GLOBAL SNOW

Reference Guide

Kronnect

Contents

Introduction

Thanks for purchasing!

Global Snow is a drag & drop solution that converts your scene into a winter landscape with a couple of clicks. Just add a single script to your camera and customize the results.

Global Snow features:

- Realistic snow with relief shading, occlusion and glitter.
- Automatic snow coverage with exclusion options (by layers, per gameobject, mask texture).
- Footprints and pack of footfalls sounds.
- Terrain marks and terrain collision detector.
- Snowfall.
- Camera frost effect.

How it works

Global Snow supports forward and deferred rendering.

- **Deferred rendering (recommended)**. In this mode Global Snow will modify the GBuffers in realtime producing a great looking effect at the best performance.

- **Forward rendering**. In this mode Global Snow uses a full-screen image effect to blend a precomputed snow layer over the landscape. In this mode, Global Snow uses two techniques: a) renders the scene using custom replacement shaders with snow, and b) renders snow in screen space using a fast shader aimed for far distances. The result is a great looking snow but the performance is not as good as in deferred rendering.

- **Decal approach**: all Global Snow customers have access to a special version of Global Snow that uses decal approach to render snow over existing objects. This version can be appropriate for low-end devices which can't afford full-screen blits.

TIP !
Deferred and forward rendering path support are provided by custom shaders located in Workflow_Deferred and Workflow_Forward inside GlobalSnow/Resources folder. If you're only using one of the two rendering paths (eg. deferred), then you can remove the other folder so save some resources and build time in your project.

Demo Scene

Just load the demo scene included and click "Play". You will be able to move around using WASD and mouse. Press spacebar to jump, T to toggle on/off the snow effect or L to change daylight.

You should delete the Demo folder entirely or ignore it when importing the asset into your real project.

# Quick Start

1. Add the GlobalSnow script to your main camera in your scene.

2. Check the Scene Setup parameters if needed and choose a Quality preset.

3. Optionally enable footprints, terrain marks, snowfall or camera frost features.

You can of course customize any of its parameters to match your game mood and requirements.

Please read this document to learn more about all supported features.

Global Snow Parameters

Scene Setup (Forward Rendering)
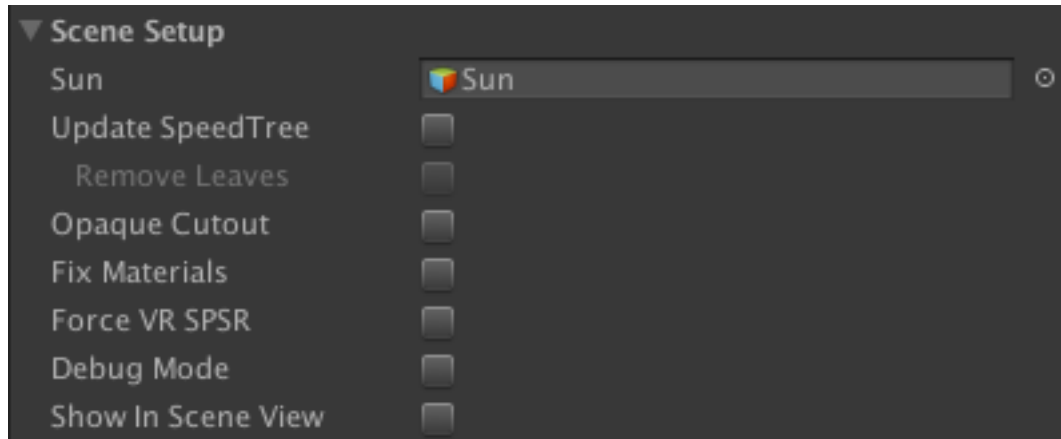
Sun: make sure you have a proper directional light assigned as the Sun. It's necessary to compute relief and occlusion effects. Global Snow will pick a directional light automatically for you but ensure this is the good one.

Global Snow uses custom replacement shaders to render the snow over the scene geometry. By default all opaque objects will be covered by snow (see FAQ section for exclusion options). The following setup parameters will help some common shaders work with Global Snow:

Update Speed Tree: enable this option to support SpeedTree and SpeedTree billboard objects.

Remove Leaves: enables a SpeedTree variant shader that removes tree leaves. For best results, it's recommended that you use custom Winter tree models. However this option allows you to quickly prune all tree leaves if you want.

Billboard Coverage: specifies the amount of snow over tree billboards and grass objects.

Opaque Cutout: by default, opaque objects will be covered by snow. But some shaders expose holes or transparent parts of a texture and still they use the Opaque rendertype. Enabling this option will force GlobalSnow to discard those transparent pixels preventing them from being covered by snow (in this case a transparent pixel needs to have RGBA=0,0,0,0 value). Some trees may require you to to enable this option so their leaves render appropriately with Global Snow.

Fix Materials: this option will force all materials to expose a RenderType tag. It's a good recommendation for shader developers to specify a RenderType per shader. However not all shaders expose this tag. Because of this, it's possible that some objects get clipped by the snow. If this happens to your scene, enable this option to force all non-transparent materials to expose an Opaque render type.

Force VR SPSR: this option is only available in Unity 5.5 or higher. Will enable compatibility with OpenVR. Don't use if your VR SDK is Oculus.

Scene Setup (Deferred Rendering)
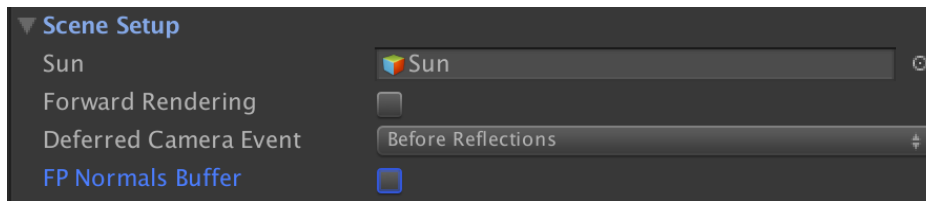
Sun: make sure you have a proper directional light assigned as the Sun. It's necessary to compute relief and occlusion effects. Global Snow will pick a directional light automatically for you but ensure this is the good one.

Forward Rendering: enable this option to override deferred rendering and use the forward rendering mode of Global Snow.

Deferred Camera Event: this option specifies when the snow shader passes will be executed. By default this setting is BeforeReflections and that should be the "good setting". However, if you disable deferred reflections in Graphics Settings, you will need to change this to "Before Lighting".

FP Normals Buffer: enable this option to use floating-point backed normal buffer (16 bit precision) instead of normal 8-bit precision. Use it only if you see snow artifacts in the distance.

Unity standard tree and billboards (only in deferred mode)

- Inside the folder Resources/Workflow_Deferred/Shaders/Grass & Billboards you can find a package containing shader replacements for the default grass and tree billboards in case you need to support them.

- Also you can use the alternate "TreeOpaque Soft Occlusion Bark" shader located inside the same folder (subfolder Tree).

- SpeedTrees are supported in this mode so you don't need to change them.

Quality

This section offers main quality/performance options:
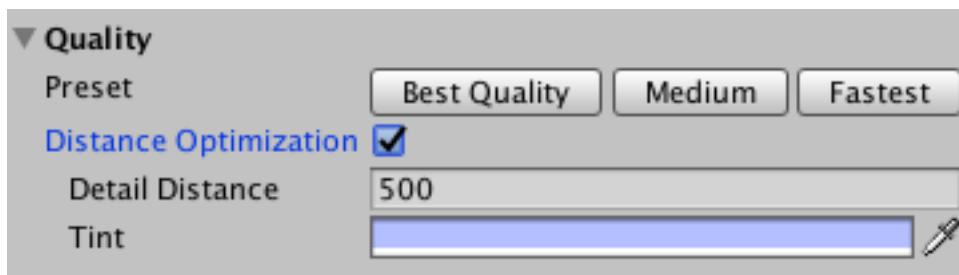
Best Quality: uses a high resolution coverage texture, relief shading, occlusion and glitter effects.

Medium: uses a smaller resolution coverage texture, relief shading, occlusion and glitter effects. It also enables Distance Optimization option with a default value of 500m.
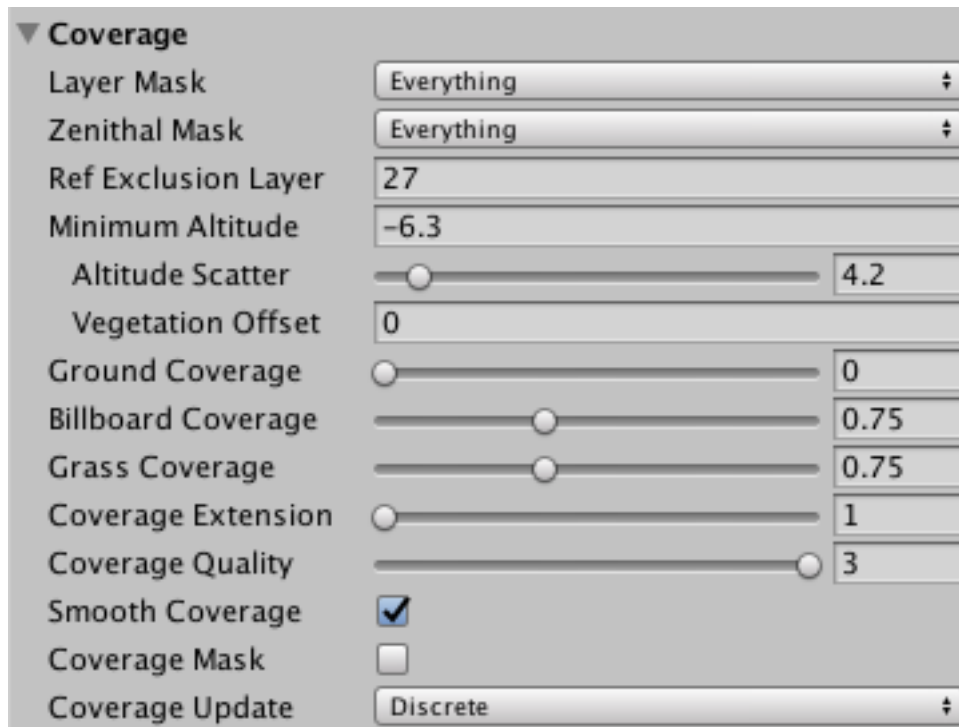
Fastest: uses a smaller coverage texture and flat shading (which still looks good!). It also enabled Distance Optimization beyond 100m.

Note that these presets just apply some factory values to other settings across the inspector. They are provided for convenience, as shortcuts to other options.

Distance Optimization: when enabled, a secondary simpler and faster algorithm is used to render the snow beyond some given distance from the camera. A "Tint" option allows you to specify the color for the far distance snow. Since this algorithm is not physically-based, the snow rendered this way is not influenced by ambient light nor global illumination. It can't also receive shadows. Additional options are self-describing (pass mouse cursor over the option label to get a short description).

Coverage

This section allows you to customize where and how the snow will be rendered:

Layer Mask: determines which objects can be "painted" with snow. By default, all transparent objects are excluded since snow is rendered before transparent queue.

Zenithal Mask: determines which objects can occlude other beneath them. By default, any object will cancel any snow on other objects beneath them. Make sure to exclude any particle system here both to improve performance and to reduce coverage issues.

Ref Exclusion Layer: this is the number of the layer, used during the rendering of snow, for objects that have been marked specifically to be excluded by snow. Use a layer number that you don't use in your project.

Minimum Altitude and Altitude Scatter: specify the starting altitude for the snow. Use Altitude Scatter to make the snow appearance more realistic. Note that above the minimum altitude, the snow will always appear. There's no "maximum altitude".

Vegetation Offset: specify a value to be added to the minimum altitude for trees and grass. With this option you can enforce vegetation to be covered by snow on greater altitude than terrain starts showing snow.

Coverage Extension: a value of 1 means an area of 256x256 meters around the camera position. A value of 2 means an area of 512x512 meters. This parameter along with Coverage Quality determines the final resolution of the snow coverage.

Coverage Quality: determines the size of the internal texture used for snow coverage computation. A value of 1 uses a 512x512 texture, a value of 2 uses 1024x1024 texture and a value of 3 uses a 2048x2048 texture.

Smooth Coverage: enabling this option will smooth the coverage producing more pleasant results.

Coverage Mask: you can use a texture as a mask to specify the areas of the world that will be covered by snow. This feature assumes that terrain is centered around 0,0 position. The alpha channel of the texture will be used to determine where can be snow (1=snow, 0=no snow). The world size parameter specifies the mapping size of the texture over the world. For instance, a texture of 2048x2048 pixels with a World Size of 2048x2048 will have a mapping of 1 meter per pixel (the Y-coordinate is ignored).

Appearance

This section allows you to customize the appearance of the snow itself:
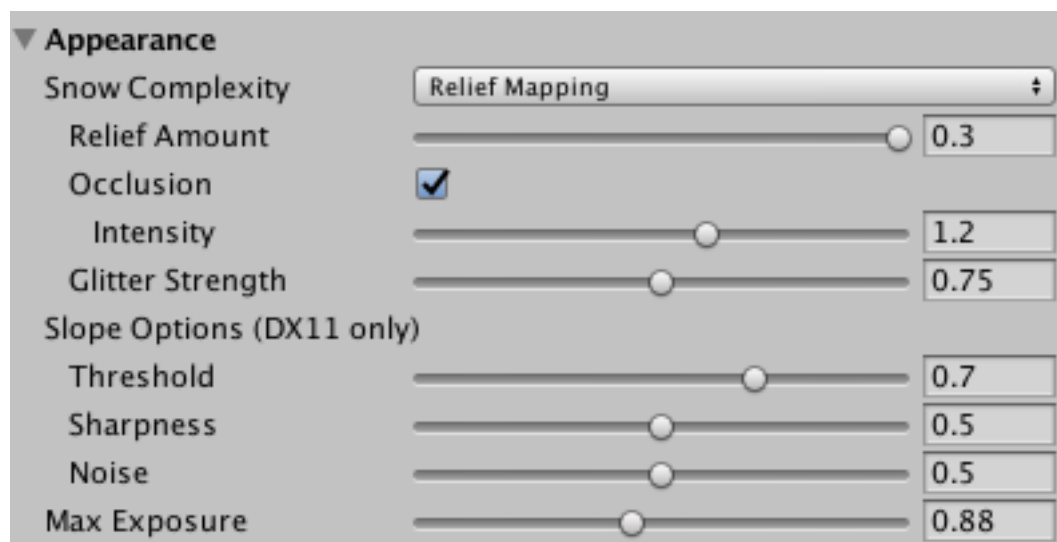
Snow Complexity: Flat shading is the fastest and produces a flat and easy looking snow. Relief mapping uses a limited raytracing algorithm to simulate 3D waves over the snow.

Occlusion: adds extra local shadows to the snow. Only available when relief shading is used. This option is useful when the Sun is high in the sky because from the zenith the snow will look very bright and white without occlusion.

Glitter Strength: controls the amount of glittering over the snow. Glitter tends to glow stronger when facing Sun.

Slope Options: customize the borders of the snow covered areas, useful for adjusting the look of snow covered cliffs and mountains. Note that this feature only works on DX11 and OpenGLCore compatible graphics.

Max Exposure: let you limit the amount of light that the snow will reflect avoiding over exposure when Sun is very high or its light intensity is greater than 1. Note that this is an artificial change and can produce odd results if abused when compared with the rest of non-covered geometry.

Features

This section allows you to unlock special Global Snow features:
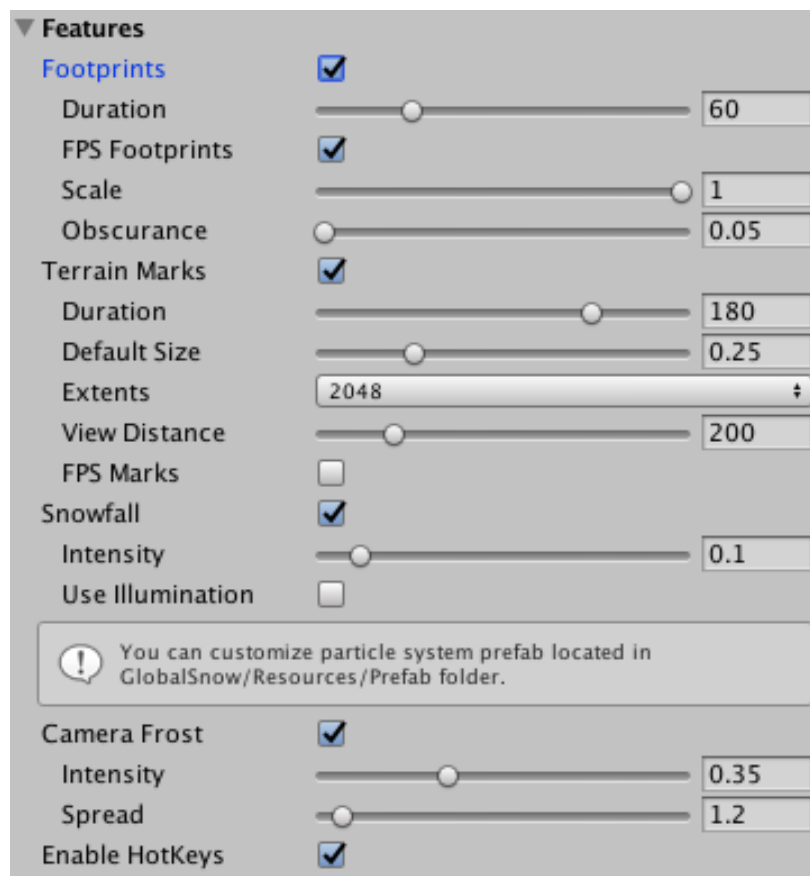
Footprints: footprints are human-shaped footprints decals that are printed over the snow. This feature is interesting to help the player walk over large distance and be able to return over the same path. Although Global Snow only supports adding footprints at the camera position, it could be modified to add custom footprints over other areas.
- Enable **FPS Footprints** to add automatically footprints after your FPS character moves.
- Use **Duration** parameter to customize the fade out duration of the footprints.

Terrain Marks: this feature allows you to cause small or medium impacts over the snow.
- Choose the **default size** in meters for the radius of the mark.
- The maximum **Extents** is the size of the internal control texture that's used to render marks over the terrain – increase to have a greater area of non-repetitive marks.
- **View distance** allows you to control the rendering distance of the terrain marks.
- Enable **FPS Marks** to leave automatic footprint tracks using terrain marks.

Snowfall: enable this parameter to instantiate the SnowParticleSystem prefab located in Resources/Prefabs. This particles system will emit snow flakes at **Intensity** rate. You can customize the look and behaviour of the snow particle system modifying the prefab itself. Enable "**Use Illumination**" to change the particle material to a Standard Shader material which will be affected by Global Illumination.

Camera Frost: adds a screen-space frosting effect which can be customize by intensity and screen spread. Useful to add a temperature sense to the player.

Enable HotKeys: allows you to change snow altitude at runtime pressing keys "K" and "I".

Snow Volumes

Global Snow volumes are special areas where snow can be completely disabled. When the player enters a snow volume, the snow disappears and when he exists the area, the snow script is re-enabled automatically.

To add a Global Snow Volume to your scene, just drag & drop the GlobalSnow Volume prefab from the GlobalSnow/Resources/Prefabs folder.

Position the volume in your scene as you do with a normal object. Change the collider scale to match the desired area size.

Mask Editor

Another way to remove/add snow on custom areas is by using a coverage mask:
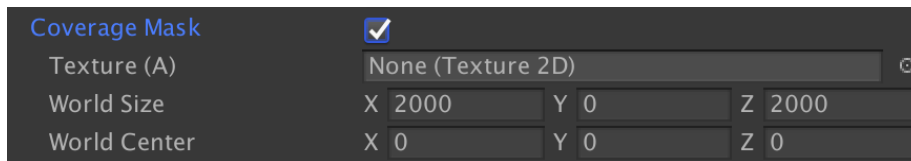
A coverage mask is a special texture whose alpha component determines the opacity of the snow at a given location. The texture is overlaid over the world space at "World Center" with extents of "World Size" (so a size of 2000 means a coverage of -2000 to 2000).

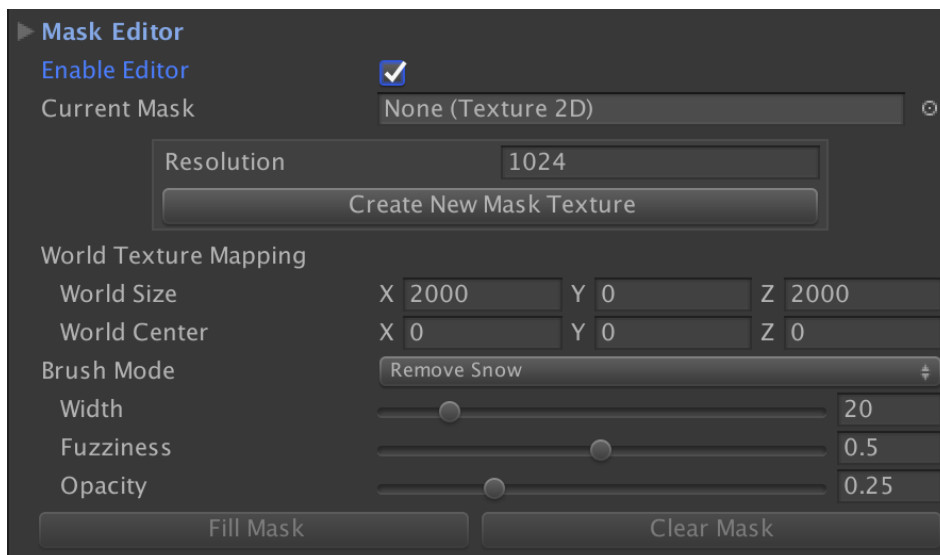You can provide your own texture mask or use the integrated SceneView editor. This mask editor is enabled using the provided section in the inspector:

The Mask Editor allows you to create a new mask texture from scratch. Select a proper resolution according to the detail level you need. For gross results, a texture of 1024x1024 may be enough. If you need finer coverage details use a bigger texture (ie. 2048, 4096 or even 8192 pixels).

The brush mode, width, fuzziness and opacity controls the behaviour of the paint tool. When the Mask Editor is enabled you can remove or add snow into the SceneView directly!

Click "Fill Mask" to reset the mask texture to fully snowed. "Clear Mask" will remove all snow from the covered area.

Please note that if you use Unity standard tree or grass you also need to edit the GlobalSnowForwardOptions.cginc or GlobalSnowDeferredOptions.cginc file (depending on the active camera rendering path) and enable the #define GLOBALSNOW_MASK line.

World Manager API integration

World Manager API is a component developed by Adam Goodrich that enables syncying some atmospheric parameters across different assets.
You can find more information on GitHub: https://github.com/adamgoodrich/WorldManager

To enable World Manager API integration, make sure the corresponding checkbox is checked at the bottom of Global Snow inspector:

(Of course WMAPI asset needs to be present in the project for the integration to work).

The current integration allows snow altitude to change depending on the minimum snow altitude and season parameters of WMAPI. Also the snow fall intensity is synced with WMAPI properties.

Frequest Asked Questions (FAQ)

Some objects that get clipped when Global Snow is enabled (forward rendering)

Global Snow uses replacement shaders to render a snow-covered version of the geometry. This snow layer is rendered before transparent objects as a screen-space effect.

Objects that render during the transparent queue won't be clipped by Global Snow, so this issue should only happen when some opaque shaders don't expose the RenderType appropriately (which is a recommendation). **Try using Fix Materials option in Scene Setup section of the inspector**.

Another solution to this issue involves editing that object shader and set the RenderType tag to one of the following:

- "**Opaque**": this render type will cover the geometry with snow, and also allow footprints and terrain marks. It can also expose holes or transparent parts not covered by snow using alpha channel IF the "OpaqueCutout" parameter is enabled in Global Snow inspector.
- "**TransparentCutout**": same than before but will also take into account the alpha channel to discard those pixels REGARDLESS of the OpaqueCutout parameter.
- "**TerrainOpaque**": same that "Opaque" but will ALWAYS ignore alpha channel. Commonly used for full opaque objects.
- "**AntiSnow**": will cause the geometry to prevent any kind of snow covering. Useful for water or liquid objects that render in the opaque queue or anything you want to exclude at shader level. Note that you can also exclude objects from GlobalSnow using the Layer Mask parameter under Coverage section in the inspector or adding the GlobalSnowIgnoreCoverage script to that gameobject.

When you edit the shader, the change should read like this (other tags should not be touched!):

```
SubShader {
    Tags { "RenderType"="Opaque" ... }
    ...
}
```

Other render types supported are:

- "**TreeOpaque**": used for covering opaque parts of standard Unity tree.
- "**SnowedSpeedTree**": used as a replacement for the SpeedTree shader. This tag is used when you enable the "Update SpeedTree" parameter in GlobalSnow inspector.
- "**SnowedSpeedTreeBillboard**": same but for the billboard trees.
- "**Grass**": used to cover standard Unity grass objects.
- "**GrassBillboard**": used to cover standard Unity grass billboard objects.

How can I exclude some objects from being covered by snow?

You have three options here:

1) Add those objects to a custom layer and exclude that layer in the Layer Mask parameter of Global Snow (Coverage section).
2) … or simply add "GlobalSnowIgnoreCoverage" script to those objects (then no need to change their layers).
3) … or edit the shader of those objects and set the RenderType to "AntiSnow".

There're no snow marks on some objects, like floating platforms. They only work on the terrain.

You need to add the GlobalSnowCollisionDetector script to any geometry (except for the terrain) if you want it to support terrain marks when collisions occurs. Demo scene contains some geometry that allows terrain marks over them using this approach. You can also add custom terrain marks using code, see below.

How can I manually add terrain marks over any place?

You can use this code to add any number of marks to the terrain or geometry that's covered by snow:

```
GlobalSnow snow = GlobalSnow.instance;

snow.MarkSnowAt(position, size);
```

Where position is a Vector3 in world space coordinates and size is a value between 0 and 1.

Can I customize footprint shape?

Yes, the footprint feature uses the Footprint texture located in Resources/Textures folder. Feel free to use another texture for your footprints ☺

Alternatively you can use the Terrain Marks feature to generate footprints-like marks as your FPS character moves (enable the FPS Marks option).

Can I change the size of the terrain marks?

You can specify a default mark size using the GlobalSnow inspector or if using the API you can pass a value between 0 and 1. You can also add GlobalSnowColliderExtraInfo script to any object and set the mark size individually in this script.

I see terrain marks over the distance, but nothing should be causing them.

Increase the **Extents parameter** or reduce the **View Distance parameter** in the Terrain Marks section of Global Snow inspector.

Is there any footfall option?

You can find 3 snow footfalls audio clips inside GlobalSnow/Resources/Footfalls folder. Assign them to your First Person Controller script (footsteps sounds property).

Note that there's currently no way to determine if there's snow under players feet. You could use a RayCast from above the player position to check if it's under cover and switch between normal footfalls and snow footfalls.

How to enable snow reflections on water? (forward rendering)

Support for snow reflections in Unity Water Pro system from Standard Assets is included.
For instructions on setting up, please check the README file inside GlobalSnow/Extras/WaterReflections folder.

Does it work with VR/Single Pass Stereo Rendering?

Yes, it has been tested with Oculus and OpenVR SDKs. For OpenVR you need to use Unity 5.5 or higher and enable the option "Force VR SPSR" available in the Scene Setup section.

It's recommended to use deferred rendering since the performance will be better.

Unity's standard tree billboards are not covered by snow. Is it possible to use them?

Forward rendering mode

Standard Tree billboards render as transparent objects hence you need to make a couple of changes:

1) Edit GlobalSnow.shader and uncomment the last SubShader section related to TreeBillboard.
2) Edit GlobalSnow.cs script and comment out [ImageEffectOpaque] just before OnRenderImage method.

Deferred rendering mode

Locate the GrassAndBillboardsReplacementShaders package inside the folder Resources/Workflow_Deferred_Shaders/Grass & Billboards and import it.
Once imported, they should automatically override the default shaders (it's possible that you need to reload the scene).

Compilation takes a long time / GlobalSnow adds many shader variants, can I reduce them?

If you use deferred mode, you can rename GlobalSnow.shader file to something like "GlobalSnow.shader.old" inside the folder Resources/Workflow_Forward/Shaders. This way, Unity won't waste time compiling this complex snow shader.

If you use forward rendering mode, edit GlobalSnow.shader file and look for these groups of keywords:

#pragma multi_compile __ GLOBALSNOW_FLAT_SHADING GLOBALSNOW_RELIEF GLOBALSNOW_OCCLUSION

#pragma multi_compile __ GLOBALSNOW_OPAQUE_CUTOUT

#pragma multi_compile __ GLOBALSNOW_FOOTPRINTS

#pragma multi_compile __ GLOBALSNOW_TERRAINMARKS

For example, if you really want to use terrain marks, you could replace this line:
#pragma multi_compile __ GLOBALSNOW_TERRAINMARKS
by
#define GLOBALSNOW_TERRAINMARKS 1

This change will enforce the use of terrain marks without multiplying the number of shader variants while also reducing compilation time.

Can I add "holes" into the terrain?

If you use a custom terrain shader that support holes, you can use the "CircularHole" or "QuadHole" prefabs located in GlobalSnow/Resources/Prefabs folder to erase the snow over the entrance of the hole.

These prefabs are just normal geometry what use a custom material "GlobalSnowEraser" and have the "GlobalSnowIgnoreCoverage" script attached. You can use any other geometry, assign the same material and script and it will act as a stencil object – snow will not appear behind that geometry.

Additional Support

Please visit kronnect.com's forum for questions, support and more info.
If you have any issue using GlobalSnow you can also reach us at contact@kronnect.me