Brute Force - Snow & Ice Shader Documentation

1. Brute Force - Snow & Ice Shader

Brute Force - Snow & Ice Shader Unity Asset Store

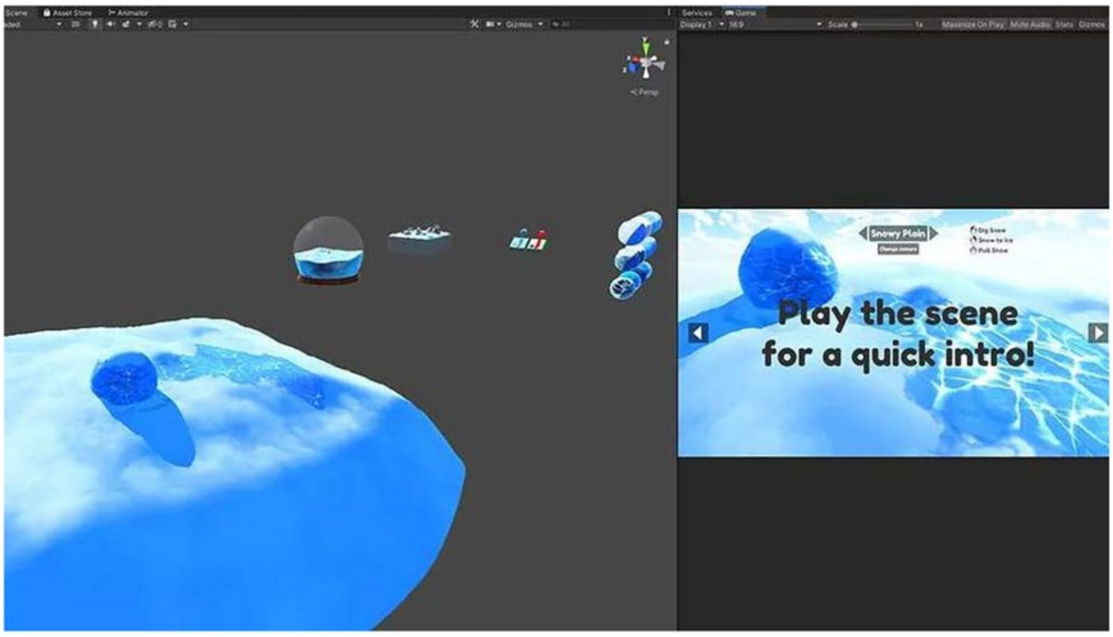
This is the **Brute Force - Snow & Ice** visual documentation.
Consider leaving a review on the asset store to support me c:
and if you have questions email me at: **bruteforcegamesstudio@gmail.com**

3. Compatibility and Features

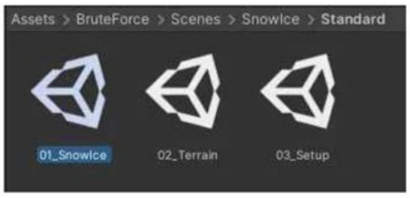
The **Brute Force Snow & Ice Shader** let's you create an **interactive fluffy snow** combined with **ice** for **PC/Mac/Linux/Nintendo Switch** and **mobile** devices. The same shader is used in my game Crumble
Compatibility with **Unity Standard** and **URP**

Features:
-**Mutlicompatibility**: works on PC/Mac/Linux/Nintendo Switch/WebGl and mobile devices
-**Simplicity**: Drag and drop materials
-**Tessellation**: Custom tessellation based on camera distance
-**Dynamic Custom Shadows**: Snow & Ice featuring custom colored shadows and shadow casting
-**Interactivity**: With a simple particle system you can create interactive effects like a trail on mouse/player position
-**Customization**: Fully customizable Snow & Ice, render the snow & ice with a single material
-**Additive Snow**: An experimental feature letting you add snow to any mesh with one simple component
-**Lights Support**: 4x point lights and spot lights support (with shadow casting)
-**Terrain**: Use texture Splat Map to draw snow & ice terrains, the asset features a compatibility tool to work with other terrain assets. The terrain feature is currently limited in variety, will be improved
-**VR Support**: Works with Multi and Single Pass Stereo rendering

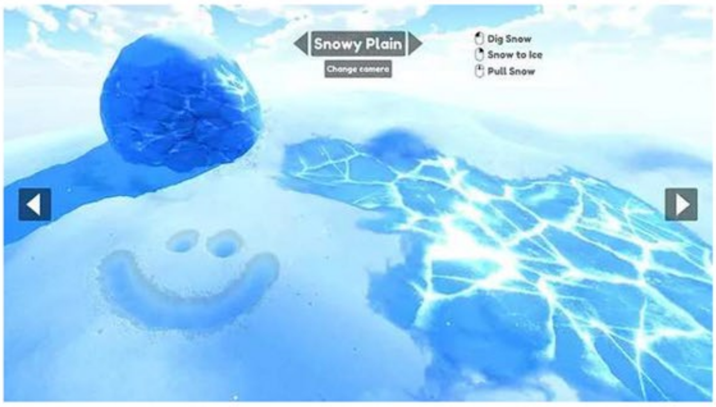
4. Getting started



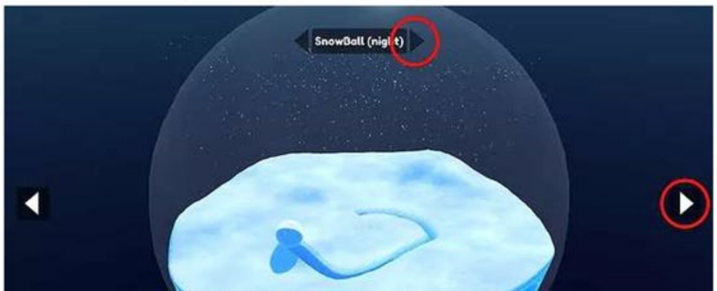
First thing you should do is load up the snow demo scene inside the asset package "**Assets\BruteForce\Scenes\SnowIce**" You will see that some files have a Standard and URP version, choose accordingly.



Once the "**01_SnowIce**" or "**01_SnowIceURP**" is loaded you will be prompted to play the scene for a quick introduction. This will help you understand the asset better and showcase all its basic features in a complete environment.
In the first showcase you can use **Left Click** to dig the snow and wet the ice, **Right Click** to change snow into ice, **Middle Click** to pull snow upward.

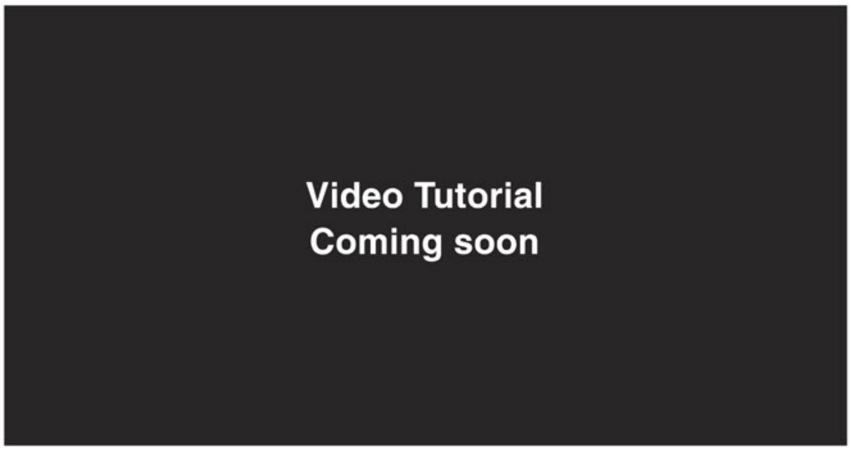


You can click the top arrows on the screen to switch sub-showcases and the left and right arrows to switch showcases.

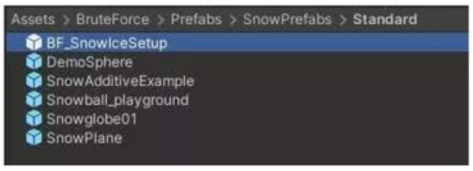


Additionally you can use WASD to move the ball when showcased, it works great as a player template leaving an interactive trail on the snow.

Importing it in your project and Components

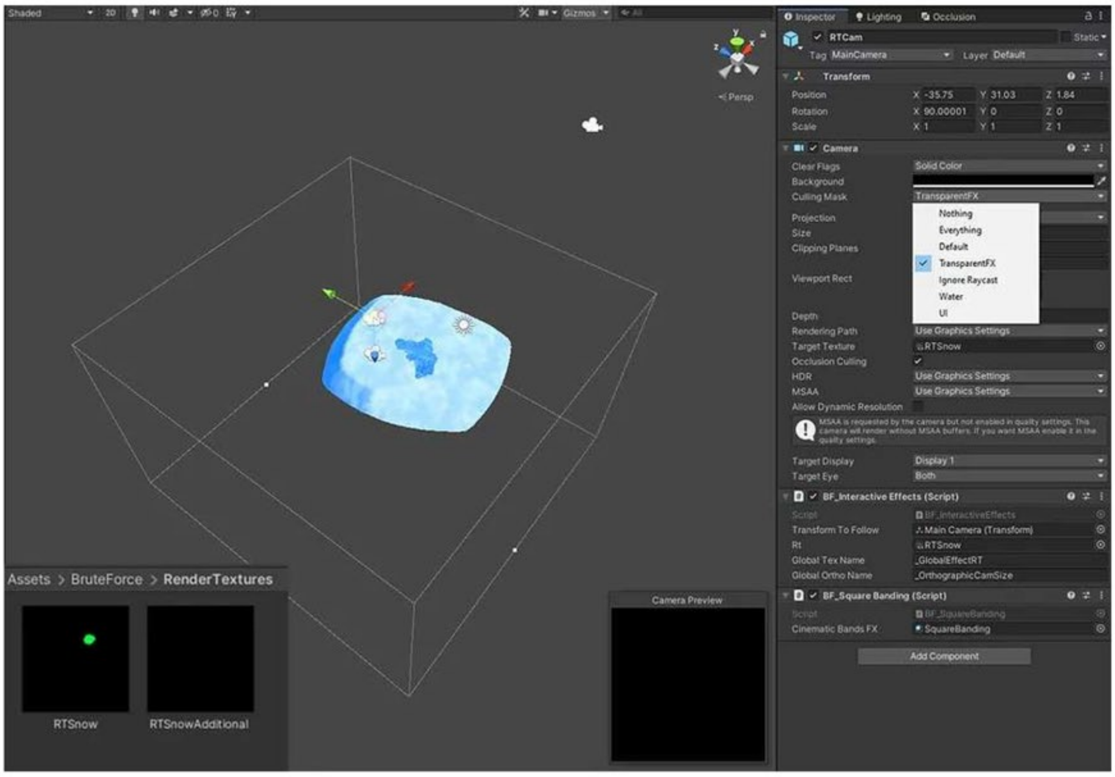


To append the snow assets in your projects you can either copy and import one of the working **showcase** as a starting template or you can use the **setup prefab**. Keep in mind that if you need the interactive feature you have to append the **CameraEffects** in your scene. I will go into details on what components you need in the prefab for the snow to work properly:



Assets > BruteForce > Prefabs > SnowPrefabs > Standard
- BF_SnowIceSetup
- DemoSphere
- SnowAdditiveExample
- Snowball_playground
- Snowglobe01
- SnowPlane

Camera Effect



Assets > BruteForce > RenderTextures

RTCam
Tag MainCamera Layer Default

Transform
Position X -35.75 Y 31.03 Z 1.84
Rotation X 90.00001 Y 0 Z 0
Scale X 1 Y 1 Z 1

Camera
Clear Flags: Solid Color
Culling Mask: TransparentFX
Projection
Size
Clipping Planes
Viewport Rect
Depth
Rendering Path: Use Graphics Settings
Target Texture: RTSnow
Occlusion Culling: ✓
HDR: Use Graphics Settings
MSAA: Use Graphics Settings
Allow Dynamic Resolution
MSAA is requested by the camera but not enabled in quality settings. This camera will render without MSAA buffers. If you want MSAA enable it in the quality settings.
Target Display: Display 1
Target Eye: Both

BF_Interactive Effects (Script)
Script: BF_InteractiveEffects
Transform To Follow: Main Camera (Transform)
Rt: RTSnow
Global Tex Name: _GlobalEffectRT
Global Ortho Name: _OrthographicCamSize

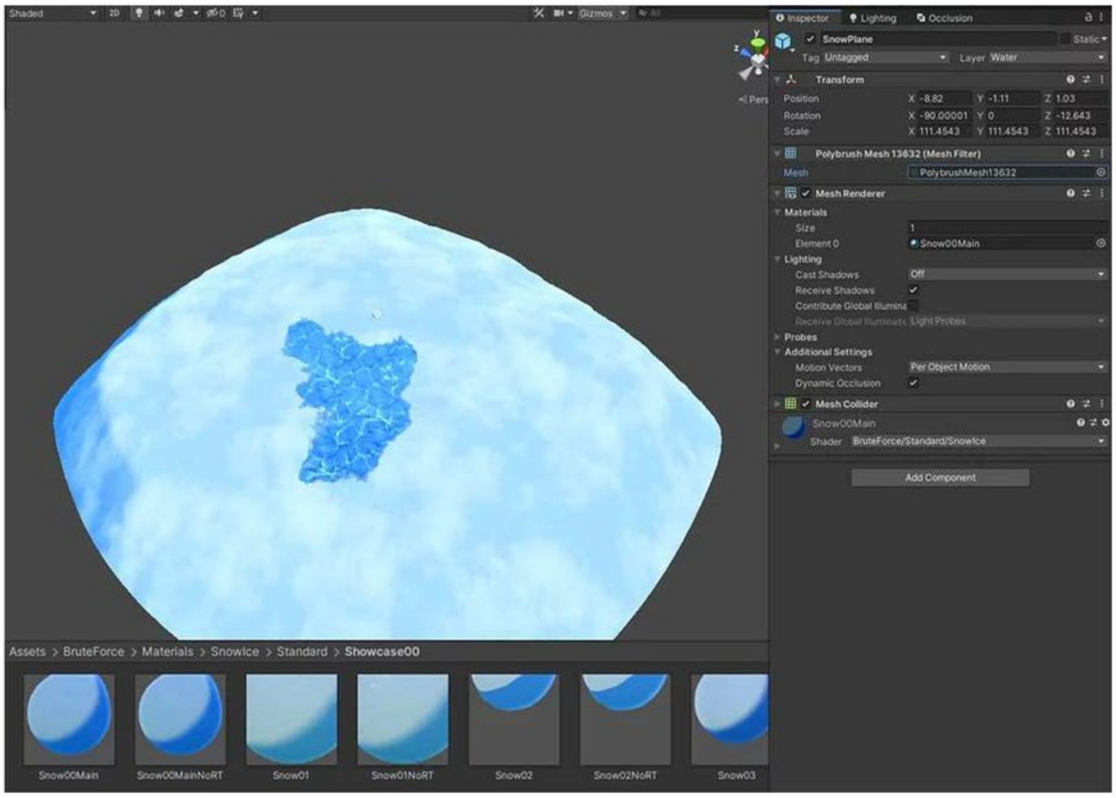
BF_Square Banding (Script)
Script: BF_SquareBanding
Cinematic Bands FX: SquareBanding

The first main component you'll need if you want **interactivity** on your snow are the "**Camera Effects**" like a regular camera it will be used for rendering, but instead this one will render to a target texture namely: "**RTSnow**" and "**RTSnowAdditional**" and will have only ONE culling mask set to "**TransparentFX**", we only need to render the effects.
It needs to have a target texture set to a render texture and it also needs the script "**BF_InteractiveEffects.cs**" for the first one and "**BF_InteractiveEffectsAdditional**" for the second one (which is optional, more details below). This script will make sure your cameras will render to the render textures.

If you have a moving camera you'll need to set a transform to follow (preferably the camera). RTCamAdd MUST have your main camera as transform value.
If you don't know what the global names are do not change them.
If you do not need the interactive part of the shader you can disable the CameraEffects and the "**Use RT**" in the material pragmas for a small performance boost.

You can increase the size of the camera projection to increase the radius of the rendered effect at a precision cost (more details below), do not rotate the RTcamera.

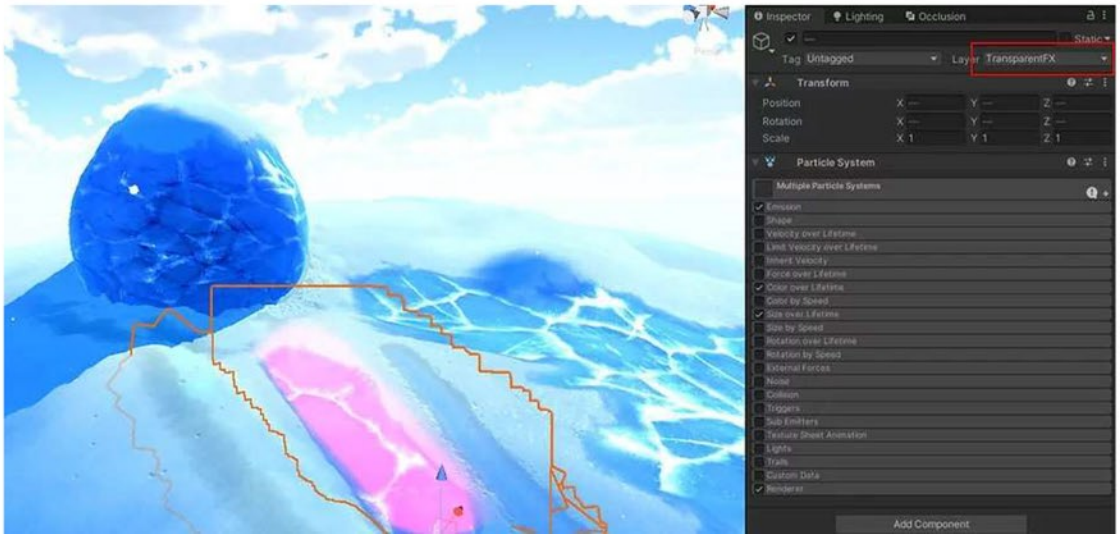
Snow/Ice Material



Inspector | Lighting | Occlusion
SnowPlane — Static
Tag Untagged — Layer Water
Transform
Position X -8.82 Y -1.11 Z 1.03
Rotation X -90.00001 Y 0 Z -12.643
Scale X 111.4543 Y 111.4543 Z 111.4543
Polybrush Mesh 13632 (Mesh Filter)
Mesh PolybrushMesh13632
Mesh Renderer
Materials
Size 1
Element 0 Snow00Main
Lighting
Cast Shadows Off
Receive Shadows ✓
Contribute Global Illumina
Receive Global Illuminatic Light Probes
Probes
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusion ✓
Mesh Collider
Snow00Main
Shader BruteForce/Standard/SnowIce
Add Component

I refer to the Snow/Ice shader simply as snow inside the project.
You can drag and drop any snow materials to a mesh or any assets you want to use as ground, I will explain in more depth how the Snow Shader works in the next page so you can be able to customize or create a snow material for your projects.

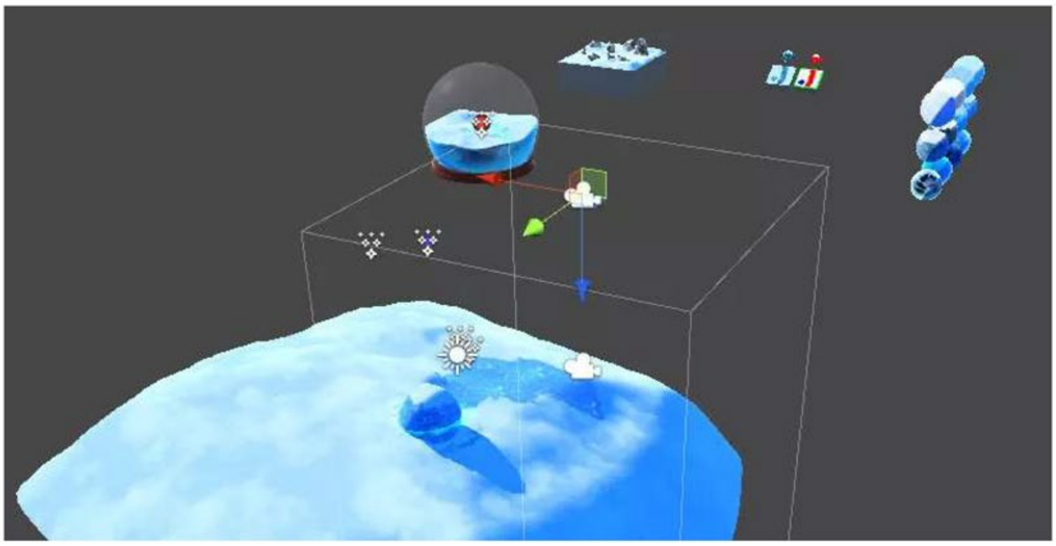
Particle System Effects


Inspector | Lighting | Occlusion
— Static
Tag Untagged — Layer TransparentFX
Transform
Position X — Y — Z —
Rotation X — Y — Z —
Scale X 1 Y 1 Z 1
Particle System
Multiple Particle Systems
✓ Emission
Shape
Velocity over Lifetime
Limit Velocity over Lifetime
Inherit Velocity
Force over Lifetime
✓ Color over Lifetime
Color by Speed
✓ Size over Lifetime
Size by Speed
Rotation over Lifetime
Rotation by Speed
External Forces
Noise
Collision
Triggers
Sub Emitters
Texture Sheet Animation
Lights
Trails
Custom Data
✓ Renderer
Add Component

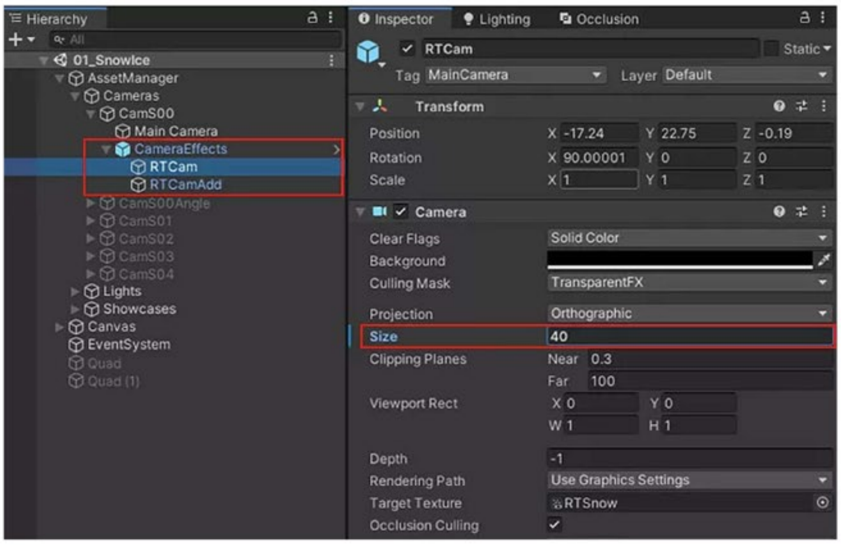
Particle Effect
Play Stop
Particles 299
Speed Range 0.0 - 0.0
Show Bounds
Show Only Selected

To control the interactive effect of the snow you can use a particle system set on the **TransparentFX layer**, the important thing to note is that only the rendered color matters:
Red color will turn snow into ice, **Blue** color will dig the snow, **Green** color will pull up the snow.
You can use anything that can be rendered to achieve the desired effect: meshes, sprites, particle systems etc...
This means you can place a **colored quad** in the **TransparentFX** layer to have a **permanent snow effect** in your scene.
Alternatively you can edit the "Duration" and "Start Lifetime" parameters of the particle system to either shorten or extend the effect duration.

Increasing the Render Texture size

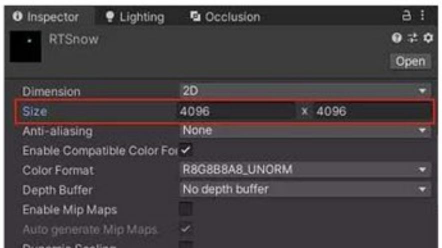


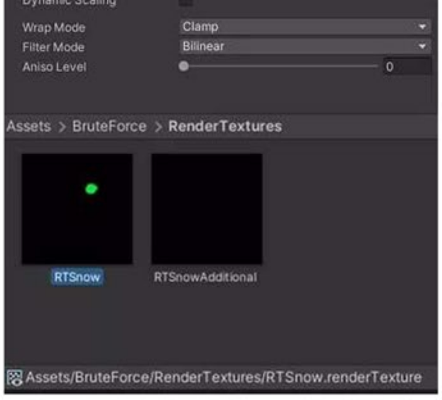
Depending on your project, you can increase the size of the Render Texture field so you would be able to see snow effects from much further away.
To do so you'll need to select your **RTCam** in your scene and increase the "Size" Camera variable, keep in mind that increasing this value will decrease the effect resolution.


Hierarchy
01_SnowIce
AssetManager
Cameras
CamS00
Main Camera
CameraEffects
RTCam
RTCamAdd
CamS00Angle
CamS01
CamS02
CamS03
CamS04
Lights
Showcases
Canvas
EventSystem
Quad
Quad (1)

Inspector | Lighting | Occlusion
RTCam — Static
Tag MainCamera — Layer Default
Transform
Position X -17.24 Y 22.75 Z -0.19
Rotation X 90.00001 Y 0 Z 0
Scale X 1 Y 1 Z 1
Camera
Clear Flags: Solid Color
Background
Culling Mask: TransparentFX
Projection: Orthographic
Size: 40
Clipping Planes: Near 0.3, Far 100
Viewport Rect: X 0 Y 0 W 1 H 1
Depth: -1
Rendering Path: Use Graphics Settings
Target Texture: RTSnow
Occlusion Culling: ✓

To increase the effect resolution you can change the "**Size**" of the **RenderTexture** to 2048x2048 or 4096x4096 by selecting the "**RTSnow**" in your project. Changing this value will greatly increase memory consumption.


Inspector | Lighting | Occlusion
RTSnow
Open
Dimension: 2D
Size: 4096 x 4096
Anti-aliasing: None
Enable Compatible Color For ✓
Color Format: R8G8B8A8_UNORM
Depth Buffer: No depth buffer
Enable Mip Maps
Auto generate Mip Maps ✓



Wrap Mode: Clamp
Filter Mode: Bilinear
Aniso Level: 0

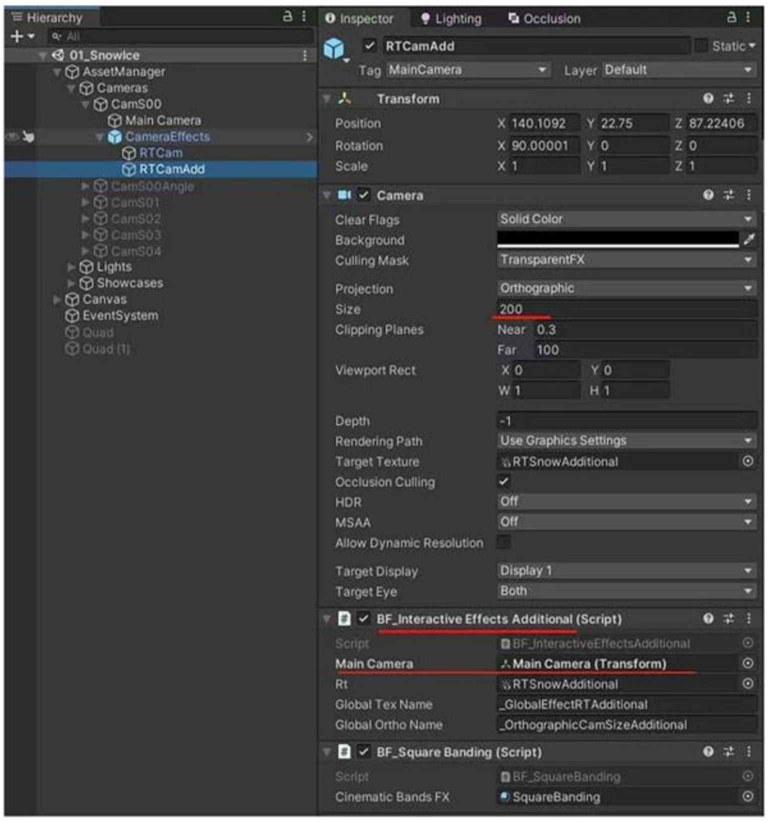
Assets > BruteForce > RenderTextures

RTSnow
RTSnowAdditional

Assets/BruteForce/RenderTextures/RTSnow.renderTexture

Using the RTCamAdd to expand the Snow effect area

In addition to the RTCam I created an experimental additive RenderTexture effect which is much larger. This camera is optional but its use is highly recommended if you intend to use the effects on a large area.



Hierarchy

- 01_SnowIce
  - AssetManager
    - Cameras
      - CamS00
        - Main Camera
        - CameraEffects
          - RTCam
          - RTCamAdd
      - CamS00Angle
      - CamS01
      - CamS02
      - CamS03
      - CamS04
    - Lights
    - Showcases
  - Canvas
  - EventSystem
  - Quad
  - Quad (1)

Inspector

RTCamAdd — Tag: MainCamera — Layer: Default

Transform
Position: X 140.1092, Y 22.75, Z 87.22406
Rotation: X 90.00001, Y 0, Z 0
Scale: X 1, Y 1, Z 1

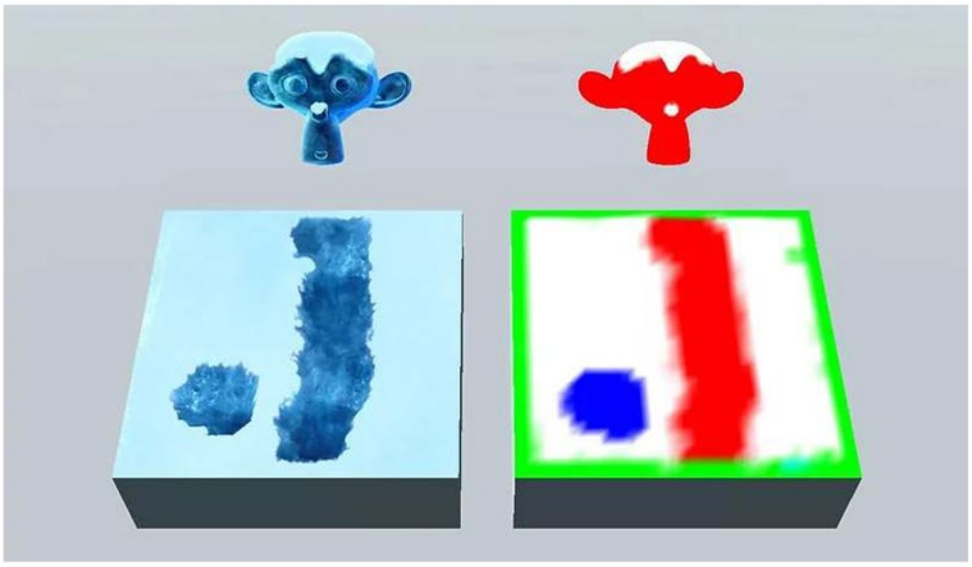
Camera
Clear Flags: Solid Color
Background
Culling Mask: TransparentFX
Projection: Orthographic
Size: 200
Clipping Planes: Near 0.3, Far 100
Viewport Rect: X 0, Y 0, W 1, H 1
Depth: -1
Rendering Path: Use Graphics Settings
Target Texture: RTSnowAdditional
Occlusion Culling: ✓
HDR: Off
MSAA: Off
Allow Dynamic Resolution
Target Display: Display 1
Target Eye: Both

BF_Interactive Effects Additional (Script)
Script: BF_InteractiveEffectsAdditional
Main Camera: Main Camera (Transform)
Rt: RTSnowAdditional
Global Tex Name: _GlobalEffectRTAdditional
Global Ortho Name: _OrthographicCamSizeAdditional

BF_Square Banding (Script)
Script: BF_SquareBanding
Cinematic Bands FX: SquareBanding

When you assign the main camera to the "Main Camera" field in the **BF_InteractiveEffectsAdditional** script; the RTcamAdd will automatically change its position based on the main camera FOV.

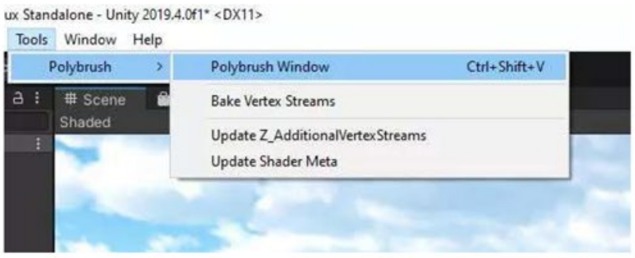
5. Paint Snow & Ice



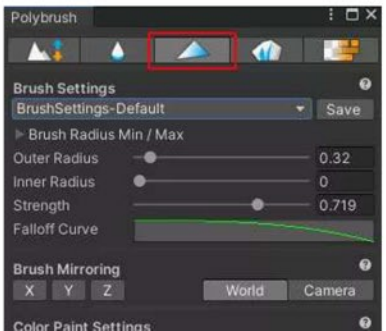
Using the Brute Force - Snow & Ice shader you can paint the vertex color of your assets to dynamically draw snow or ice.
The default value and state of the shader is snow.
With that in mind let's see in details what each vertex color channel do:
- -**White**: Renders Snow
- -**Red**: Renders Ice
- -**Green**: Renders Snow without displacement
- -**Blue**: Renders Ice without interactive effects

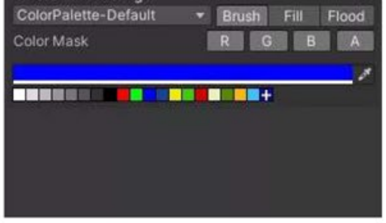
How to paint Vertex Color on your mesh

The asset comes with a Polybrush dependency (see documentation) which is an official Unity assets that let's you draw colors on vertex. In truth the snow & ice shader doesn't require this asset to work but it does greatly improve workflow.
You can also use your favorite 3D software to draw vertex color and it will work fine.
For the rest of this page I will assume you are using the Polybrush asset.


ux Standalone - Unity 2019.4.0f1* <DX11>
Tools
Window
Help
Polybrush
Polybrush Window Ctrl+Shift+V
Bake Vertex Streams
Update Z_AdditionalVertexStreams
Update Shader Meta
Scene
Shaded

Open the Polybrush Window by going to: **Tools** > **Polybrush** > **Polybrush Window**
You will have this window opened:


Polybrush
Brush Settings
BrushSettings-Default
Save
Brush Radius Min / Max
Outer Radius 0.32
Inner Radius 0
Strength 0.719
Falloff Curve
Brush Mirroring
X Y Z
World Camera
Color Paint Settings



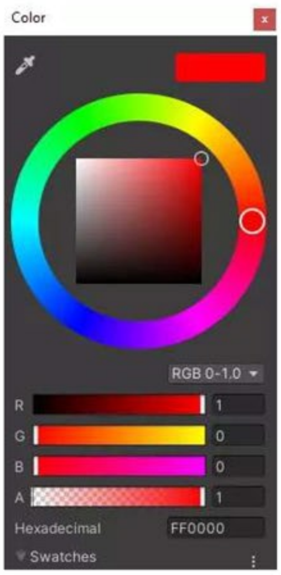
The tab that interests us is the "**Paint vertex colors on meshes**" click on that.

You can see there is a "Outer Radius" value that controls the radius of the brush size and a Color Paint Settings at the bottom.

You can choose which color you want to paint by clicking the big color field below color mask.

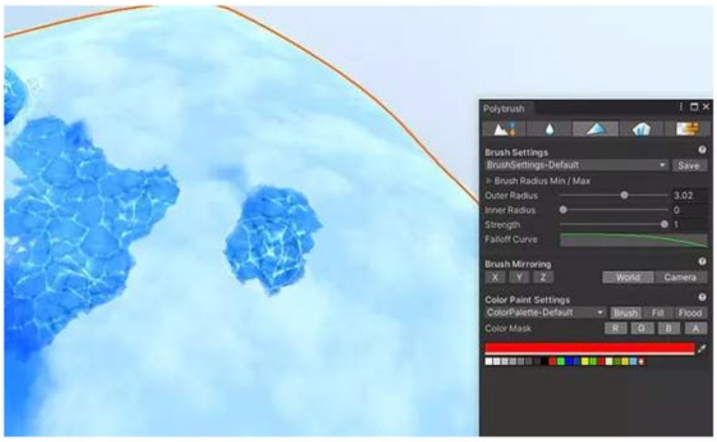
When choosing between White, Red, Green or Blue colors make sure it is the absolute color; meaning white is fully white (1,1,1) and red is fully red (1,0,0). Again, tead the polybrush documentation for more details.

Select red color:

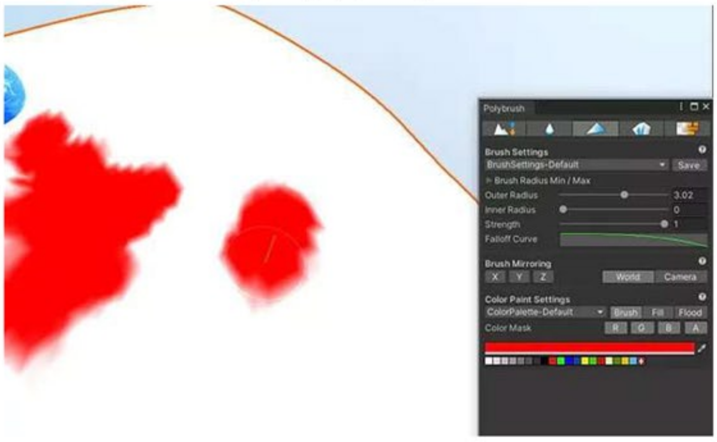


R 1
G 0
B 0
A 1
Hexadecimal FF0000

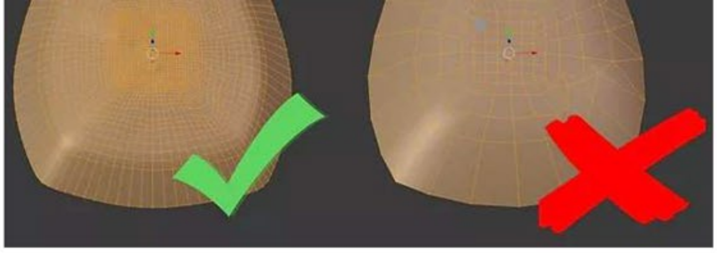
Now click on the mesh you want to paint on with the snow & ice material already placed, you can start painting ice:



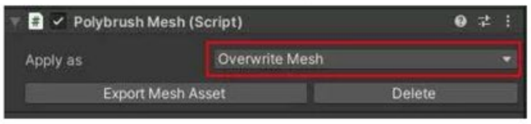
This is what you are actually doing by painting:



Keep in mind you need a certain **number of vertices** to have a higher effect definition, here's a good guideline:



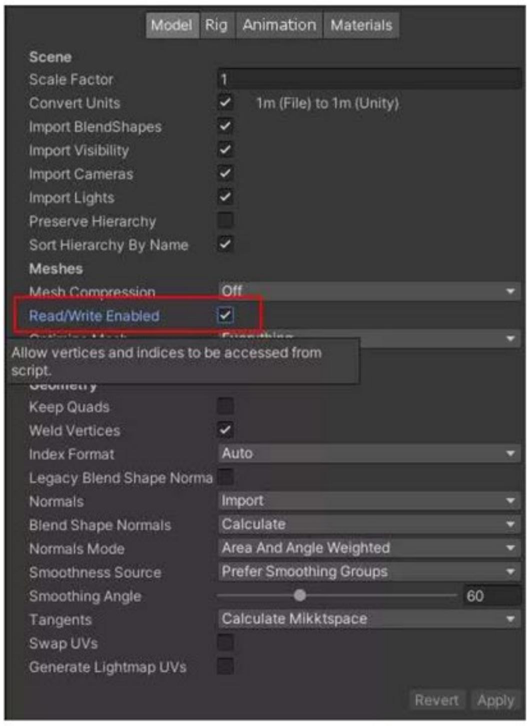
Finally when you are happy with your mesh make sure the "Apply As" is set to **Overwrite Mesh**



Polybrush Mesh (Script)

Apply as: Overwrite Mesh

When importing a custom mesh inside Unity and you want to use the PlayerSnowball featured in the demo scene **you are required to enable Read/write** in the import settings of said mesh.

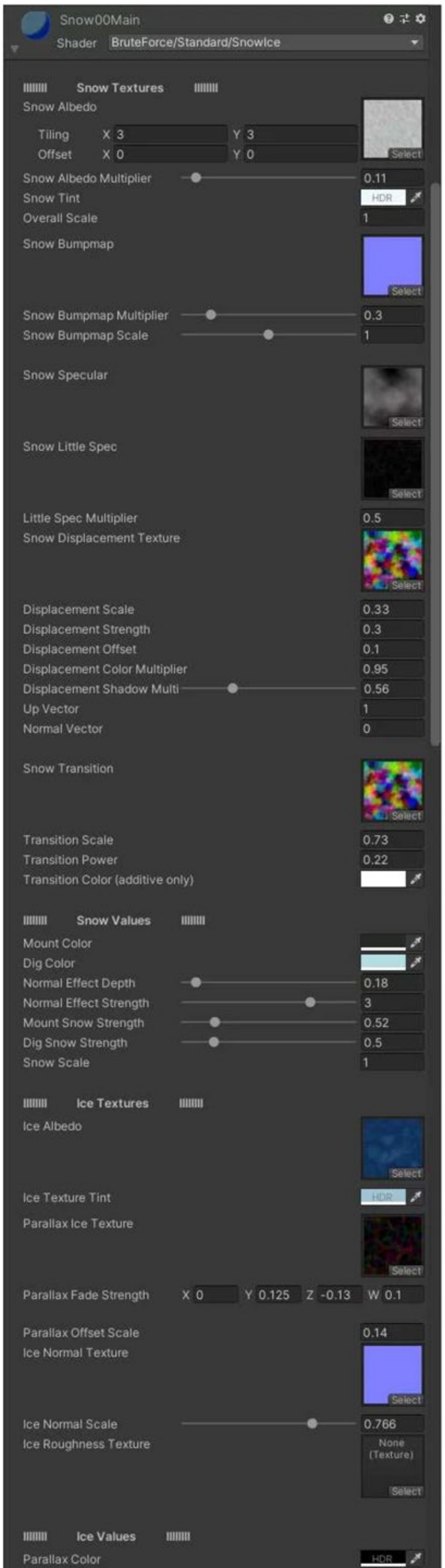


Scene
Scale Factor: 1
Convert Units: ✓ 1m (File) to 1m (Unity)
Import BlendShapes: ✓
Import Visibility: ✓
Import Cameras: ✓
Import Lights: ✓
Preserve Hierarchy:
Sort Hierarchy By Name: ✓
Meshes
Mesh Compression: Off
Read/Write Enabled: ✓

Allow vertices and indices to be accessed from script.

Keep Quads:
Weld Vertices: ✓
Index Format: Auto
Legacy Blend Shape Norma:
Normals: Import
Blend Shape Normals: Calculate
Normals Mode: Area And Angle Weighted
Smoothness Source: Prefer Smoothing Groups
Smoothing Angle: 60
Tangents: Calculate Mikktspace
Swap UVs:
Generate Lightmap UVs:

6. Snow & Ice Properties



Snow00Main

Shader: BruteForce/Standard/SnowIce

Snow Textures

Snow Albedo	Select
Tiling	X 3, Y 3
Offset	X 0, Y 0
Snow Albedo Multiplier	0.11
Snow Tint	HDR
Overall Scale	1
Snow Bumpmap	Select
Snow Bumpmap Multiplier	0.3
Snow Bumpmap Scale	1
Snow Specular	Select
Snow Little Spec	Select
Little Spec Multiplier	0.5
Snow Displacement Texture	Select
Displacement Scale	0.33
Displacement Strength	0.3
Displacement Offset	0.1
Displacement Color Multiplier	0.95
Displacement Shadow Multi	0.56
Up Vector	1
Normal Vector	0
Snow Transition	Select
Transition Scale	0.73
Transition Power	0.22
Transition Color (additive only)	

Snow Values

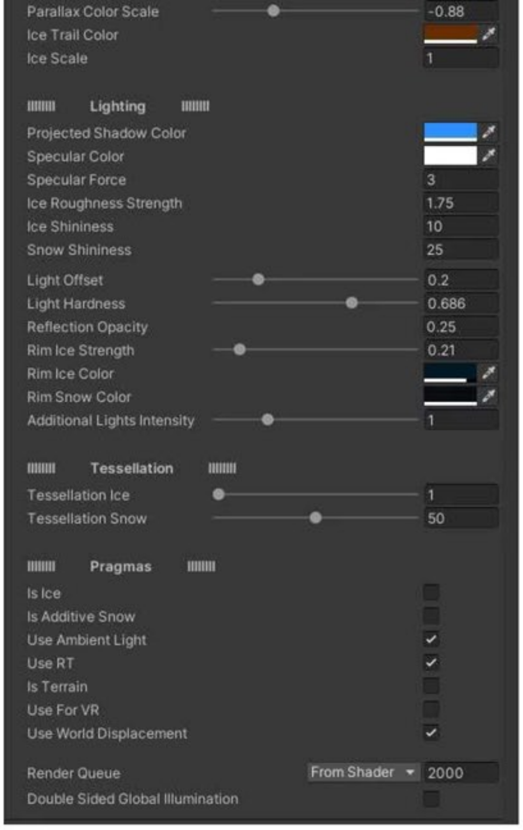
Mount Color	
Dig Color	
Normal Effect Depth	0.18
Normal Effect Strength	3
Mount Snow Strength	0.52
Dig Snow Strength	0.5
Snow Scale	1

Ice Textures

Ice Albedo	Select
Ice Texture Tint	HDR
Parallax Ice Texture	Select
Parallax Fade Strength	X 0, Y 0.125, Z -0.13, W 0.1
Parallax Offset Scale	0.14
Ice Normal Texture	Select
Ice Normal Scale	0.766
Ice Roughness Texture	None (Texture) Select

Ice Values

Parallax Color	HDR



Parallax Color Scale -0.88
Ice Trail Color 🟫
Ice Scale 1

IIIIIII Lighting IIIIIII
Projected Shadow Color 🟦
Specular Color ⬜
Specular Force 3
Ice Roughness Strength 1.75
Ice Shininess 10
Snow Shininess 25
Light Offset 0.2
Light Hardness 0.686
Reflection Opacity 0.25
Rim Ice Strength 0.21
Rim Ice Color ⬛
Rim Snow Color ⬛
Additional Lights Intensity 1

IIIIIII Tessellation IIIIIII
Tessellation Ice 1
Tessellation Snow 50

IIIIIII Pragmas IIIIIII
Is Ice ☐
Is Additive Snow ☐
Use Ambient Light ✓
Use RT ✓
Is Terrain ☐
Use For VR ☐
Use World Displacement ✓

Render Queue From Shader 2000
Double Sided Global Illumination ☐

The Snow asset is controlled entirely through the material property block, I suggest toying with the values to familiarize yourself with them. The material being very long I categorized the fields into:
Snow Textures || Snow Values || Ice Textures || Ice Values || Lighting || Tessellation || Pragmas

I will describe the important parameters in order:

|| Snow Textures ||
- **Snow Albedo**: A colored texture to control the snow color, use this field to add some organic noise
- **Overall Scale**: The scale parameter controlling every texture coordinates except the snow displacement
- **Snow Little Spec**: The B&W texture controlling the specular intensity where the light reflects on the snow surface
- **Snow Displacement Texture**: The texture controlling the displacement of the snow vertex
- **Displacement Offset**: The value differentiating the height between snow and ice, 0 = no differences
- **Displacement Color Multiplier**: The parameter multiplying the value (color) difference based on displacement height
- **Displacement Shadow Multiplier**: The parameter multiplying the shadow value based on displacement height
- **Up Vector**: The higher this value is the more the displacement will move towards the world upward vector
- **Normal Vector**: The higher this value is the more the displacement will move towards the vertex normal
- **Snow Transition**: The texture controlling the transition shape between snow and ice

|| Snow Values ||
- **Mount Color**: The color tint of the pulled up snow based on interactive effect (green)
- **Dig Color**: The color tint of the dug up snow based on interactive effect (blue)
- **Normal Effect Depth**: The multiplier offset of the interactive effect, does not scale with the RenderTexture dimension if you do not understand what it does leave it between 0.10-0.20
- **Normal Effect Strength**: The value multiplying the overall interactive effect
- **Mount/Dig Snow Strength**: The vertex displacement multiplier of the respective effect (Vertex position only)
- **Snow Scale:** The scale parameter controlling the snow texture coordinates

|| Ice Textures ||
- **Parallax Ice Texture:** The RGB texture controlling the parallax mapping of the ice; Red = 1st layer / Green = 2nd layer / Blue = 3rd layer
- **Parallax Fade Strength:** The separate intensity values of each parallax layers;
  X = The overall intensity value of all the layers (leave it to 0 if possible, goes from -1 to 1)
  Y = The 1st layer intensity
  Z = The 2nd layer intensity
  W = The 3rd layer intensity
  Every values can go beyond -1 and +1 but I don't recommend doing so, keep those values fairly low
- **Parallax Offset Scale:** The depth multiplier of the parallax effect, the higher it is the deeper the effect looks

|| Ice Values ||
- **Parallax Color:** The tint color of the ice based on parallax effect, it represents the deepest color of the parallax
- **Parallax Color Scale:** The general color multiplier of the parallax color based on parallax depth
- **Ice Trail Color:** The tint of the interactive effect on the ice (color is reversed; white = very dark tint trail)
- **Ice Scale:** The scale parameter controlling the ice texture coordinates

|| Lighting ||
- **Projected Shadow Color:** The color of the snow & ice shadows

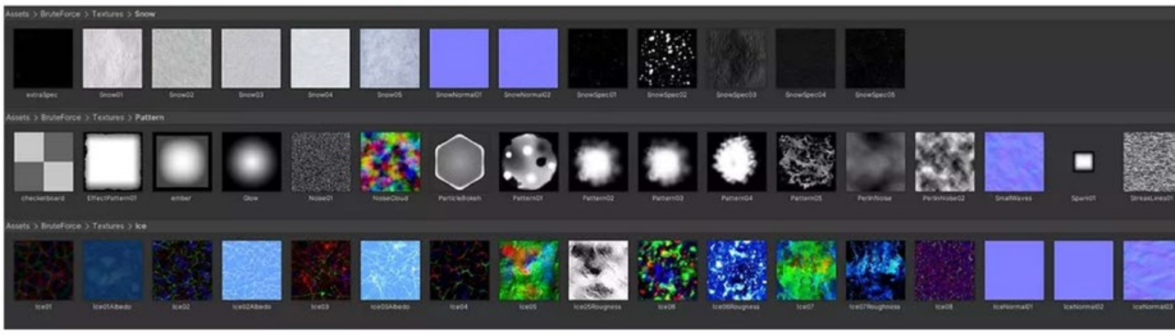
- **Reflection Opacity:** The opacity of the reflected environment cubemap onto the ice
- **Rim Ice Color:** The tint color of the ice rim light (alpha controls the scale)
- **Additional Lights Intensity:** The additional lights (points, spots) intensity multiplier, only works with additional lights

|| Tessellation ||
- **Tessellation Ice:** The tessellation control value of the ice, 1 = No Tessellation | 100 = maximum Tessellation, in the majority of uses you SHOULD leave it to 1
- **Tessellation Snow:** The tessellation control value of the snow, a good optimized value should be between 20 and 50

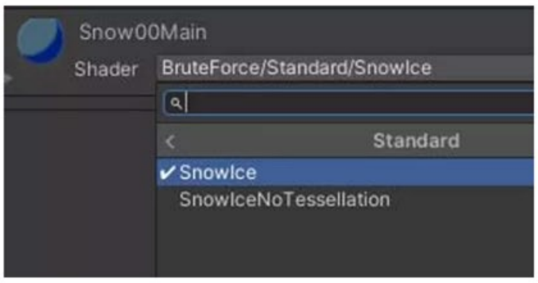
|| Pragmas ||
- **Is Ice:** If enabled it will turn the whole mesh into ice independently from vertex color
- **Is Additive Snow:** If enabled it will activate the additive snow feature culling the ice, do not enable this feature by itself instead you should use the Additive Snow component, more details here
- **Use RT:** If enabled it will activate the interactive feature, disable it if you don't intend to use the interactive part
- **Is Terrain:** If enabled it will activate the Terrain feature, do not enable this feature by itself instead you should use the Snow Terrain component, more details here
- **Use World Displacement:** The If enabled it will project the snow displacement texture in world coordinates independent from mesh uv, turn it off if you want to use your mesh uvs for snow displacement

You can switch textures around or try different materials from the large collection in the package:

You can create your own textures for any texture slots, if you want to make your own Parallax Ice texture keep in mind that the color channels should be treated separately.



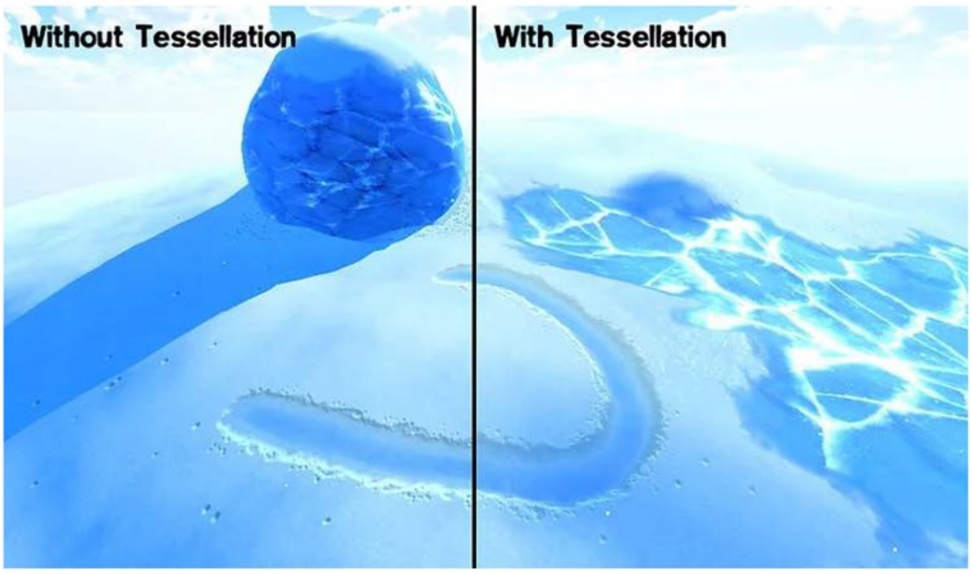
If you want to create your own materials here is the list of the Snow & Ice shaders:



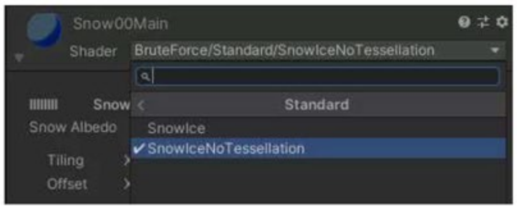
SnowIce: The standard Snow & Ice shader

SnowIceNoTessellation: The same as the above but without the vertex tessellation feature, use this variant to build for WebGl or low-end mobile; see more about WebGl and Mobile Build

7. WebGl and Mobile Build

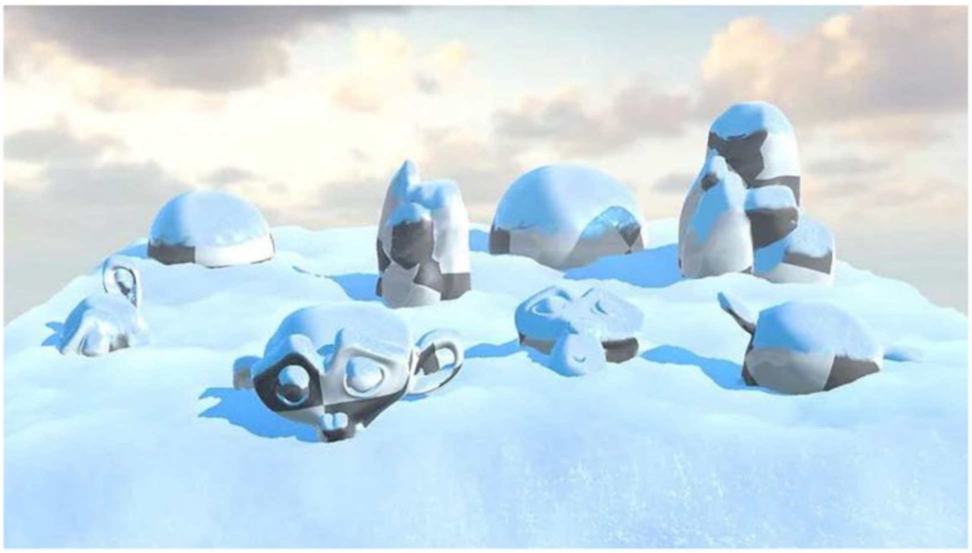


If you want to use the Snow & Ice shader for your WebGl or low-end mobile projects the asset features a tessellation-less variant of the shader that is compatible with low render target APIs. Simply switch any snow materials to the variant:

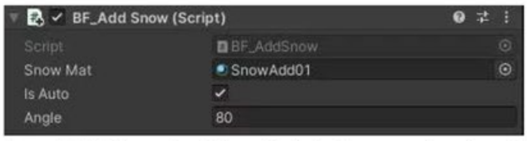

Snow00Main
Shader BruteForce/Standard/SnowIceNoTessellation

As you can see in the image above you will not lose much fidelity and you will gain compatibility and performance.

8. Additive Snow Feature



The additive Snow feature is an experimental component that you can attach to any meshes to add procedural snow to them:

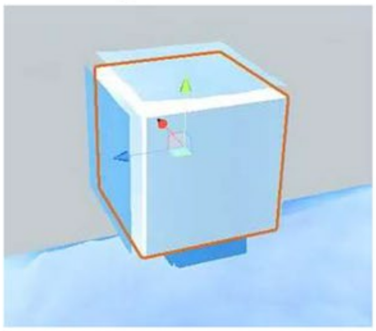


BF_Add Snow (Script)

Script	BF_AddSnow
Snow Mat	SnowAdd01
Is Auto	✓
Angle	80

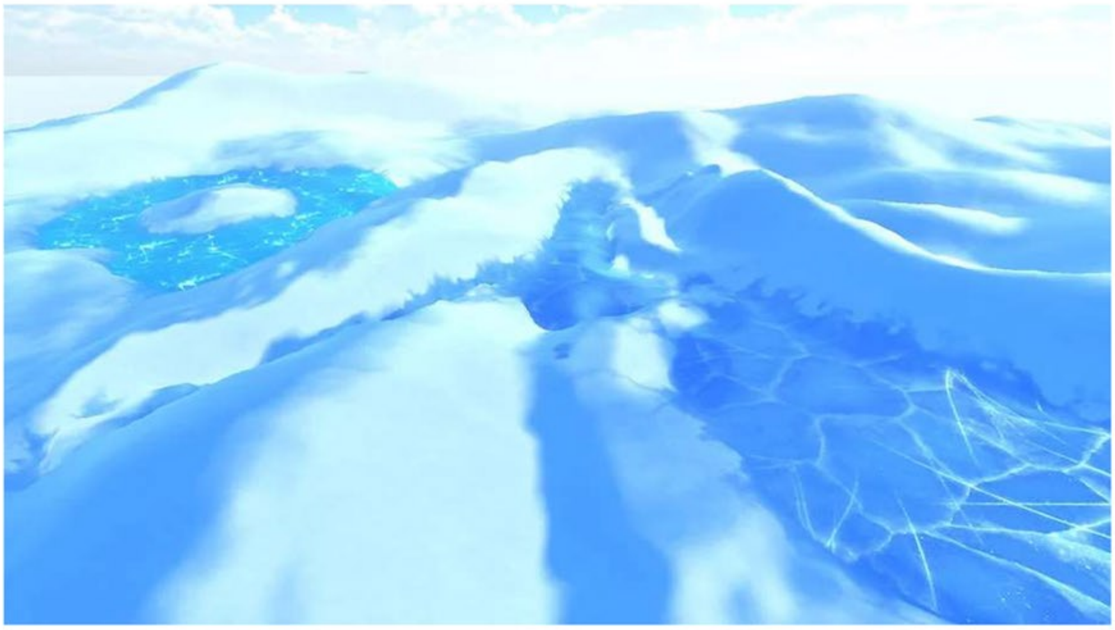
Snow Mat: The material to attach to the procedural mesh
IsAuto: Is the area covered by snow calculated procedurally based on original face normal, if this is set to false the calculation will take the original mesh vertex color to render the snow area so make sure to have painted you mesh before disabling this option
Angle: The value controlling the procedural calculation of snow area, if **IsAuto** is disabled you can ignore this value. The higher the value the more faces that look in another direction than upward will have snow

This feature is still in experimental phase and suffers some limitation, one that is very noticeable is the incompatibility with hard-edged meshes:

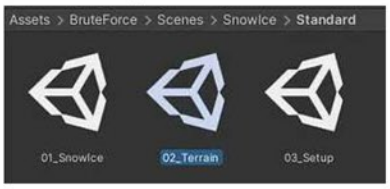


If your custom mesh has a flat shading like this primitive cube then it won't be able to make a coherent snow coverage, you should only use this feature on completely smooth shading like a sphere or a smoothed out rock. An upgraded version of this feature will be released in the coming weeks ignoring original mesh shading.

9. Terrain Feature

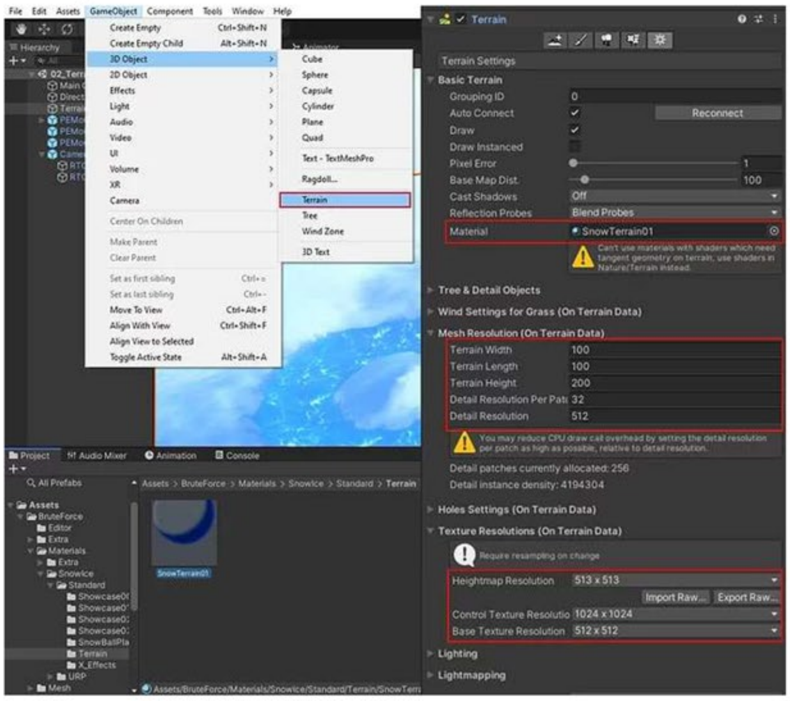


There is a template scene for terrain environment in the asset package (both built-in and URP) located here:



If you want to create your own terrain and implement the snow shader here's the important steps to follow:

1. Setup the terrain



Terrain

Terrain Settings

Basic Terrain
- Grouping ID: 0
- Auto Connect: ✓
- Draw: ✓
- Draw Instanced
- Pixel Error: 1
- Base Map Dist.: 100
- Cast Shadows: Off
- Reflection Probes: Blend Probes
- Material: SnowTerrain01

Can't use materials with shaders which need tangent geometry on terrain, use shaders in Nature/Terrain instead.

Mesh Resolution (On Terrain Data)
- Terrain Width: 100
- Terrain Length: 100
- Terrain Height: 200
- Detail Resolution Per Patch: 32
- Detail Resolution: 512

You may reduce CPU draw call overhead by setting the detail resolution per patch as high as possible, relative to detail resolution.

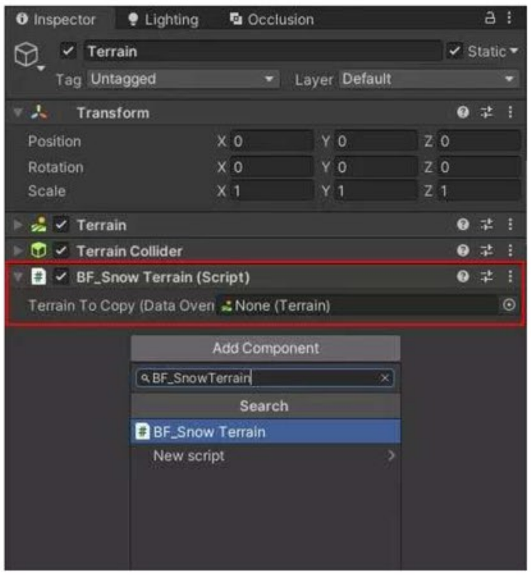
Detail patches currently allocated: 256
Detail instance density: 4194304

Texture Resolutions (On Terrain Data)

Require resampling on change
- Heightmap Resolution: 513 x 513
- Control Texture Resolution: 1024 x 1024
- Base Texture Resolution: 512 x 512

Create a new terrain with a 100x100 dimension (works best out of the box rather than the HUGE 1000x1000 default terrain)

Then add the "SnowTerrain01" material or its URP equivalent or your own Snow & Ice material



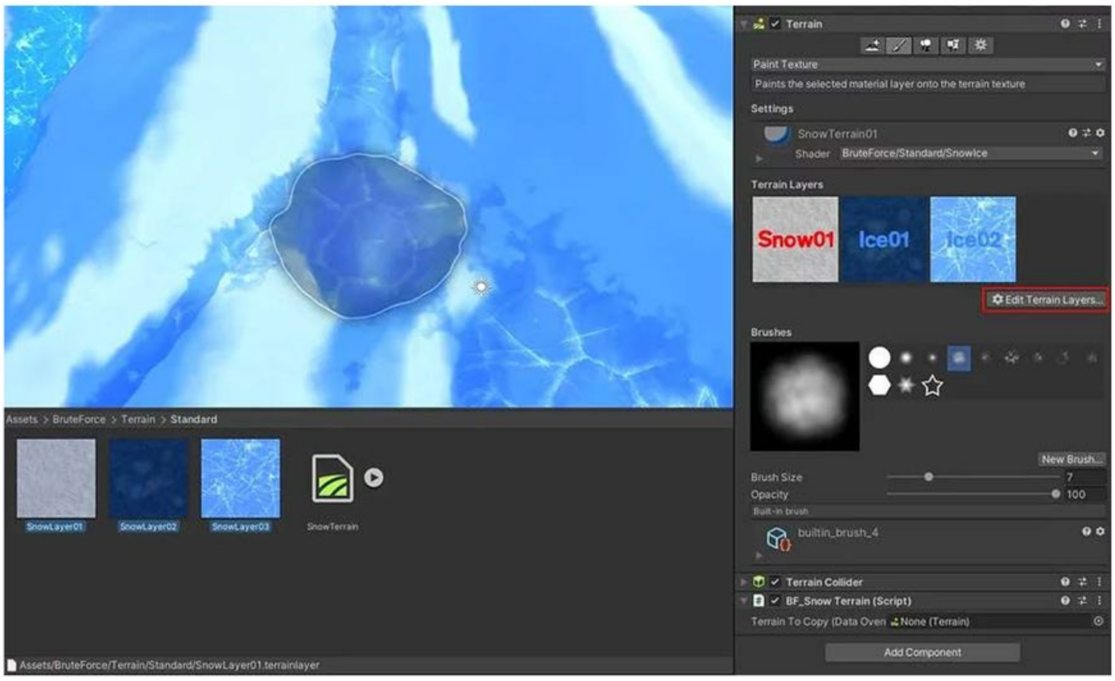
Terrain ✓ Static
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Terrain
Terrain Collider
BF_Snow Terrain (Script)
Terrain To Copy (Data Overr None (Terrain)
BF_SnowTerrain
BF_Snow Terrain

IMPORTANT: You need to add the "**BF_SnowTerrain**" script to your terrain for it to work properly and if you're only using the terrain feature of the grass shader without regards for other terrain assets compatibility you are not required to add a "terrainToCopy" parameter nor should you press the "Sync terrain Data" button

VERY IMPORTANT: Be extra careful with these fields, if you assign or press the sync terrain button on a terrain you do not want to change its previous height map it will override and change it, you can press the revert button to set the values back to how they were.

2. Edit terrain layers

After you've set the right shader/material to your terrain get over the paint terrain tool in the terrain inspector:

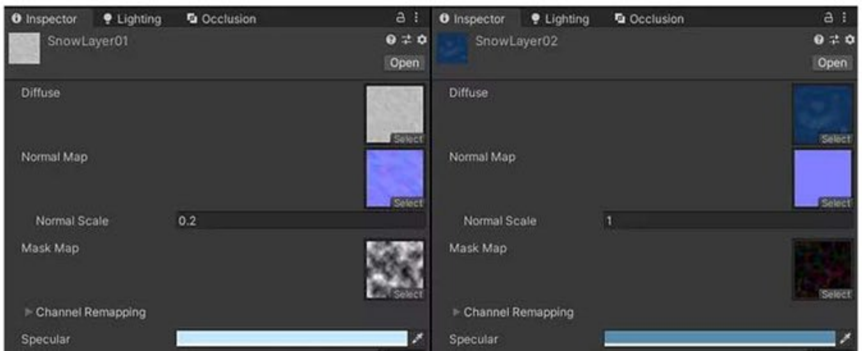


Paint Texture
Paints the selected material layer onto the terrain texture
SnowTerrain01
Shader BruteForce/Standard/SnowIce
Brush Size 7
Opacity 100
builtin_brush_4
BF_Snow Terrain (Script)
Terrain To Copy (Data Overr None (Terrain)
Assets > BruteForce > Terrain > Standard
Assets/BruteForce/Terrain/Standard/SnowLayer01.terrainlayer

To draw snow on the terrain use the **SnowLayer01**, the position in the terrain layers arrangement is very important, the first one SHOULD ALWAYS be the snow layer.

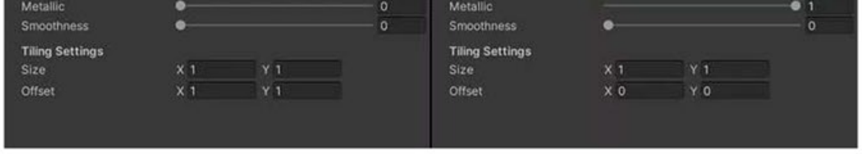
To draw ice on the terrain use the other **SnowLayer** (02,03), if you don't want/need other snow/ice layers you can delete them via the "Edit Terrain Layers" button (Currently limited to 4 layers).

If you want to edit the layers to have a new snow/ice layer you should head over to the corresponding "**NewLayerXX**" in the asset project:



SnowLayer01
Diffuse
Normal Map
Normal Scale 0.2
Mask Map
Specular

SnowLayer02
Diffuse
Normal Map
Normal Scale 1
Mask Map
Specular


Metallic 0
Smoothness 0
Tiling Settings
Size X 1 Y 1
Offset X 1 Y 1

Metallic 1
Smoothness 0
Tiling Settings
Size X 1 Y 1
Offset X 0 Y 0

The first layer which should ALWAYS be a snow layer;

If you want to have a Snow Layer:
Diffuse is the Albedo Texture,
Normal Map is the bump map,
Normal Scale is the normal multiplier,
Mask Map is the displacement map,
Specular is the overall color of the snow,
Metalic should be set to **0**,
Size.XY is the Tiling size

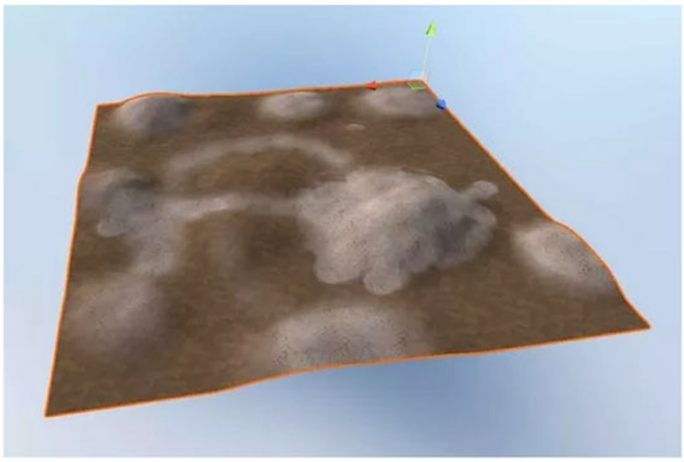
If you want to have an Ice Layer:
Diffuse is the Albedo Texture,
Normal Map is the bump map,
Normal Scale is the normal multiplier,
Mask Map is the parallax ice texture,
Specular is the overall color of the ice,
Metalic should be set to **1**,
Size.XY is the Tiling size

The only difference between a Snow Layer and an Ice Layer are Mask Map and Metallic slider.

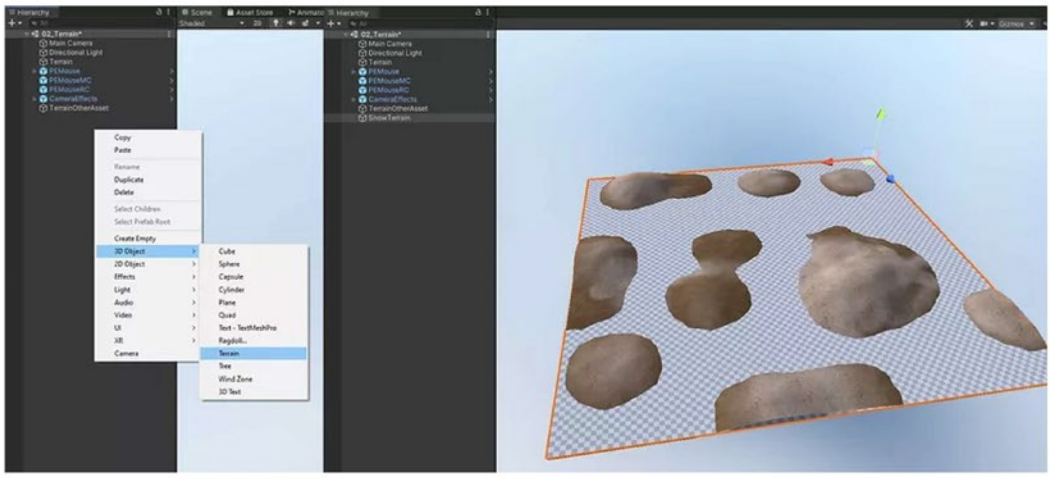
STOP there if you don't use any other terrain assets and are not looking for cross-compatibility!

3. Terrain Asset compatibility (optional)

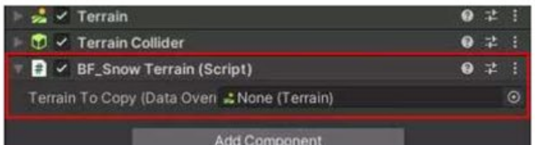
I implemented a solution for users to use my snow shader with any other terrain assets out there.
Here is what you need to do in order to get it working:



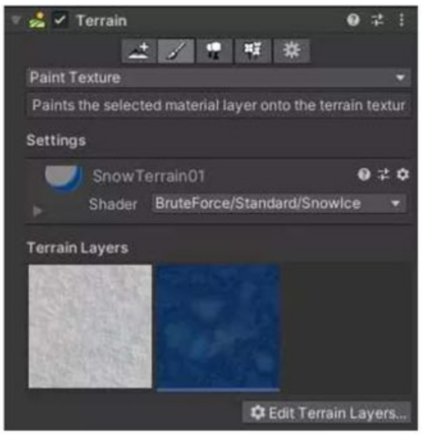
Let's assume you have a working terrain working with another terrain asset (it must be using the built-in terrain as a base)



Setup the snow terrain just like described in the steps before, add the snow terrain material and the **BF_SnowTerrain**.cs script.

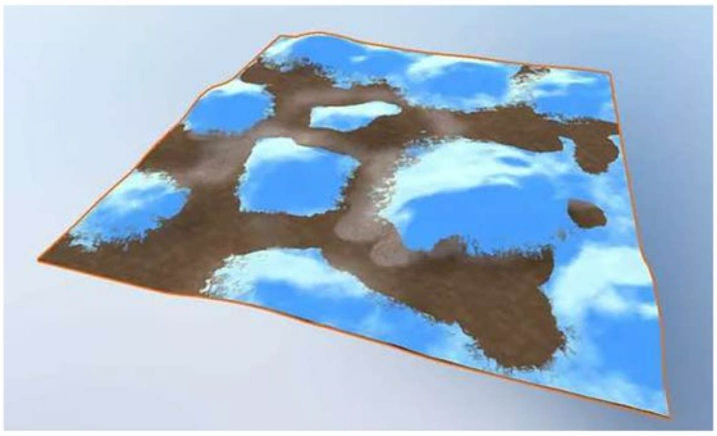

Terrain
Terrain Collider
BF_Snow Terrain (Script)
Terrain To Copy (Data Overi None (Terrain)
Add Component

In the **BF_SnowTerrain** component; drag and drop the other asset terrain in the "TerrainToCopy" field.
Then press the Sync Terrain Data button to sync the dimensions and height texture.
(I strongly recommend you to lower the tessellation count on the snow terrain material, you don't need as much precision for snow on a terrain)



Terrain
Paint Texture
Paints the selected material layer onto the terrain textur
Settings
SnowTerrain01
Shader BruteForce/Standard/SnowIce
Terrain Layers
Edit Terrain Layers...

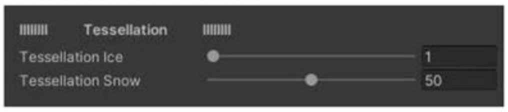
The first layer should be a snow layer with metallic set to 0 just like the regular setup. And the second layer has to be an ice layer with metallic set to 1, use this layer to remove snow of the first layer.



Now you should be able to draw snow with the snow layers and cull the snow to show the other terrain asset with the second layer of ice! You can add snow layers to this process for more snow variety.
Please keep in mind that these steps are only necessary when you want to use my snow shader terrain feature with other terrain assets, you do not need to do this if you only use my shaders.

10. Performance Tips

Tessellation Factor


Tessellation

Tessellation Ice	1
Tessellation Snow	50

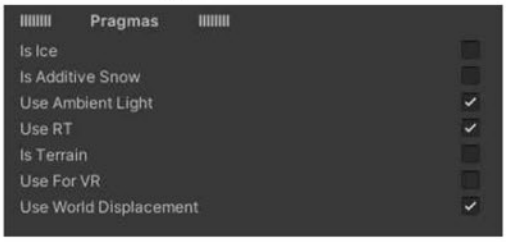
Probably the biggest performance factor of the snow & ice shader is the tessellation factor of the snow.

You should keep the **Tessellation Ice to 1** (which is equal to no tessellation)

For PC/Linux I recommend setting **Tessellation Snow between 20 and 50** and for mobile/VR/AR you should stay **below 20**.

The performance hit is relative to the view distance between the snow and the camera.

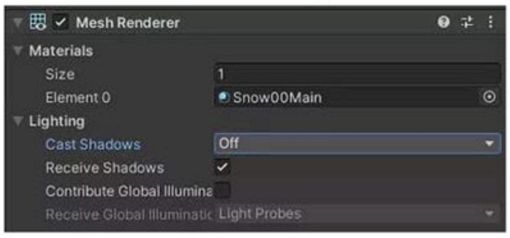
Material Pragmas


Pragmas

Is Ice	☐
Is Additive Snow	☐
Use Ambient Light	✓
Use RT	✓
Is Terrain	☐
Use For VR	☐
Use World Displacement	✓

Check or uncheck the features you don't need in the material, if you don't need Interactive effect you should disable "Use RT".

Disable Shadow Casting


✓ Mesh Renderer

Materials

Size	1
Element 0	Snow00Main

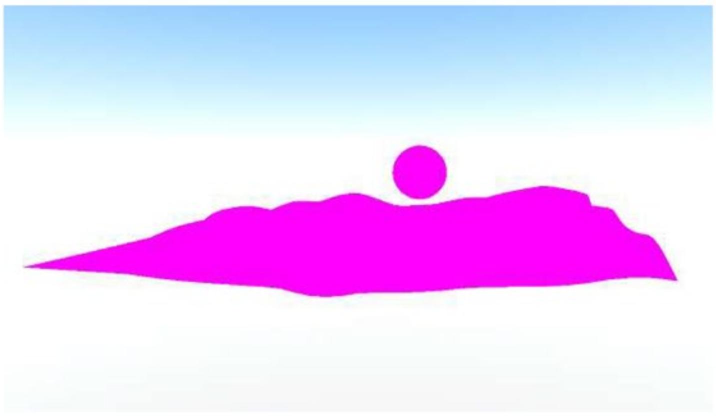
Lighting

Cast Shadows	Off
Receive Shadows	✓
Contribute Global Illumina	☐
Receive Global Illuminatic	Light Probes

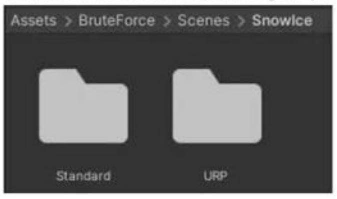
You might not need the shadow casting feature of the mesh renderer in which case this could save you a good performance chunk. This is especially true for terrains which have a Double shadow casting enabled by default!

11. FAQ & Troubleshooting

I opened the package scene and all I see is pink!



You opened the wrong scene, there is 2 variations of the Grass Shader: one for Unity Standard and one for URP.
Use the folders corresponding to your render pipeline:

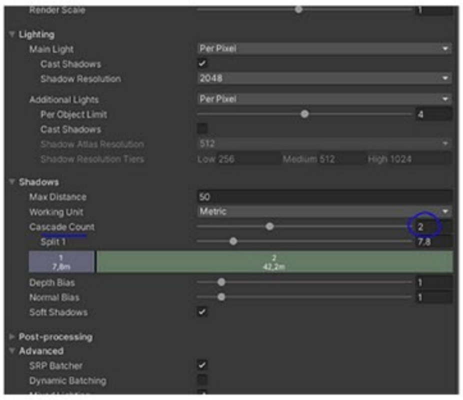


I can't find the URP folders and when I import the package it says only v1.0

Contact me at bruteforcegamesstudio@gmail.com with your unity asset store order and I will give it directly to you, chances are the unity package manager is forcing the wrong one on your unity version.
You can solve this issue without my help by installing any unity above 2019.4 and importing the package, then you can export the unity package in your projects.

I have shadow artefacts on Mobile URP

The problem comes from the URP pipeline itself, to get around this issue you need to set your shadow cascade count to 2 or more:

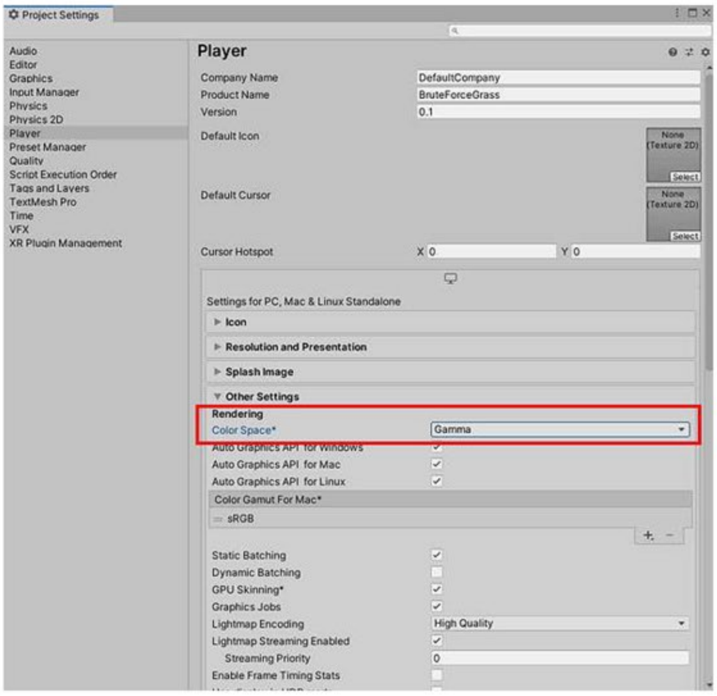


```
Render Scale                          1
▼ Lighting
  Main Light                  Per Pixel
    Cast Shadows              ✓
    Shadow Resolution         2048
  Additional Lights           Per Pixel
    Per Object Limit                  4
    Cast Shadows              ☐
    Shadow Atlas Resolution   512
    Shadow Resolution Tiers   Low 256   Medium 512   High 1024
▼ Shadows
  Max Distance                50
  Working Unit                Metric
  Cascade Count                       2
    Split 1                           7,8
  1 7,8m | 2 42,2m
  Depth Bias                          1
  Normal Bias                         1
  Soft Shadows                ✓
► Post-processing
▼ Advanced
  SRP Batcher                 ✓
  Dynamic Batching            ☐
```

The snow looks so pale/bright in my project

Color differences may occur when you're not using the default unity color space depending on your Unity version and Render Pipeline, (Gamme for Built-in and Linear for URP)

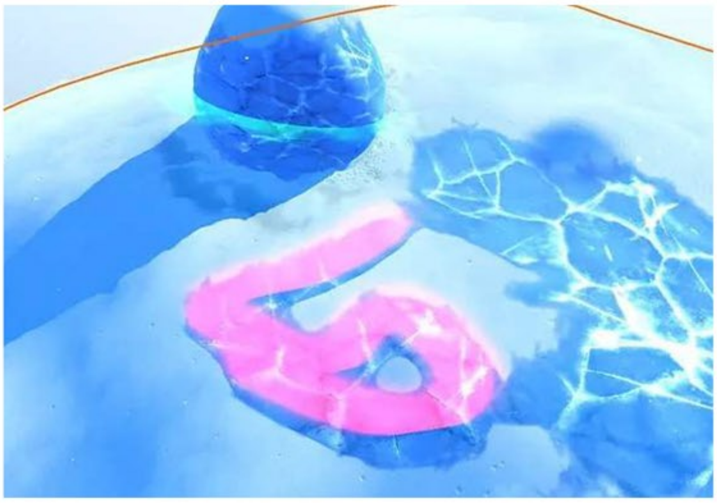
Either change the rendering color space or simply tune the materials to look the way you want



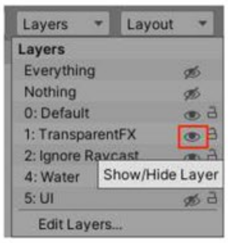
Company Name: DefaultCompany
Product Name: BruteForceGrass
Version: 0.1
Default Icon: None (Texture 2D)
Default Cursor: None (Texture 2D)
Cursor Hotspot: X 0 Y 0

Settings for PC, Mac & Linux Standalone
Icon
Resolution and Presentation
Splash Image
Other Settings
Rendering
Color Space*: Gamma
Auto Graphics API for Windows ✓
Auto Graphics API for Mac ✓
Auto Graphics API for Linux ✓
Color Gamut For Mac*: sRGB
Static Batching ✓
Dynamic Batching ☐
GPU Skinning* ✓
Graphics Jobs ✓
Lightmap Encoding: High Quality
Lightmap Streaming Enabled ✓
Streaming Priority: 0
Enable Frame Timing Stats ☐

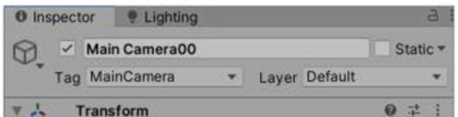
The Particle Effect is visible in the editor or at runtime

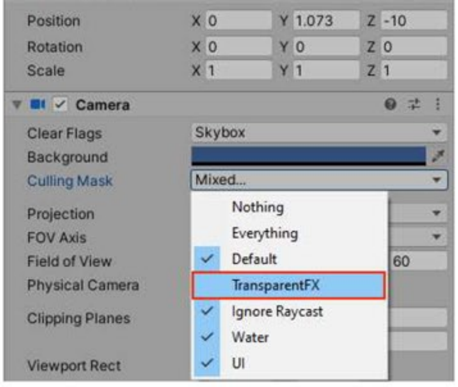


If you want to not see the particle effect rendering inside the editor, you have to disable the same layer it's in by using the layers visibility option, here the layer used is **TransparentFX**:



And if you need to disable it's visibility inside the game window or on build you need to do the same for the camera culling masks of your main camera:




Position X 0 Y 1.073 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Camera
Clear Flags Skybox
Background
Culling Mask Mixed...
Projection
FOV Axis
Field of View 60
Physical Camera
Clipping Planes
Viewport Rect