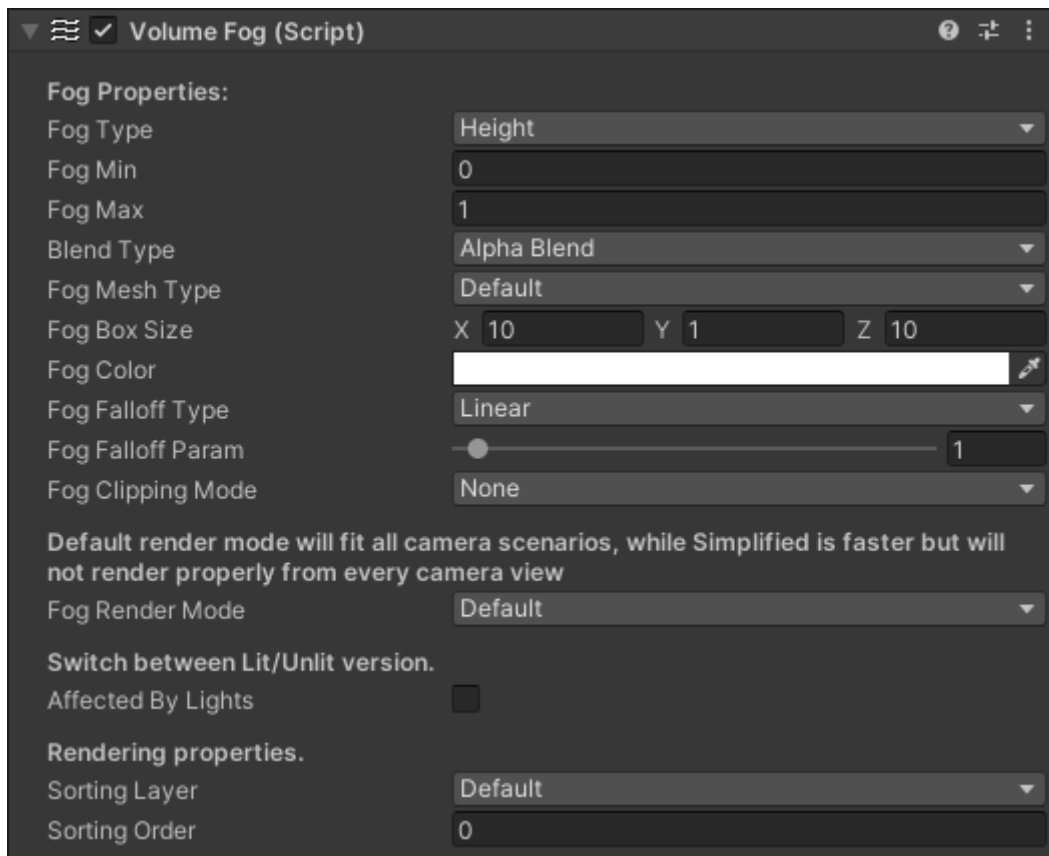
For box type fog volumes (see **Available Fog Types** below) adjust **Fog Box Size** property to resize it on scene.

For sphere fog volumes adjust **Fog Max** parameter to specify its size.

Since 1.0.4 it's also possible to change transform's scale to adjust the volume size. NOTE that when using Custom Mesh scale must be set to 1,1,1 - otherwise behavior is unexpected.

Configuring fog volumes

Every parameter of the fog is set up in **VolumeFog** component:



| | |
|-------------------------|--|
| Fog Type | Most important parameter that defines how fog is rendered. Selecting this will configure a specific shader variant. See Available Fog Types part of this document for all fog types description. |
| Fog Min/Fog Max | Those two values will define where fog starts and ends - or where the fog has 0% density and 100% density based on Fog Type selected. |
| Blend Type | Blending mode Alpha Blend or Additive - Unlit shader only |
| Fog Mesh Type | Default - box or spherical shape is used based on Fog Type selected. Custom - specify custom mesh for fog volume |
| Custom Mesh | Only available if Fog Mesh Type is set to Custom. Set the mesh that should be used by the fog renderer. The limitation here is that transform scale must be set to 1,1,1 |
| Fog Box Size | Only available if Fog Mesh Type is set to Default. Size of the box volume for the fog - this is only available for box shaped fog types. Spherical fog types will define its size based on the Fog Max parameter. |
| Fog Color | Color of the fog - this is the color at 100% density. Use alpha for less dense fog. |
| Fog Falloff Type | Function for fog interpolation between 0% and 100% density - choose the one that best fits your use case. |

| | |
|--------------------------------|---|
| Fog Falloff Param | Parameter for tweaking interpolation smoothness/shape. |
| Fog Clipping Mode | Enable fog clipping. None - no clipping, ClipToSkybox - fog will not be rendered on skybox or background without depth. ClipToBounds - fog will not be rendered on objects outside of the fog bounds volume. |
| Fog Render Mode | Default - will work for most use cases including camera inside fog. This is the most universal render mode. Simplified - does less calculations making shader faster, but will not render properly when the camera is inside fog, or fog volume is viewed from specific angles. This mode might be thought as fake volumetric rendering that will only work for specific camera types - like Top Down camera. |
| Affected By Lights | Switches between Unlit and Lit versions of the shader. Lit shader will work with Point and Directional Lights. |
| Light Scattering Factor | [Lit Only] How much fog scatters light inside fog volume. Use this to tweak light effect strength, size and shape. |
| Light Reflectivity | [Lit Only] How much light is reflected from fog particles inside volume. Use this to tweak light effect strength, size and shape. |
| Light Transmission | [Lit Only] How much light is transmitted by fog particles inside fog volume. Use this to tweak light effect strength, size and shape. |
| Sorting Layer | Fog is rendered as a transparent object. This means that it will be using the Unity transparency sorting algorithm. Change this if you need to render fog volume before or after some other objects. |
| Sorting Order | Rendering order in sorting layer. |

VR Support

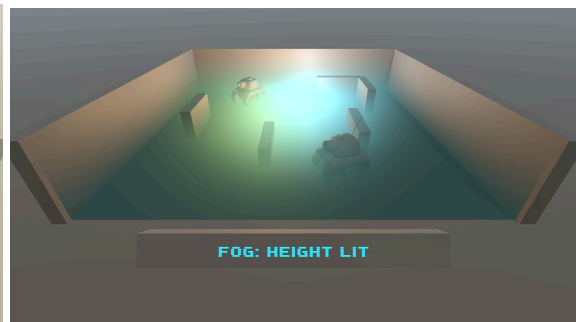
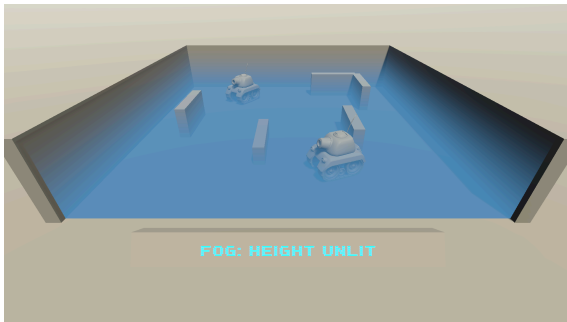
If for some reason you are experiencing issues with fog rendering on a VR device (most likely when the camera is inside fog). Call the static method **VolumeFog.SetEnabledXR(true)** somewhere on startup.

Available Fog Types

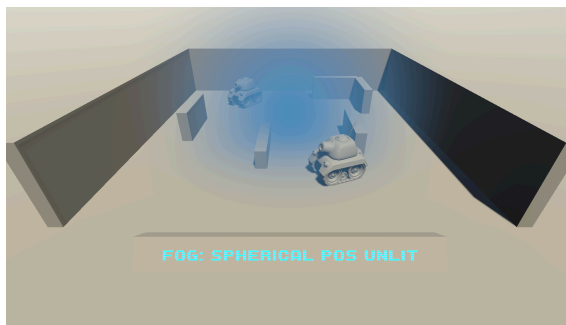
Fog Type is the most important setting and the first that should be selected, as it defines how other parameters will behave.

- Basic Fog Types

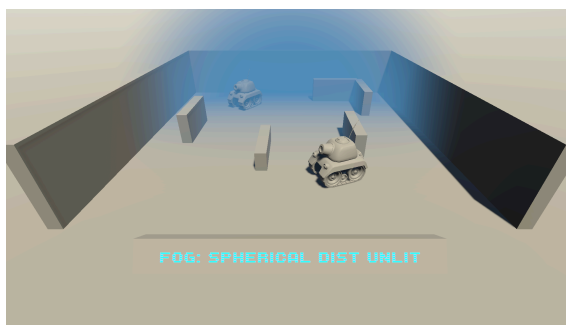
- ☐ **Height** - Fog density is distributed on the object's local Y axis.



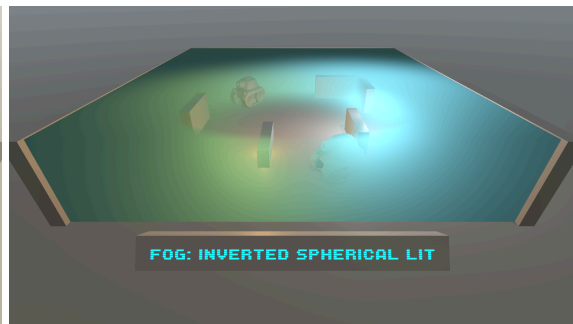
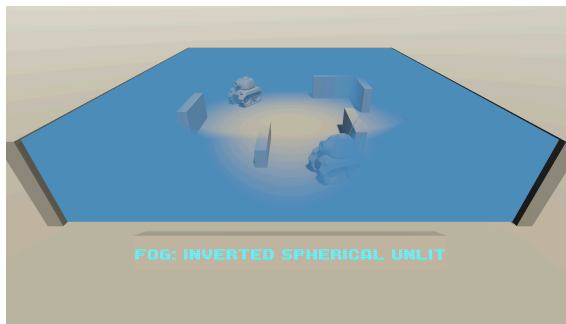
- ☐ **SphericalPos** - Fog density is distributed inside sphere volume, density depends on point position 100% at sphere center - 0% on sphere radius.



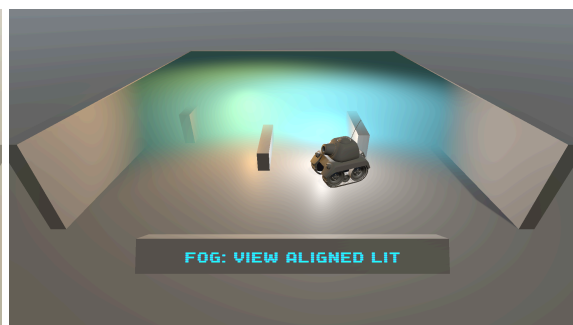
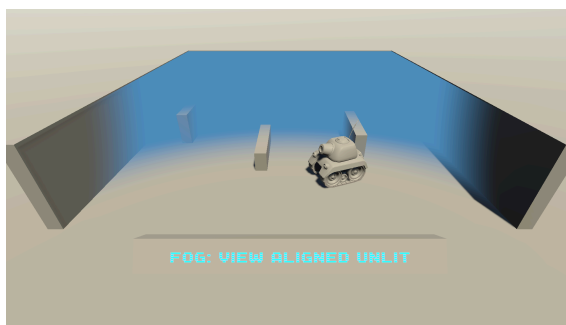
- ☐ **SphericalDist** - Fog density is distributed inside sphere volume, density depends on the distance that view ray travels inside sphere volume.



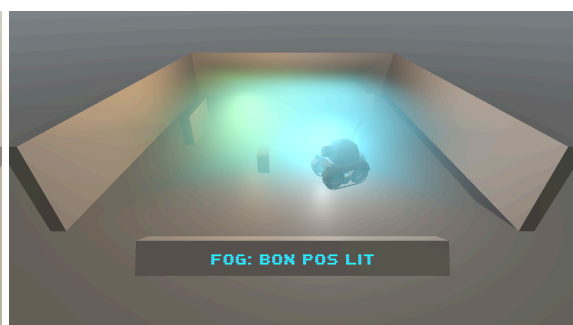
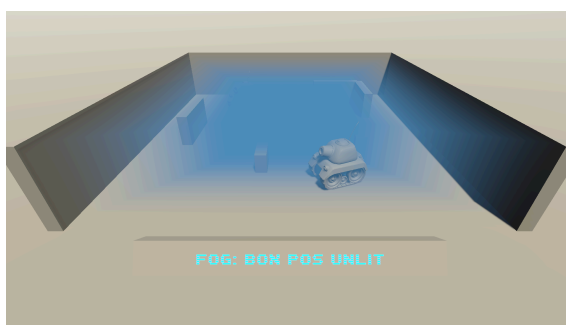
- ☐ **InvertedSpherical** - Fog is rendered inside box volume, but there is a spherical area without fog in the center of the volume. This might be useful for Battle Royale type games.



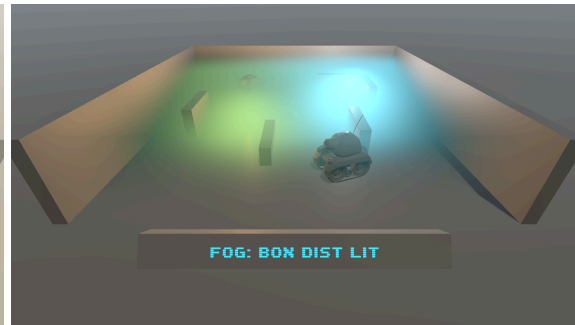
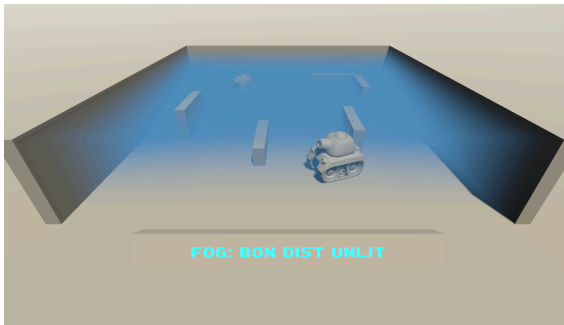
- ☐ **ViewAligned** - fog is rendered relative to camera position and view. This is as the default global fog is rendered but - in this case it's limited to specific box volume.



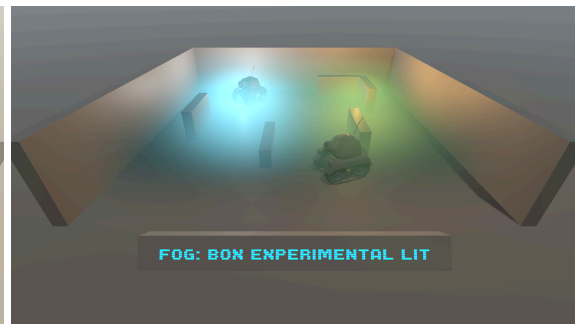
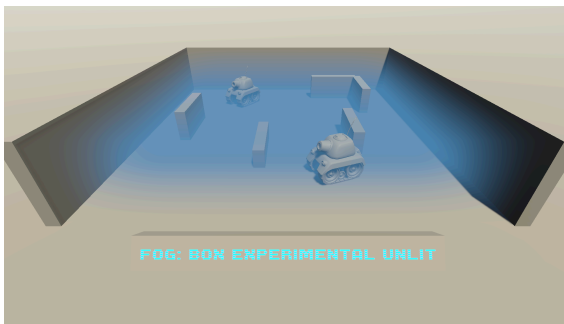
- ☐ **BoxPos** - Fog is rendered inside box volume - density depends on position and distance from the center of the box.



- ☐ **BoxDist** - Fog is rendered inside box volume - density depends on the distance that view ray travels inside box volume.



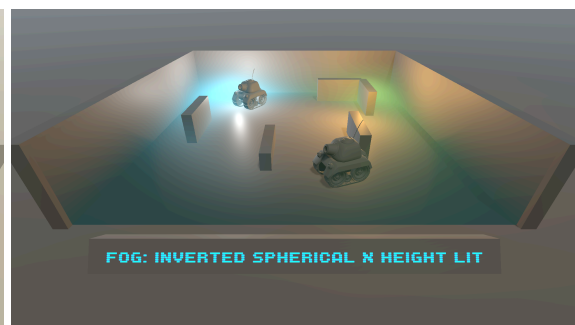
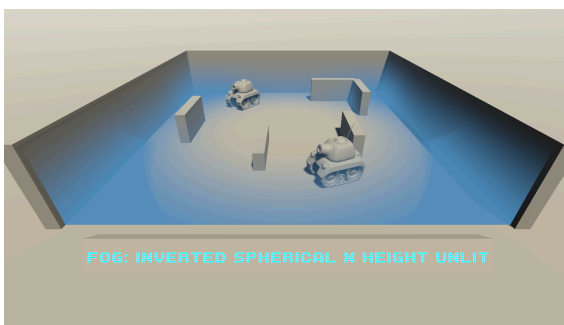
- ☐ **BoxExperimental** - This is a more complex fog type, that renders inside box volume. It works similar to BoxPos - density depends on position and distance from the center of the box, but it's much smoother. There are however some imperfections when looking from different angles. Select the best type for your use case.



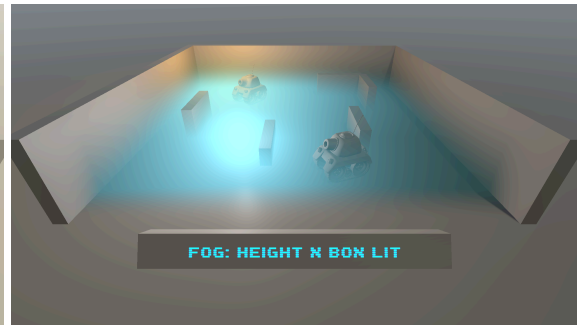
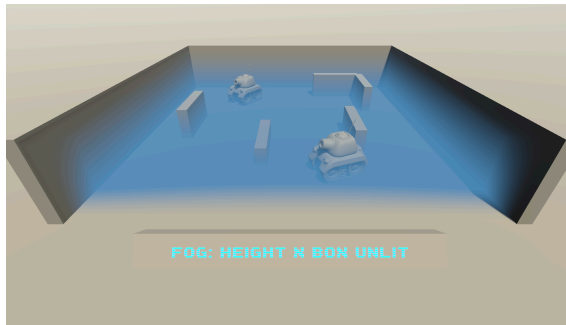
- Mixed Fog Types

Those special fog types are mixing two Basic Fog Types to achieve better visual results like smooth edges for height fog.

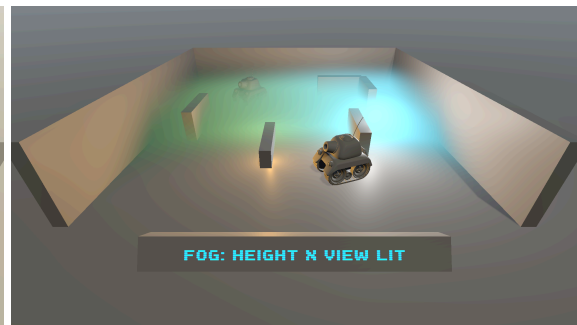
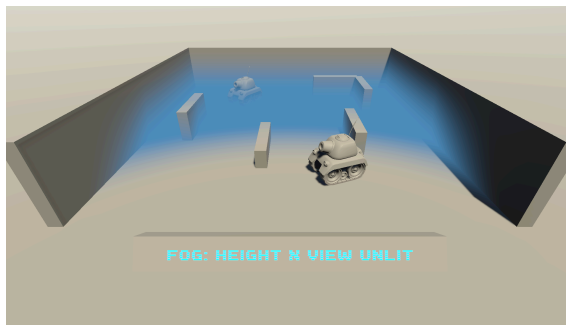
- ☐ **InvertedSphericalXHeight**



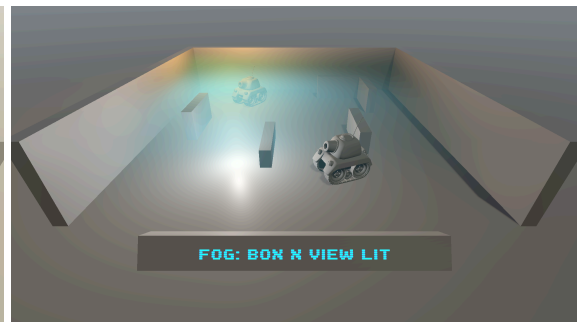
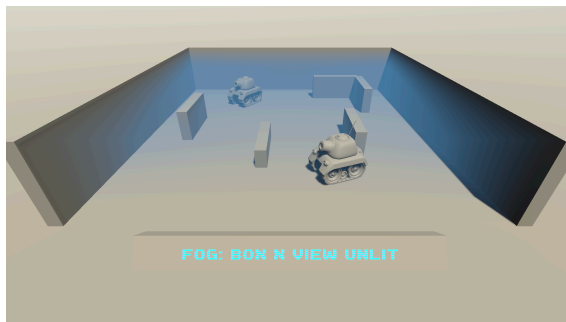
- ☐ **HeightXBox**



☐ HeightXView



☐ BoxXView



Using Fog Groups

The **VolumeFogGroup** component is useful for control of multiple similar fog volumes. It allows changing parameters of multiple fog objects at once. VolumeFogGroup must be added as the parent game object of multiple VolumeFog objects. In a screenshot below VolumeFogGroup is added to *CloudsFogGroup* object, and *FogClouds* objects are VolumeFog objects.



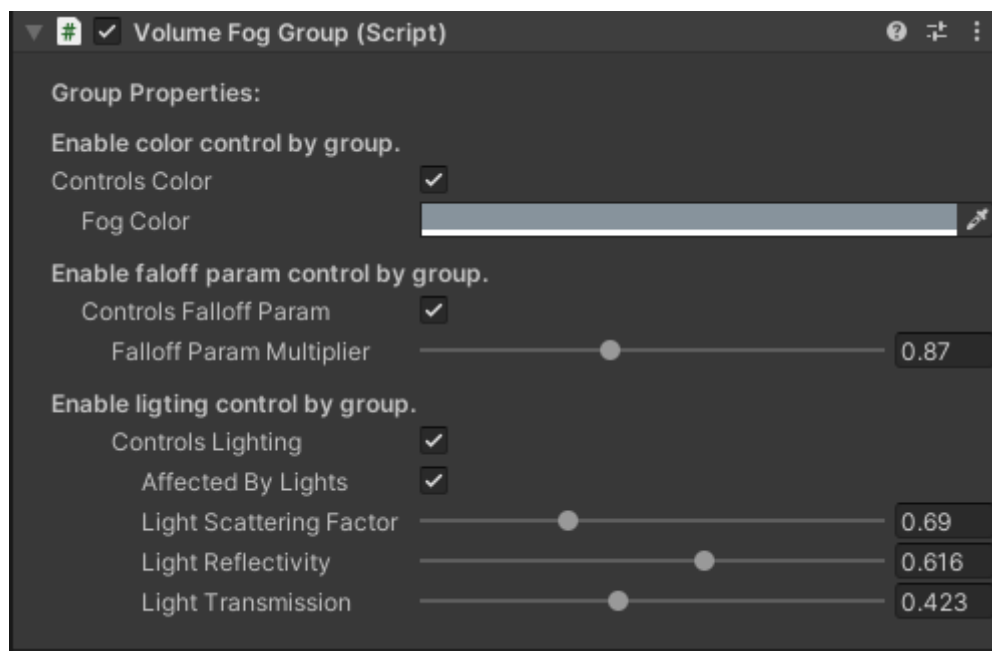
VolumeFogGroup allows changing:

Color - overwrite

Falloff Param - this one is multiplied by original value.

Lighting parameters - overwrite lighting setup of original object.

To enable control of specific parameters by the group - turn on the corresponding checkbox.



Changelog

1.0.11

- fixed asset initialization bug that caused fog not to initialize properly and render with bugs on older Unity versions

1.0.10

- improved compatibility with pre 2021.1.31f1 Unity versions
- improved compatibility across different mobile devices
- Single Pass instancing VR support for BuiltIn render pipeline
- fixed Lit shader bug with Light Layers option disabled in URP

1.0.9

- added VolumeFog.SetEnabledXR(bool) static method to force VR rendering support.

1.0.8

- fixed fog disappearing on VR when camera was inside fog volume
- fixed compatibility issues with OpenGL ES on URP with Lit shader
- Mesh Filter and Mesh Renderer components are now hidden in the inspector when a Fog component is added to avoid changes made directly in those components.
- fixed shader compilation warnings

1.0.7

- fixed rendering problems with orthographic camera
- fog materials for each type are now cached in resources folder and all fog objects are referencing those materials to avoid frequent asset file changes

1.0.6

- fixed rendering left/right eye problems in VR with URP single pass instanced and multipass rendering
- code compatibility improvements for conflicts when importing asset package

1.0.5

- added fog clipping option with possibility to clip fog effect to background/skybox or volume bounds

1.0.4

- added custom mesh shape support
- better handling of sorting layers and order in layer
- box and spherical fog volume can now be adjusted by changing object's transform scale
- new demo scene with custom mesh example

1.0.3

- fixed Lit shader in URP Forward+ path

1.0.2

- fixed bug with calculation of Spherical Distance fog when camera is inside volume