Introduction

First I would like to thank you for purchasing the 2D Sky package. This package was developed for some of my 2D and 2.5D games and it has proven incredibly useful in-house. Because of its ease of use and almost endless configurability I hope that you too will find it useful in your projects. Again thank you and enjoy. Now let's get started.

Quick Start

- Open Unity and create a new project.
- Import the 2D Sky package from the Unity Store.
- Open the Demo scene located under the "CygensGames/Sky2D" folder in the Project tab (see figure 1).
- There is no need to save changes if prompted.
- Hit play. That's all there is to it.

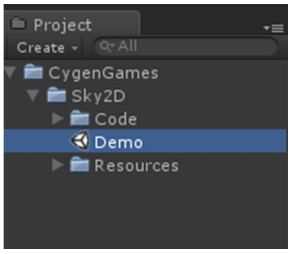

Figure 1

Using the 2D Sky Creation Wizard

- In a new scene or in an existing scene that does not already contain a 2D Sky open the 2D Sky creation wizard located under GameObject->Create Other->2D Sky/Sky… (see figure 2).

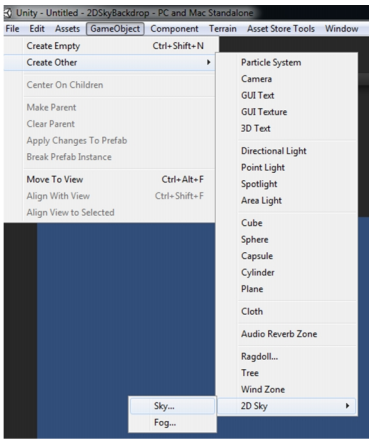

Figure 2

- A new wizard window will open (see figure 3). There are a lot of options in this wizard that will be discussed in the next section, but for now we will keep it simple.

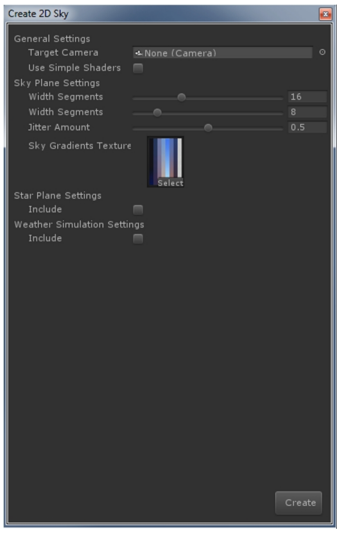

Figure 3

- Under General Setting->Target Camera select the main camera.
- Check the Include box under Star Plane Settings.
- Check the Include box under Weather Simulation Settings.
- Leave the rest of the settings at their defaults. Your wizard should now look like this (see figure 4).
- Once you have verified the settings hit the [Create] button to build the sky.
- Hit play to see your 2D Sky in action.

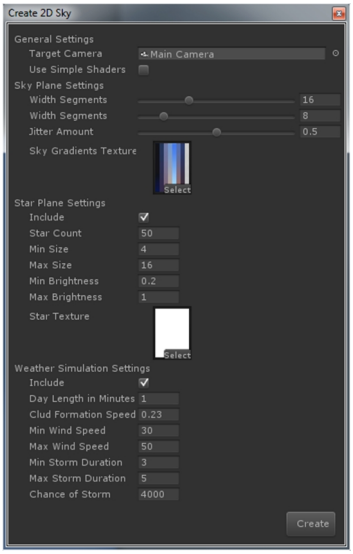

Figure 4

2D Sky Creation Wizard Settings

General Settings

Target Camera – This is the camera that you wish to parent the sky to. The sky will automatically size itself to fill the entire camera that it is parented to. Note: A target camera is not required but if one is not specified then the sky will have to be positioned and scaled manually or through your own custom scripts.

Use Simple Shaders – When checked this will tell 2D Sky to render using a simplified version of the Sky shader. While this isn't as pretty as the full shader it will improve performance on devices where the full shader is too slow.

Sky Plane Settings

Width Segments – This is the number of horizontal segments to divide the sky into.

Height Segments – This is the number of vertical segments to divide the sky into.

Jitter Amount – The amount of displacement applied to each interior vertex. This in conjunction with the width and height segments determines how painterly the sky will look (see figures 5 & 6).

Figure 5

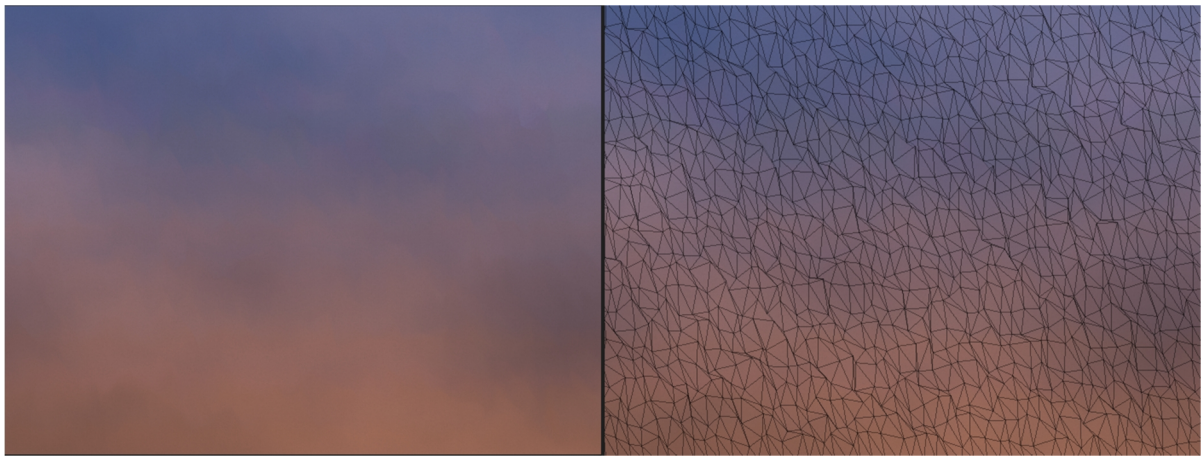

Figure 6

Caution – Increasing both height & width segments too high can cause your game to slow down significantly. I've found that a 2 to 1 ratio works best with twice the number of segments on the long side of your display. (I.e. using 32x16 for a landscape game) Also mobile devices won't run as fast as a desktop using the same settings.

Sky Gradients Texture – This is a specially formatted texture that provides the base information that the sky system uses to build the backdrop. A default texture is provided, but you don't have to use it. Feel free to create your own using the included texture as a guide (see figure 7).

Figure 7

Guidelines

- 256 x 32 pixels
- Texture is divided into eight 32x32 regions
- The first seven regions from the left represent various times of day starting at midnight.
- The eighth region is a seamless noise pattern to breakup color banding.
- The alpha channel contains a seamless noise pattern that is used for the clouds (see figure 8).

Figure 8

Star Plane Settings

Include – When checked this will add a star plane to the sky and make the relevant options available.

Star Count – This is the number of stars visible in the sky.

Min Size – This is the minimum size of the smallest stars.

Max Size – This is the maximum size of the largest stars.

Min Brightness – This is the minimum brightness of the faintest stars.

Max Brightness – This is the maximum brightness of the brightest stars.

Star Texture – This is the texture that will be applied to each star in the sky. This texture requires an alpha mask to work correctly. Again a default texture has been provided for you (see figure 9) but feel free to create your own.

Figure 9

Weather Simulation Settings

Include – When checked this will add a basic weather simulation to the sky and make the relevant options available.

Day Length in Minutes – This determines how many minutes in real-time that it takes to complete one day / night cycle.

Cloud Formation Speed – This determines how rapidly cloud formations change with higher numbers resulting in faster changes to the clouds.

Min Wind Speed – The minimum speed the clouds will move in either direction.

Max Wind Speed – The maximum speed the clouds will move in either direction.

Min Storm Duration – The minimum time if hours (scaled to day length) a storm will last.

Max Storm Duration – The maximum time if hours (scaled to day length) a storm will last.

Chance of Storm – The chance is one out of this number that a storm will occur at some point in the day. Note: with the basic storm simulation storms will only occur during the day time.

Scripts

Overview – This section will cover the scripts that are included in this package, their function, and any specific details required for their use.

Sky.cs

Description – This is the main script and is required on a GameObject that you wish to make into a Sky object.

Function – This script creates and updates the sky and the sky shader.

Specifics – In addition to the settings available from the creation wizard (see the 2D Sky Creation Wizard Settings Section) there are several other settings available from the inspector tab (see figure 10).

Cloud Color – This is the color of the clouds, with the alpha channel driving the cloud intensity.

Cloud Cover – This is the amount of cloud coverage, with zero being no clouds and one all clouds.

Sky Intensity – This is how dark the sky is, with zero being black and one being full sky color. This value is useful for making the sky look stormy or for special effects like forest fires.

Wind Speed – This controls the speed and direction of cloud motion.

Time of Day – This controls the time of day in hours from 0 to 24 with zero being midnight.

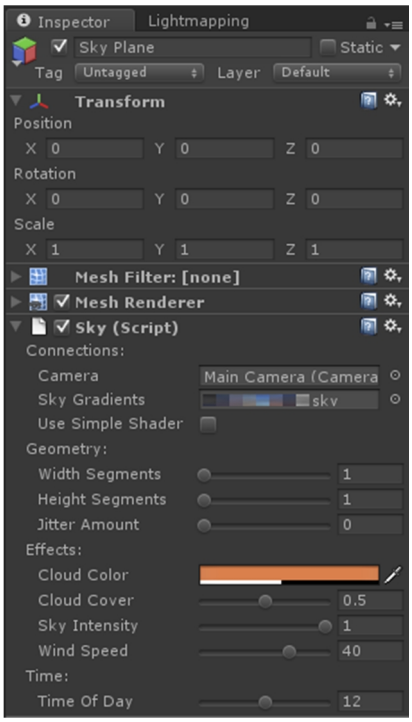

Figure 10

Stars.cs

Description – This is an optional script that can be added to a GameObject to create a star plane.

Function – This script creates and updates the stars.

Specifics – In addition to the settings available from the creation wizard (see the 2D Sky Creation Wizard Settings Section) there is one addition setting that must be set (see figure 11).

Sky – This is a connection to the Sky.cs script and it is required for the Stars.cs script to function properly.

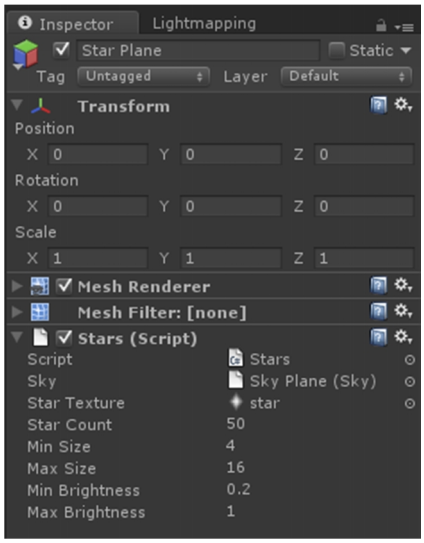

Figure 11

BasicWeatherSimulation.cs

Description – This script is included to demonstrate the power of the 2D Sky package and has some basic functionality that you can use in your own projects as is or you could customize this simulation or even write your own using the framework provided.

Function – This script acts as an entry point for the weather simulation which is a collection of decorator classes that each in turn control one aspect of the simulation. To see how the weather simulation is put together start by looking at this script and then into each of the decorators found in the "CygenGames/Sky2D/Code/SkySimulations" folder and its subfolders. A full discussion of this framework is beyond the scope of this document but a tutorial series is being put together (or may be out already) that will discuss its inner workings in much more detail.

Specifics – There are no addition inspector settings apart from those already covered (see the 2D Sky Creation Wizard Settings Section). This script however must be attached to the same GameObject that contains the Sky.cs script.

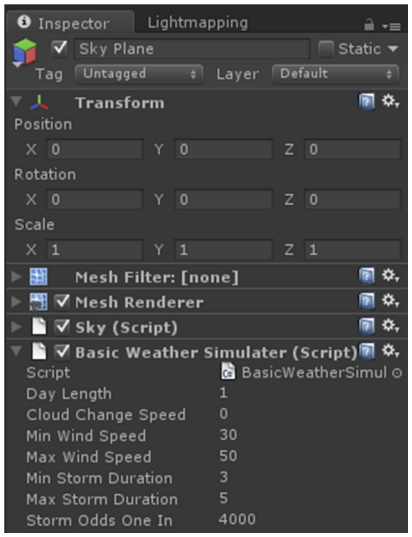

Figure 12

Shaders

CG_Sky.shader – The full sky shader that includes two layers of animated clouds.

CG_SimpleSky.shader – A simplified version of the sky shader that only uses on layer of animated clouds to improve performance on slower devices.

CG_Star.shader – Shader used to display the stars. This is a simple alpha transparency shader that used vertex color to tint the starts.

Prefabs

2D Sky – This is a starting prefab for those who prefer not to use the creation wizard. Simply drag this prefab into the scene and you are ready to go. Optionally this prefab can be parented to a camera and then connect Target Camera setting to the camera to achieve the same functionality provided from the creation wizard.

Textures

sky.tga – This is the default texture used to generate the procedural skies.

star.tga – This is the default texture used for the stars.