# Volumetric Clouds 2.x

This document is freely modifiable, you can add stuff you think would look better or would be interesting.
Once an edit has been made, the owner of this document (Eideren) will have to approve it before being applied globally.

## // Instructions

1. Drag the "RaymarchedCloud" script inside the folder to your camera.
2. Drag and drop the cloud material onto the script's material slot.

## // Material variables

- **Perlin Normal Map**
  - Texture used once rendering to determine where and how clouds should be rendered.
- **Colors**
  - **Base Color**
    - rgb = Main color, a = shading intensity towards white
  - **Shading Color**
    - Rgb = Shaded areas color,  a = shading intensity towards black
  - **Use Normalmap**
    - Shade based on surface direction.
    - **Indirect Lighting**
      - Amount of GI received
    - **Normalized**
      - How far light scatters before stopping
  - **Depth Intensity**
    - Top down shading
  - **Distance Blend**
    - Alpha decreases based on distance from the camera to blend nicely with the skybox.
  - **Screen space shadows**
    - **Shadow Color**
      - Shadow color.
    - **Draw Distance**
      - Maximum shadow rendering distance.
- **Shape**

    - **Density**
        - How much cloud there is
    - **Alpha**
        - How thick are those clouds
    - **AlphaCut**
        - Discard pixels where opacity is less than that
- **Animation**
    - **Speed**
        - Movement speed of the first layer
    - **Speed Second Layer**
        - Movement speed of the second layer, should be different from 1st to create cloud evolution
- **Dimensions**
    - **Cloud Transform**
        - x is cloud height
        - y is cloud size on the y axis
        - z is offset on the x axis
        - w is offset on the y axis
    - **Tiling**
        - Cloud/meters
- **Raymarcher**
    - **Draw distance**
        - Stop rendering once reached past that distance
    - **STEPS**
        - A raymarcher works by launching a ray from the camera's point of view and figuring out if we hit something. To do so, we check each x unit along the ray a function which tells us if we hit something,
          so a step is just that, a check along the ray.
        - **Steps Max**
            - Maximum number of steps to do, will stop before getting to that amount in most cases.
        - **Step Size**
            - How far should we go before asking if we hit something. Base distance.
        - **Step Skip**
            - How far should we go based on the current distance travelled, is useful to save performance on distant clouds.
        - **Step near surface**

        - All of the above multiplied by that once we are near a surface to accurately represent surface shapes.
    - **Lod Base**
        - Changes the mipmaps used, effectively reducing texture resolution.
    - **Lod Offset**
        - Same as above but based on distance to the camera.
    - **Opacity Gain**
        - The overall gain in opacity each time we hit a cloud.
    - **Skip Pixel**
        - Skips pixels and takes the derivatives instead, helps performances
- **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.

## // *Tips*

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

## // *Optimisations*

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

*Direct3D 11(Dx11) should render them a LOT faster, consider using it in your project if you target the PC platform.*

## // *Not behaving correctly ?*

*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.*

*Check the variable "Cloud Material" on the sliced Volume script, it has to be the same as the one used on the same game object.*

*If the clouds simply don't show up, you might have to create a depthmap.*

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