Table of Contents

About kode80 Clouds

kode80 Clouds is a realtime volumetric cloud rendering solution for Unity3D, offering fully dynamic, animated, 3D cloudscapes along with a custom editor for painting clouds directly into your scene. All properties; from cloud density and shape to sun and ambient lighting can be adjusted in realtime. Cloud coverage and type are controlled by a special coverage texture which can also be updated in realtime, opening the door for custom cloud animations and weather systems.

To get the most out of kode80 Clouds, please take a moment to read this manual and explore the included example scene.

If you have any questions, get in touch!
email: support@kode80.com
twitter: @kode80

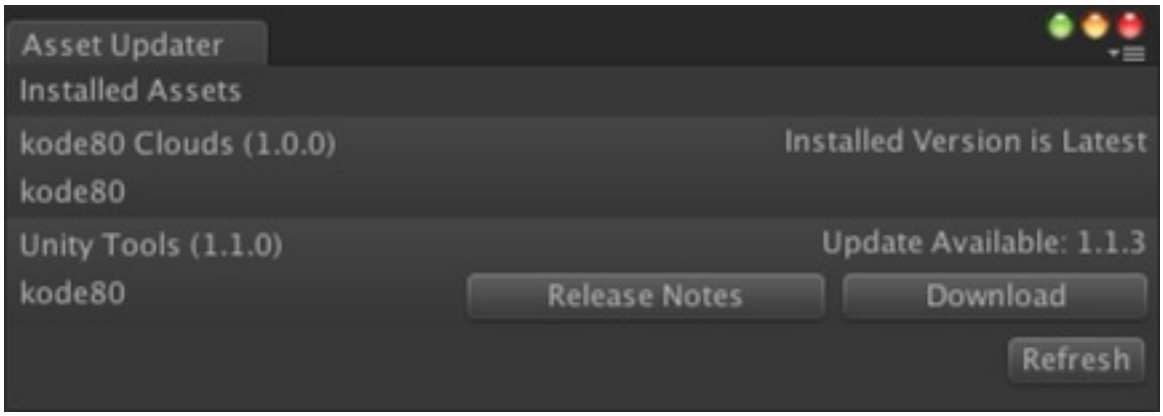
kode80 Clouds License Details

kode80 Clouds for Unity3D is free to use for strictly **non-commercial** purposes. The full non-commercial license is included in the project root.

To use kode80 Clouds for Unity3D for **any commercial** purposes, or to support this & future kode80 development, **commercial licenses** are available for purchase from: http://kode80.com/

Asset Updater

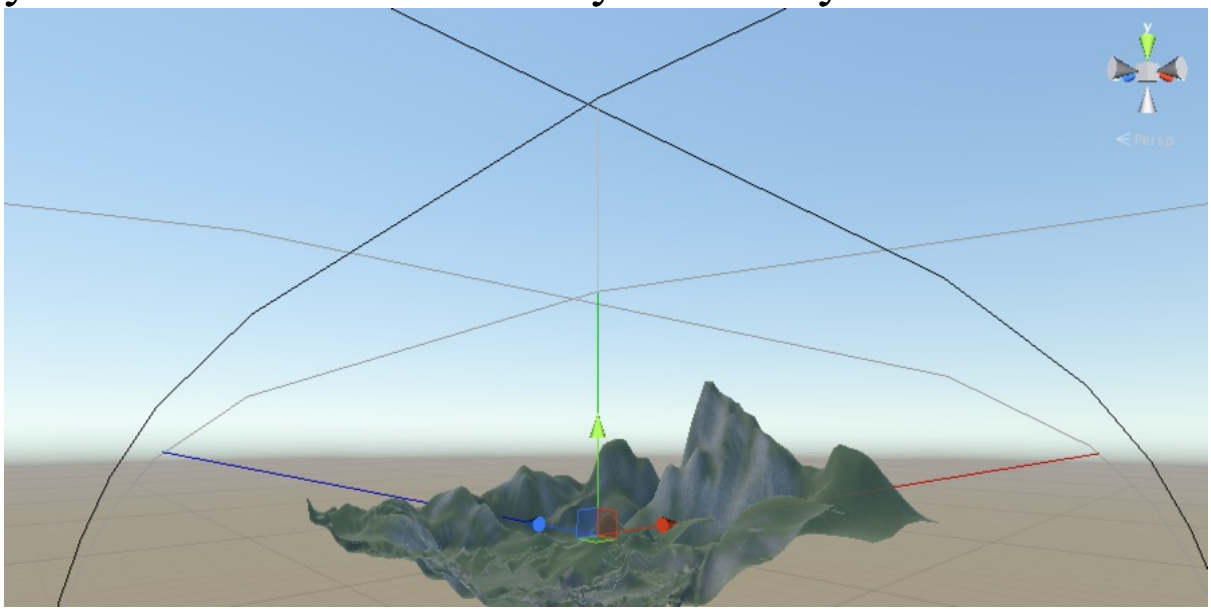
To help you keep up to date with new releases of kode80 Clouds, the package includes the kode80 AssetUpdater. At any time you can go to (Window/kode80/Check for Asset Updates) to display a list of all AssetUpdater supported assets in your project. The AssetUpdater will automatically check if new versions are available and if so allow you to check release notes and download the new version's unitypackage.



Keeping your assets up to date ensures you have the latest features and bug fixes, so check often!

Virtual Sky Dome

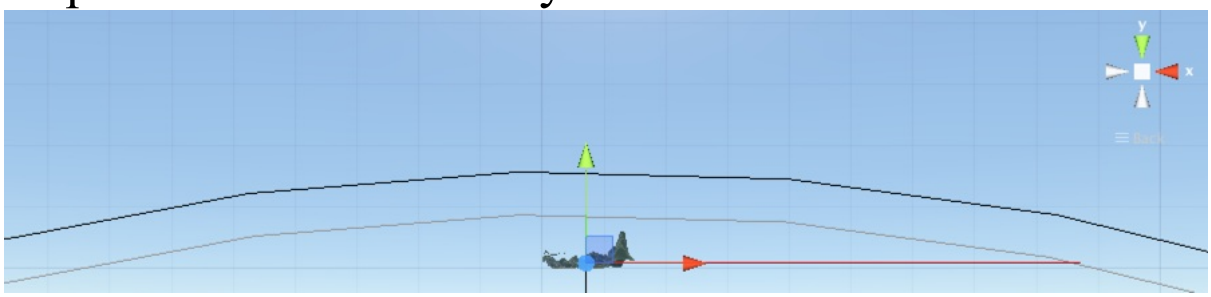
All of kode80 Cloud's rendering takes place in the pixel shader. For each pixel a ray is cast *through* the atmosphere, clouds are modeled mathamatically and light samples are calculated for each cloud sample along the ray. Since there is no actual geometry involved, kode80 Clouds implements a gizmo to help you visualize this virtual sky dome in your scene.



The gizmo draws the X and Z axis stretched out to the Horion Distance setting of the kode80 Clouds component, the Y axis stretched up to the Atmosphere Start Height setting and two wireframe spheres representing where the atmosphere starts and ends respectively. The space between these two spheres is where your clouds exist.

kode80 Clouds Transform

Using the kode80 Clouds Transform it is possible to position the virtual sky dome in your scene in world space. At runtime the camera is free to move anywhere within the virtual sky dome, so it is advisable to center the world space position of the kode80 Clouds GameObject at the lowest point the player can reach in your world, with the Horizon Distance and Atmosphere Start Height set to values that are *at least* larger the respective boundaries of your world.



Clouds Properties

This is the page describing what each of the clouds properties do.

Render

Describes the render properties.

Target Camera

The camera used to render the clouds.

Sub Pixel Size

The number of frames a full resolution render is spread across using temporal reprojection.

Max Iterations

The maximum number of ray steps per pixel

Render Size

Either use camera pixel dimensions (with optional downsample) or a fixed resolution.

Note: for VR projects you should set Render Size to Fixed and Fixed Width/Height manually as Unity's native VR support does not currently report the correct per-eye resolution.

Fixed Width/Height

The dimensions of the cloud render, when Render Size is set to fixed.

Downsample

The amount to downsample the cloud render, when Render Size is set to Camera Dimensions

Coverage

Describes the cloud coverage properties.

Cloud Coverage

The coverage texture. This controls the coverage amount, type and position of clouds in your scene. For more details, read the Coverage Editor section.

Coverage Offset

The x/y offset from center of the cloud coverage. Use

this to *scroll* the clouds.

Horizon Coverage Start/End

The normalized distance from the camera that coverage starts to fade from the coverage map to full coverage.

Lighting

Describes the lighting properties.

Cloud Base/Top Color

The ambient light color for the bottom and top of clouds near the camera.

Sun Light

The Unity3D directional light used as the sun. The light's direction and color are used to render direct cloud lighting.

Sun/Ambient Scalar

Multiplier for sun/ambient light.

Sun Ray Length

Normalized length of sun rays when calculating cloud lighting.

Cone Radius

Normalized radius of sun ray cone when calculating cloud lighting.

Density

Cloud particle density when calculating cloud lighting. Higher values will give clouds a 'thicker' appearance.

Forward/Backward Scattering G

The Henyey-Greenstein G values for forward/backward scattering when calculating cloud lighting.

Dark Outline Scalar

Alpha value for cloud dark outline, when camera is facing along sun direction.

Animation

Describes clouds animation properties.

Animation Scale

Global time scalar for all animation properties.

Coverage Offset Per Frame X/Y

The amount to scroll the coverage map each frame.

Base Offset Per Frame X/Y/Z

The amount to scroll the low frequency noise 3D texture each frame.

Detail Offset Per Frame X/Y/Z

The amount to scroll the high frequency noise 3D texture each frame.

Modeling (Base)

Descrobes the modeling base properties.

Base Scale

The amount the low frequency noise 3D texture is scaled. Setting to 1 scales to fit the height of the atmosphere.

Cloud Gradient 1/2/3

The cloud gradients determine how the different frequencies of 3D noise are blended together based on height in atmosphere.

Sample Scalar

Scales the cloud particle alpha values *before* passing to cloud lighting calculations.

Sample Threshold

Determines the cutoff value for cloud particle values from the 3D noise textures when modeling the clouds.

Cloud Bottom Fade

The normalized height in the atmosphere that cloud bottom fading ends. Setting to higher values results in softer cloud bottoms.

Modeling (Detail)

Describes the detail cloud modeling properties.

Detail Scale

The amount the high frequency noise 3D texture is scaled. Setting to 8 matches the scale of the low

frequency noise.

Erosion Edge Size

A normalized value that determins how far in from the edges of the base cloud shape high frequency noise should erode.

Cloud Distorion

A normalized value that controls the level of curl distortion applied to the high frequency noise during cloud modeling.

Cloud Distortion Scale

Controls the scale of the curl noise distortion texture. Setting to 0.5 is a good default value.

Optimization

Describes the optimization properties.

LOD Distance

The normalized distance from the camera that raycasting switches to low level of detail (ignores high frequency noise modeling step).

Horizon Level

The normalized height of the horizon. Any clouds below the horizon level will not be rendered, this can be useful if the horizon is largely obscured by your scene's geometry since clouds will always be rendered *regardless of occlusion*. As an example, if in your scene the only time the clouds are viewable is by looking directly up through a skylight window, then you can set the Horizon Level property to a high value.

Horizon Fade

The normalized distance from the horizon to the camera over which clouds are faded to completely transparent.

Horizon Fade Start Alpha

The starting alpha value when blending clouds from the horizon. Useful for blending the horizon with another skybox.

Atmosphere

Describes the properties that control the atmosphere and by extension, the size of the planet.

Horizon Distance

The distance in world units from the camera to the horizon.

Atmosphere Start/End Height

The height in world units that the atmosphere starts and ends. i.e. End - Start = atmosphere *thickness*.

Camera Position Scalar

A multiplier that is applied to the Unity3D camera's world position when rendering the clouds.

Cloud Shadows

Using the kode80CloudShadows component, it is possible to add an extra level of realism by having the clouds cast shadows *onto* your scene geometry. To enable cloud shadows:

1. Add the kode80CloudShadows component to your directional light
2. Assign your kode80Clouds instance to the clouds property of the kode80CloudShadows component

Note: as with all shadows in Unity, the distance at which cloud shadows are rendered is controled by the Shadow Distance property in your project's Quality settings.

Coverage Editor

This is page describes how to use the clouds editor.

Editor Modes

The Coverage Editor has 3 main modes; Camera, Coverage and Type. When in Camera mode you can navigate your scene using the mouse and WASD controls. When in Coverage or Type mode, you can paint directly into your scene using the mouse, holding down shift will negate the opacity value of the brush (allowing you to erase Coverage/Type data).

Editor Properties

Enable continuous update mode to view a continuous animated preview in the Coverage Editor.

Editor Camera

When opening the Coverage Editor, a special camera is automatically created by duplicating the clouds target camera, including image effects. This is useful for getting an accurate preview of the game's render directly in the Coverage Editor. Any image effect on

this camera can be manually disabled from the Editor Camera foldout in the Coverage Editor.

Brush Properties

When painting clouds in the Coverage Editor you can control the brush size, opacity and texture from the Brush Properties foldout. When the Blend Values toggle is checked, coverage painting will accumulate, when it is unchecked paint values will be constrained to the Brush Opacity value.

Cubemap Export

Clicking the Cubemap Export button in the Coverage Editor will render a new cubemap from the current camera and save it to your Assets folder.

Table of Contents