CloudsToy v2.2.5

CloudsToy is a cloud system that creates a lot of shuriken particles, each of one creates a cloud. It's focused in being very simple to use and works even on mobile (it depends on the number and complexity of the clouds). Almost all scene examples are made using CloudsToy.

It includes several presets so you can see the range and quality of the clouds that can be created with it.

CloudsToy Simple is a new basic set of clouds using a single Shuriken Particle System has been created so you can see how it works. If you want to play and tweak it, I recommend you duplicate the CloudsToy Simple prefab so the original one isn't modified in any way.

The scene '8.-SimpleClouds' contains an example of this simple clouds.

New: CloudsToy Simple Animated is a new basic set of clouds using a shader created with Shadergraph (Builtin & URP) and used in a plane to simulate clouds. If you want to play and tweak it, I recommend to duplicate the CloudsAnimate material and assign it to new plane so you can play with it without modifying the original one.

The scene '9.-SimpleCloudsAnimated' contains an example of this simple clouds.

New: URP Compatibility: Import CloudsToy as usual in your URP project and double click in the file located at folder **/CloudsToy/URP Addon/CloudsToy URP Addon.unitypackage** to update the materials used in the demo scenes to URP.

CloudsToy needs to **include** the **Shadergraph** package to be able to use the Cloudstoy Simple Animated.

How to test CloudsToy:

1.- Create a layer named **CloudsToy** in any layer number you want.

2.- Open the demo scene located in: Assets/CloudsToy/ Scenes/CloudsToy

NOTE: There are far more example scenes in this assets.

How to use CloudsToy in a new scene:

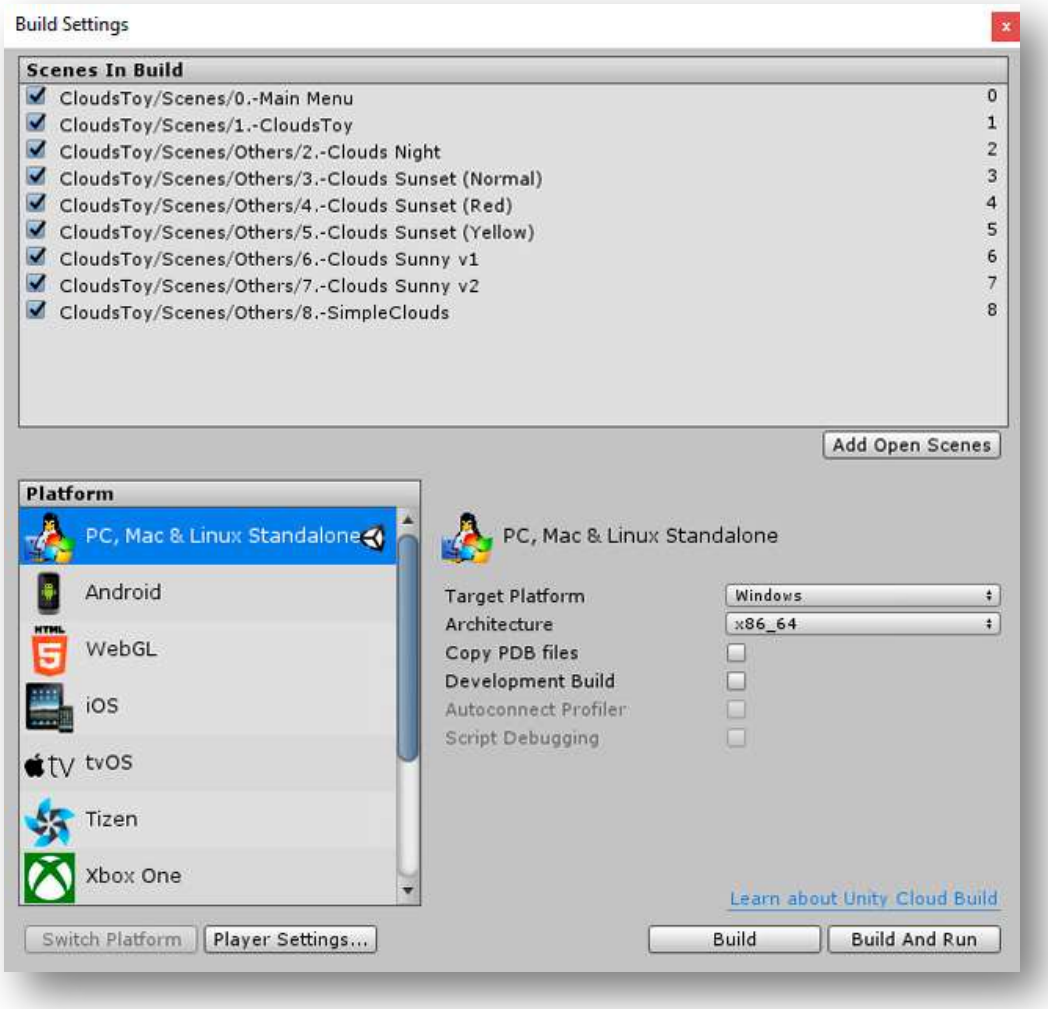
1.- Create a layer named **CloudsToy** in any layer number you want.

2.- Just drag the **CloudsToy Mngr** prefab into your scene.

3.- Hit Play and enjoy!

How to build or play CloudsToy Demo using the included scenes:

1.- Go to File -> BuildSettings and setup the scenes in the following order:



2.- Hit Build!

Notes about CloudsToy

The first variable "**Maximum Clouds**" will create internally all the clouds that CloudsToy is going to manage, they will be visible or invisible depending on the "Clouds Num" variable. Ideally, once you fine-tune your clouds to fit your needs, the Maximum Clouds and the Clouds Num variable should have the same value (so all the clouds you see are all the clouds the system is managing).

IMPORTANT: *Try NOT change this variable in runtime, it can brake how CloudsToy works.*

There is a new advanced variable that can be used to get so more FPS. It is in Particles Advanced Settings -> Position Checker Timer. If set to zero It will check the clouds position in every frame, if set any other value it will use that value as a timer interval to check it. The position checker is used to 'see' when clouds reaches a side of the limits in the CloudsToy (yellow rectangle) and move them to the opposite side starting the movement cycle.

The **Repaint** button is used because not all actions in the editor are going to change all the clouds in real-time mode (it could be too slow depending on the number of clouds that CloudsToy is managing), so if you want to be sure how actually the clouds looks like, hit the Repaint button just in case.

Soft Clouds is used to make sunrise type of clouds. They are very large, soft and almost transparent clouds.

Be aware that depending on the number of clouds, the type of render, the detail of the clouds (number of particles that the clouds have) and the size width, height and depth, the system's FPS may drop down dramatically. All these variables can be changed in the editor in real-time mode so be careful with what values you use.

All the clouds can have their own shadow (configurable by the user). To work properly, the clouds are stored in the **30th layer** by default. You can create a new layer named **CloudsToy** in any layer number you want. CloudsToy will detect the new layer and use it.

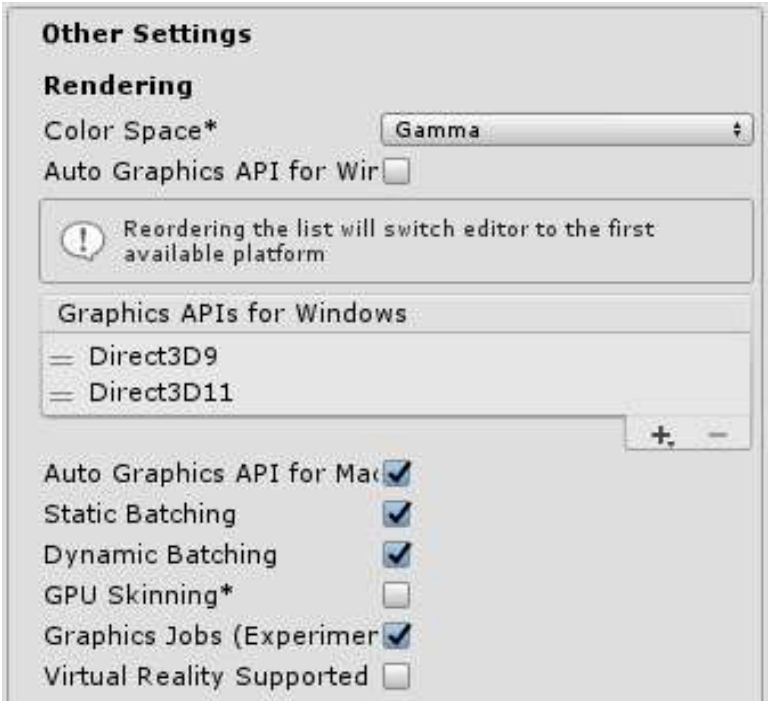
CloudsToy Window (welcome window). For quick review just check this window that can be opened click on CloudsToyMngr icon in inspector or going to Unity's Menu -> Window -> CloudsToy Wellcome

IMPORTANT ISSUES
I see white artifacts in the screen.

You can try two things:

1. *Check you're not using Anti Aliasing:*
   - *Go to Menu-->Edit-->ProjectSettings-->Quality*
   - *Set "Anti Aliasing" property to "Disabled "*

2. *Try using DX9 instead of DX11 (if possible It depends on the Unity version you're using).*
   - *Go File --> BuildSettings --> PlayerSettings and setup it as follows:*



Sometimes some part of the cloud is occluded by another part when moving/rotating the camera.

It's a normal issue. Each cloud is a particle system so its shape is made of particles emitted to create that cloud. So, when you move around a cloud some of the particles are rendered in front of others depending on camera position. When moving around that weird effect appears so the cloud change its aspect.

This will happen in the scene view if you rotate the view when adjusting your 3d objects or terrain in the scene.

Check if it happens in your game view, normally if you are using Cloudstoy in some kind of fly simulator, this visual glitch will occur, if you're using it in a First/Third person controller, it will not (you can't rotate the camera around a cloud in that case as you do in the editor or flying around in a plane).

***Solution**: Just made the clouds color with a lot of transparency. If the Cloud particles are very transparent, the effect just disappear. Try changing the "Cloud Color" & "Main Color" variables in the Inspector (just try to increasing the alpha value). If you need more particles to compensate, adjust "Cloud Detail" variable to High.*

Procedural Textures.

The procedural texture can be used in the clouds in the editor in real-time if you choose Cloud Type == PT1 (Procedural Texture 1) and hit play. The goal is to try to generate new textures using some noise generator to make it possible and save the texture in the same project when finish.

It can use two different noise generators to create the texture: simplenoise and perlin.

Using the same values when you run the application will not generate the exactly same textures twice. You can test it by clicking play button in Unity twice and you'll see the texture is different (sometimes completely different) using exactly the same texture parameters, that's the normal behavior not any estrange error.

Notice that Clouds Textures works drawing all the corners of the textures in black (see at the textures Unity provide in his Standard Assets to create smoke, for example), you can do the same using the Halo parameters (Halo and Radius) and seeing the result in the Procedural texture editor window and in the clouds themselves in the scene window (if you have chosen TypeCloud == PT1).

CloudsToy expect you to use a 64x64 or 128x128 texture size only, using greater values will cause a slow execution of the cloud system (depending on your computer) and different behaviors on Halo functions. Use mixed values like 64x128, will cause weird textures as result, so use that kind of texture sizes at your own risk!

Hit 'Save Texture' once you have fine tune your cloud texture. It will be saved at /Volumetric Clouds/Textures/Procedural. Do NOT erase that folder or CloudsToy will fail for sure. Once saved, use the new texture (Add and Blended) dragging them to the textures field in the clouds parameter. The blended textures are the same that Add ones but with an alpha channel (transparency) active. These are used depending on the type of render you want to use (Bright --> Add/Realistic --> Blended). Important: This feature to save textures will not work in 'WebPlayer' targeted developments.

Runtime Modifier Example.

A basic example has been included in the project so you can see how to make new clouds or modify existing ones by script in runtime. The example is very simple and do both, all the code is commented.

Last Notes.

Since version v2.0, CloudsToy uses the shuriken particle system instead of the legacy used in previous versions. A lot of time has been invested in getting the clouds look exactly the same than before (getting the shuriken system up and running wasn't so difficult) and some features -cloud animation- has been improved since shuriken can make it by default (instead of using the update to make the animation manually).

Notice that the shaders used in CloudsToy can be changed by dragging others in the inspector. This could be useful if you can't see the clouds in a specific platform and no runtime error popup in your specific compiler console IDE output error watcher (in that case usually the cause is the shader).

The CloudsToy important code is CloudsToy.cs, CloudsToyEditor.cs, CloudParticle.cs and CloudMovement.cs (and the runtime example: CreateCloudsToy.cs).

Any other script is mostly used to make possible the procedural Texture procedure (noise scripts, ProceduralCloudTexture.cs) which is not used at all in runtime, some First Person Controller basic stuff and some basic utilities. So if you want to change something and you have bought the SRC version, now you know where to look mostly CloudsToy.cs & CloudParticle.cs scripts.

CloudsToy has been tested in Windows, Mac OS, Android and IOS.

That's basically all the important stuff you have to know. Play with the cloud system a little bit and you'll see how fun is it. **NOTE:** With the appropriate skybox the clouds will look much better.

Jocyf 2010-2025.
Contact: jocyf@jocyf.com
Web page: www.jocyf.com
Forum threat: Unity forum threat
CloudsToy web page: www.jocyf.com/utilsEnglih.html