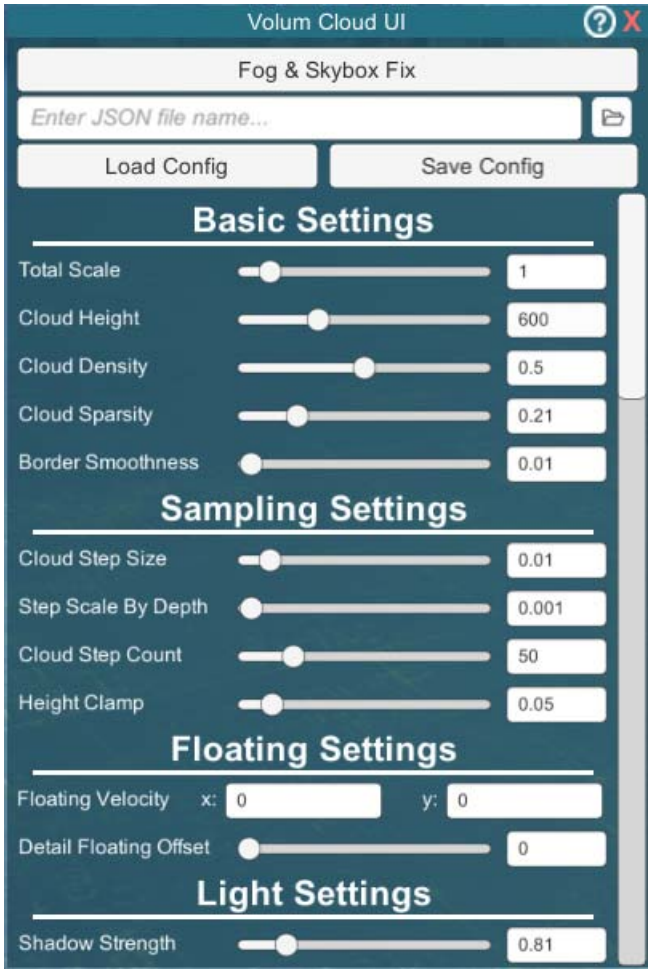
Volume Cloud MOD Instruction Manual

This mod was developed for Level Editor. There could be some issues if you want to use it in old sandboxes. It is recommended to use this mod in Level Editor as much as possible

Chapter 1. UI basic functions

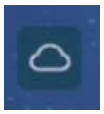
This mod provided an UI menu with several widgets to control each values of the volume cloud. It also supports read and write configs (picture 1-1).



picture 1-1

1.1 Dag and hide

Click the red cross at top right could hide the menu as a thumbnail (picture 1-2), then click the thumbnail to expand the menu. Hold the thumbnail or top of the menu would drag the whole UI.



picture 1-2

1.2 Skybox and fog fix

If volume clouds blocked by Besiege's skybox or fog, click 'Fog & Skybox Fix' then the mod should fix that.

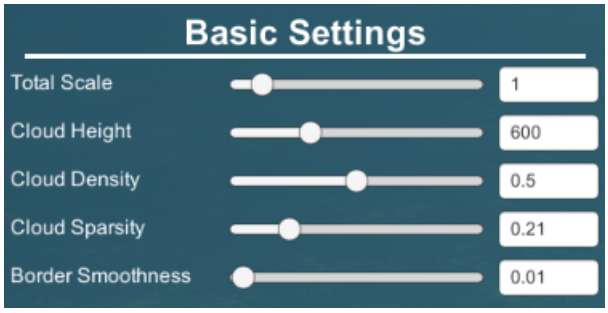
1.3 Config read and write

This mod could save all widgets as a json file. Input the filename into the topper input field of the menu (picture 1-1) then click ‘Load Config’ or ‘Save config’ to read or write config files.

Click the folder icon beside the input field to open the path of config folder (picture 1-3). This mod provided several presets, you can copy them from the folder ‘saves’ under parent directory of this manual into config folder if you want use them.

picture 1-3

Chapter 2. Basic settings



picture 2-1

Total Scale： Volume cloud’s total scale. Because of camera’s far clip plane, volume cloud’s render area will looks like decreasing while this value is increasing.

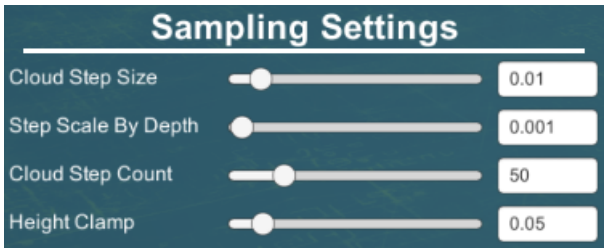
Cloud Height: Height of volume cloud, influenced by total scale.

Cloud Density: Volume cloud’s density.

Cloud Sparsity: Smaller the value, denser the cloud, but recommend to not make this value less than border smoothness.

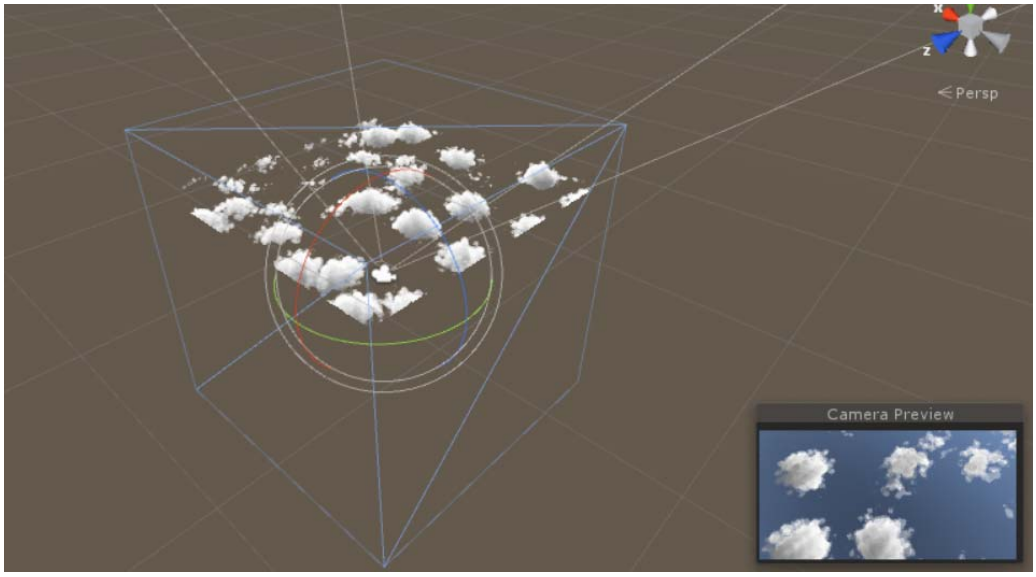
Border Smoothness: Volume cloud borders’ smoothness.

Chapter 3. Sampling settings



picture 3-1

This mod's volume cloud was a kind of volume rendering which sampling Worley noise and Perlin noise textures through raymarching. Define a cube with side length 1, and use camera's position as its center, sampling 2D and 3D noise textures in this cube area (picture 3-2) then scaling to 2000 * 2000 * 2000. Thus, each values in this chapter are based on the cube with side length 1. For example, if 'Cloud Step Size' is 0.01, it means each two steps' distance while raymarching is 0.01 in the cube, but in camera space the distance is 2000 * 0.01 = 20.



picture 3-2

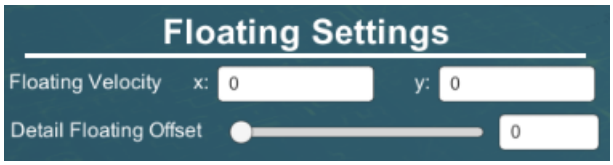
Cloud Step Size: Distance between each two steps while raymarching. While this value increasing, clouds will become blurred and banding, but GPU usage will reduce.

Step Scale By Depth: An optimization of step size. After each steps, step size will add this value to itself which means step size could become larger and larger while raymarching sampling. While this value increasing, GPU usage will reduce, distant clouds will become blurred and baning, but clouds nearby will not influenced much.

Cloud Step Count: Max count of sampling steps. Raymarching will stop if sampling times exceeds this value. While this value increasing, GPU usage and cloud rendering area will increase. Note that sampling beyond far clip plane of camera is useless. As for reference, camera's far clip plane is 4500 in Level Editor. Please consider about total scale and step size first, then adjust this value.

Height Clamp: This value defines an rendering area base on 'Cloud Height' in chapter 2, increase the height range of this value up and down. Clouds out of the area will not rendered. As an example, Cloud Height 600, Height Clamp 0.05, volume cloud will only rendered in height range [600 - 0.05 * 2000, 600 + 0.05 * 2000] = [500, 700].

Chapter 4. Floating settings

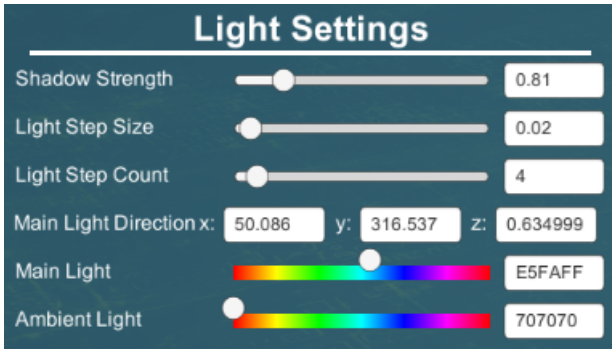


pircture 4-1

Floating Velocity: A vector2 value, defines clouds' floating velocity. Its magnitude defines clouds' floating speed (without direction).

Detail Floating Offset: According to chapter 3, this mod's volume cloud was rendered by sampling several noise textures. This value defines how details provided by noise textures floating relative to basic clouds. While this value is 0, noise details will not floating; while this value is 1, noise details floating with same velocity of basic clouds.

Chapter 5. Floating settings



picture 5-1

Shadow Strength: Controls the strength of cloud's shadow part. With this value increasing, the shadow part's color will closer to shadow color.

Light Step Size: As same as step size mentioned in chapter 3. base on a cube with side length 1. While this value increasing, shadow will become banding, but GPU will sampling a larger area to compute shadow. If you already decreased shadow strength but the cloud is still very dark, consider to increase this value.

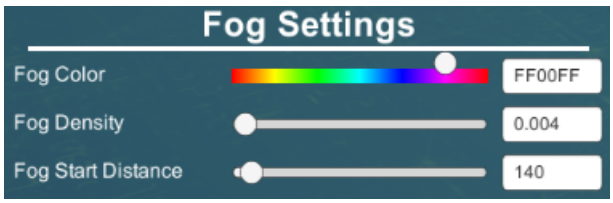
Light Step Count: Light sampling will stop when step times exceeds this value. Recommend to not set this value less than 2.

Main Light Direction: Besiege's main light's euler angles direction.

Main Light: Color of Besiege's main light.

Ambient Light: Color of Besiege's ambient light.

Chapter 6. Fog settings



picture 6-1

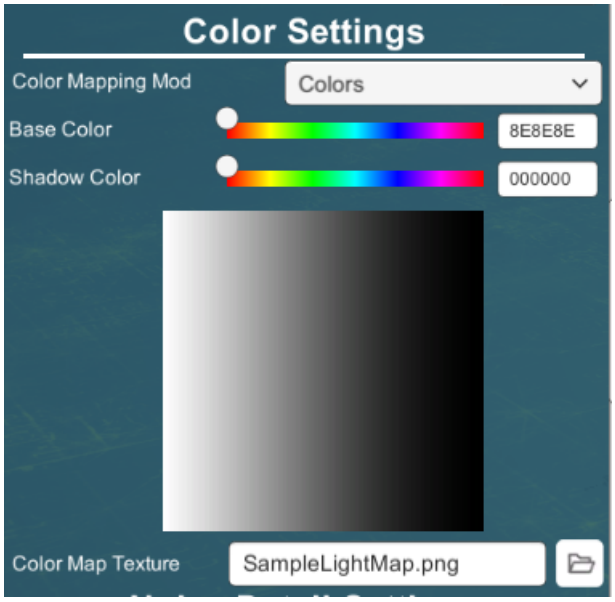
This mod could control Besiege's colorfulFog, but will change its coloringMode to solid. After adjusting the fog, it will not use texture but a single color for sampling.

Fog color: Besiege's Fog's color.

Fog Density: Besiege's Fog's Density.

Fog Start Distance: Besiege's Fog's start distance.

Chapter 7. Color settings



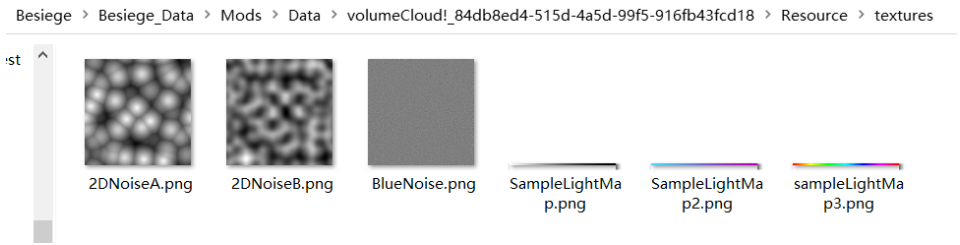
picture 7-1

Color Mapping Mod: Change color mapping mod between colors or textures.

Base Color: Volume cloud's basic color.

Shadow Color: Volume cloud's shadow color.

Color Map Texture: A texture that defines how color changes from basic part to shadow part of volume cloud. Recommend resolution is 256 * 4, from left to right is basic color to shadow color. Using this method allows finer control of the color curve from light to dark, and you can also use more than two colors to make volume cloud more colorful.



picture 7-2

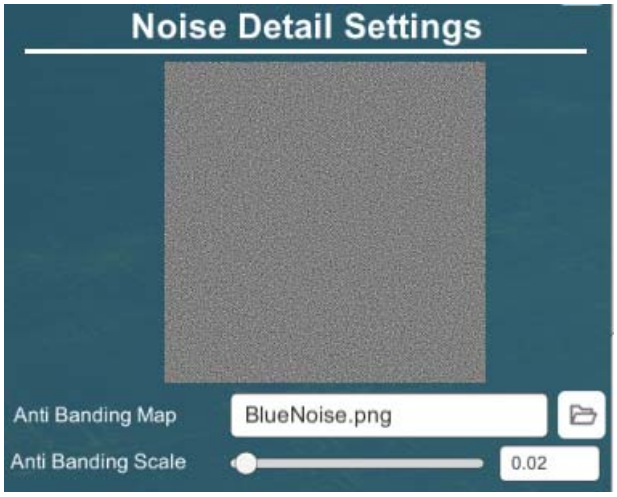
Click the folder beside input field to open texture folder (picture 7-2). To load a custom color map texture, make sure the texture file is in the texture folder and then input filename into input field. After press “Enter” button on your keyboard, you should find the texture was loaded.

this mod prepared several light map presets, you can copy them from the folder ‘textures’ under parent directory of this manual into texture folder if you want use them.

Caution: this mod only support png files.

Chapter 8. Color settings

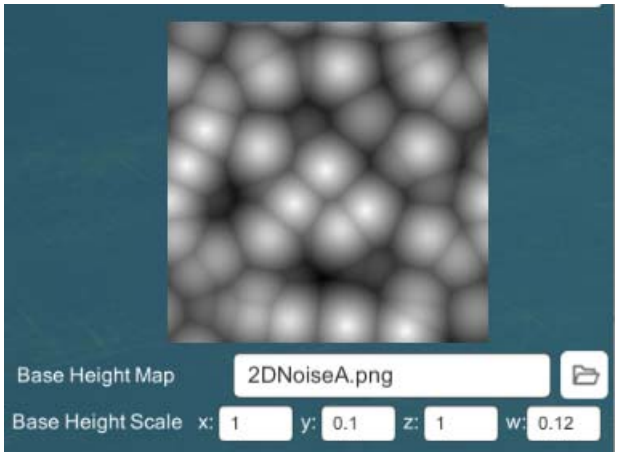
Caution: If you totally don’t understand what this chapter is talking about, DO NOT change any value than mentioned bellow.



picture 8-1

Anti Banding Map: This texture is used to decrease volume cloud’s banding caused by raymarching step size. If you don’t know what is banding or why volume rendering will cause banding, please ignore this widgets.

Anti Banding Scale: Larger this value, less clouds’ banding. But Clouds will become more blurred, and the start distance of sampling will increase, some objects near the camera will not blocked even they are inside clouds.

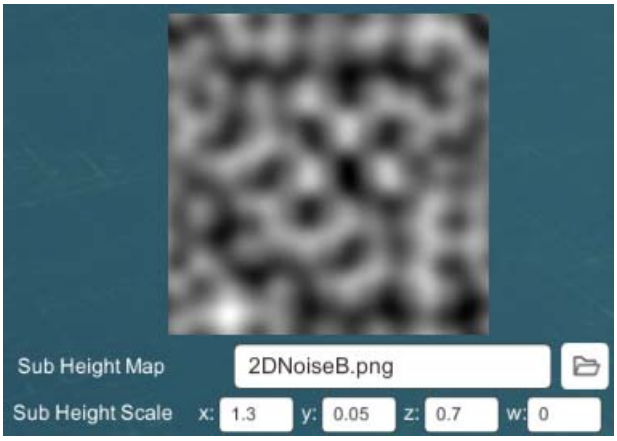


picture 8-2

Base Height Map: A texture only has alpha value that defines basic cloud’s height. Recommend resolution less than 1024 * 1024. Based on ‘Cloud Height’ mentioned in chapter 2, clouds rendering height range will increased up and down by this texture’s value.

Base Height Scale: x, z defines texture scale. Scale here means how may times a texture repeats in a unit length, same as material.SetTextureScale. Base Height Map's value will multiply y and then add w. For example, Cloud Height 600, Total Scale 1, rendering height range at coordinates (x, y) could be expressed as:

600 ± (Base_Height_Map(x,y).a * Base_Height_Scale.y + Base_Height_Scale.w) * 2000

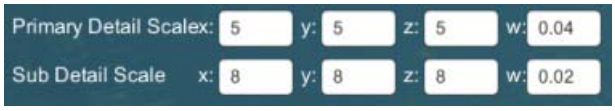


picture 8-3

Sub Height Map: Secondary height map, offset rendering height range by its value to make clouds more various. It only has alpha value, recommend resolution less than 1024 * 1024.

Sub Height Scale: x, z defines texture scale. Scale here means how may times a texture repeats in a unit length, same as material.SetTextureScale. Sub Height Map's value will multiply y and then add w. For example, Cloud Height 600, Total Scale 1, rendering height range at coordinates (x, y) could be expressed as:

600 - (Sub_Height_Map(x,y).a * Sub_Height_Scale.y + Sub_Height_Scale.w) * 2000
± (Base_Height_Map(x,y).a * Base_Height_Scale.y + Base_Height_Scale.w) * 2000



picture 8-4

Primary Detail Scale: x, y, z defines 3D noise texture scale. Scale here means how may times a texture repeats in a unit length, same as material.SetTextureScale but for 3D texture. w defines primary detail's strength. While w increasing, cloud's surface will become uneven.

Sub Detail Scale: x, y, z defines 3D noise texture scale. Scale here means how may times a texture repeats in a unit length, same as material.SetTextureScale but for 3D texture. w defines secondary detail's strength. While w increasing, cloud's surface and border will become uneven. Compared with primary detail, secondary detail is more intensive.

Reference：

Wolfgang Engel.*GPU Pro 7: Advanced Rendering Techniques.*2016