Volumetric Clouds 3.0

Suggestions & comments for this document are open and welcome

Table of Contents

// Instructions

1. Drag the “RaymarchedCloud” script inside the folder to your camera.
2. Drag and drop on of the material onto the script’s material slot.

// Material properties

- *Perlin Normal Map*

  Texture used once rendering to determine where and how clouds should be rendered.

Colors

- *Base Color*

  Main color.
- *Shading Color*

  Shading color.
- *Sky Light Attenuation*

  Tint and intensity of the natural occlusion coming from the sky.
- *Advanced Lighting*
  - *Indirect Lighting*

    Global Illumination contribution.
- *Normalmap*

  Shade based on surface direction.
  - *Normals Intensity*
  - *Normalized*

    How far light “scatters through clouds”.
- *Distance Blend*

  Blend clouds out based on draw distance.
- *Post Blend*

  When disabled, blends distant clouds more accurately when multiple clouds overlap on the same pixel but might introduce noise.
- *Screen space shadows*
  - *Shadow Color*
  - *Blend out of shadowmap*

    Filter generated shadows out of unity’s shadowmap.
  - *Volumetric*

    Change from projecting shadows onto the scene to actually sampling them in the same way that clouds are rendered, more accurate but heavier to render.

- *Render Shadows Only*
- *High quality point light*

  Render point lights for each step sample instead of once per pixel, point light rendering is only supported on unity 5.4+.

Shape

- *Coverage*

  Amount and size of clouds
- *Coverage Map*

  Same as above but using a texture mapped in world space, the red channel controls the density (black -> red = low intensity -> high intensity), tiling and offset is in world space so 100 tilling means that the texture's size is equal to 100 unity unit.
- *AlphaCut*

  Discard pixels where opacity is less than that, doesn't help performance in any way.
- Density

  Density of the clouds, how quickly they become opaque.

Animation

- *Speed*

  Wind speed over the first layer
- *Speed Second Layer*

  Wind speed for the second layer, should be different from the first one to animate properly.
- Wind Direction

Dimensions

- *Cloud Transform*

  X is cloud height.
  Y is cloud size.
  Z is offset on the x axis.
  W is offset on the y axis.
- *Tiling*

  Horizontal size of the clouds.
- *Cloud base flatness*
- *Spherical*
  - *Sphere position*

    The sphere position in 3D space, 'W' does nothing.
  - *Spherical Mapping*

    Map clouds around the sphere in a manner similar to unity's sphere model.

    - *Sphere Stretch Horizon*

      With spherical mapping, blend poles out.

      Without spherical mapping, stretches the texture towards the horizon.

- *Plane Alignment*

  Swap clouds from horizontal (Y axis) to the two other vertical alignment.

- *Orthographic Perspective*

  A factor which fakes perspective if the camera used to render clouds is orthographic.

Raymarcher *(cfr:Raymarching informations)*

- *Draw distance*

  Stop rendering once reached past that distance.

- *STEPS Configuration*

    - *Steps Max*

      Maximum number of steps allowed, keep this at the lowest amount possible while still retaining the visual quality that you want.

    - *Step Size*

      Distance between each steps, lose overall rendering precision and gain performance/draw distance by increasing it.

    - *Step Skip*

      Increases the distance between each steps when the sample didn't land on a cloud.

- *Lod Base*

  Changes the mipmaps used, similar to reducing texture resolution.

- *Lod Offset*

  Same as above but based on the distance to the camera.

Debug

- *Render Queue*

  When should the clouds be rendered, 2501 is the default and is right after opaque objects, 3000 is for most transparent objects, modify this value if clouds are rendered behind or on top of something it shouldn't.

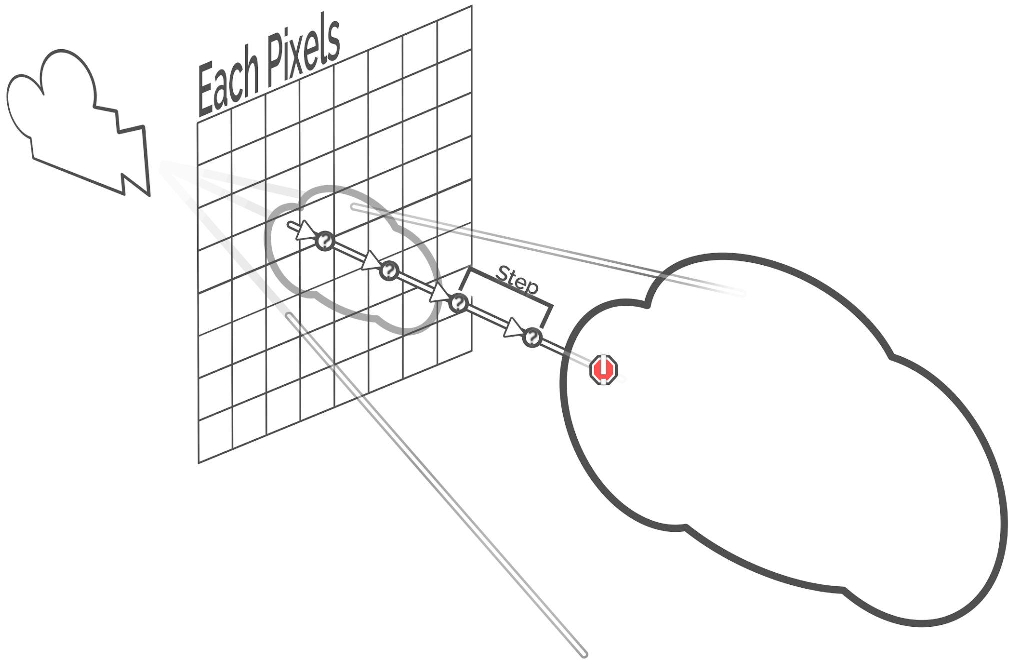
- *Show Step Count*

  Render the amount of steps for each pixels on screen as a green to red gradient, the redder the pixel is, the closer it is to your 'Steps Max', and so the more demanding that pixel was overall.

You should use this toggle when trying to optimize this material. Don't forget that the amount of steps used is view dependant, it is pretty cheap to render from a top-down perspective but far less cheap when rendering from inside clouds.

// *Raymarching*

A raymarcher works by launching rays from the camera's point of view towards infinity and test them against volumes, in this case, clouds. To do so, at each x unit along those rays we use a function to check whether we are inside or outside of a volume at that position, if we are inside we render clouds.
A step is the distance we travel between each check along the ray.



If the steps are too large the ray will skip past clouds and will appear like noise or cuts.
If the amount of steps is too low farther clouds won't be rendered.

// Tips

The two Speed variables shouldn't have the same value, clouds won't be able to morph through time if they are.

Draw distance too short ? Try increasing the variable Draw Distance to the max that you would like to render (distance is in unity unit, 1 to 1) then increase Steps Max, and if that didn't do anything, change Lod Offset.

// Optimisations

Everything under the raymarcher header changes performances a ton, play with those values and you should see significant changes.

Direct3D 11(Dx11) and other more recent APIs should render them a LOT faster, consider using them in your project.

// Common issues

Clouds don't show up :

> Check that the variable "Material Used" on the "Raymarched Clouds" script is filled with your material.
> You might have to enable depth map generation on your camera.

Clouds rendering blinks on and off :

> Increase "Plane Offset" on the C# script until it doesn't anymore.

Clouds render weirdly :

> Skim through the chapter on the raymarcher.

"Blend out of shadowmap" doesn't work/doesn't work as expected :

> You’ll have to decrease your render queue to 2500 or lower, unity removes the shadow mask from materials within that range.

The texture used with the shader needs four texture channel(RGB = Normals, A = Heightmap), do not mark your texture as a normal map and check if the compression used doesn’t discard one of those four channels.

If none of this helped then drop by the official thread on the Unity forum.