

# TECHNICAL DESIGN DOCUMENT

Procedural Winter Generation System

## Abstract

Procedurally generates Particle Systems, Snow Cover, and Icicles on a structure.  
Customizable, efficient, multi-purpose

Sarthak (Pixx) Sachdeva  
2023-2024

## Table of Contents

Glossary.....	1
Introduction .....	2
List of Features.....	2
High Level Diagrams (if any / delete as appropriate) .....	<b>Error! Bookmark not defined.</b>
Engine Requirements.....	2
User Setup Guide .....	2
Minimum System Requirements .....	4
UI of the Tool/Artefact/System .....	4
External Library Dependencies .....	5
Asset Pipeline.....	5
Material Pipeline.....	7
Assessment Related Requirements to be answered: .....	8
Horizontal Prototype.....	8
Vertical Prototype .....	8
Polished System .....	8
Test Plan.....	9
System User Manual .....	12
User Guide .....	13

## Glossary

*Abbreviation followed by the wording.*

BP – Blueprint

CPP – C++ Programming Language

UE – Unreal Engine

VFX – Visual Effects

NS – Niagara System

## Introduction

The proposed tool is a set of procedural generation systems that allows the user to quickly create a winter scene from within the editor without having to create any assets in an external 3D software.

## List of Features

The system must have the following features:

1. Stylized Snow Particle Effects
  - Control over the wind speed and direction.
  - Ability to change the radius and amount of snow.
  - Option to change the snow colour and texture.
2. Procedural snow cover generation
  - Control over snow thickness.
  - Options to change the snow's remeshing, and deformation.
  - Ability to select multiple actors.
  - Ability to change the snow material.
3. Procedural icicles generation
  - Variable icicle length, thickness, and shape randomness.
  - Control over clustering and distribution.
  - Options for different icicle types.
  - Ability to change the icicles material.

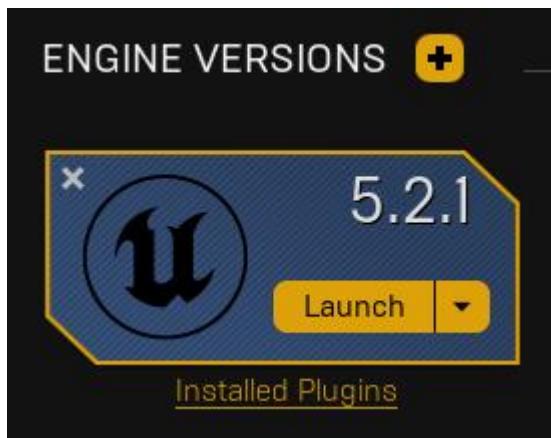
## Engine Requirements

Minimum Engine Version: Unreal Engine 5.0

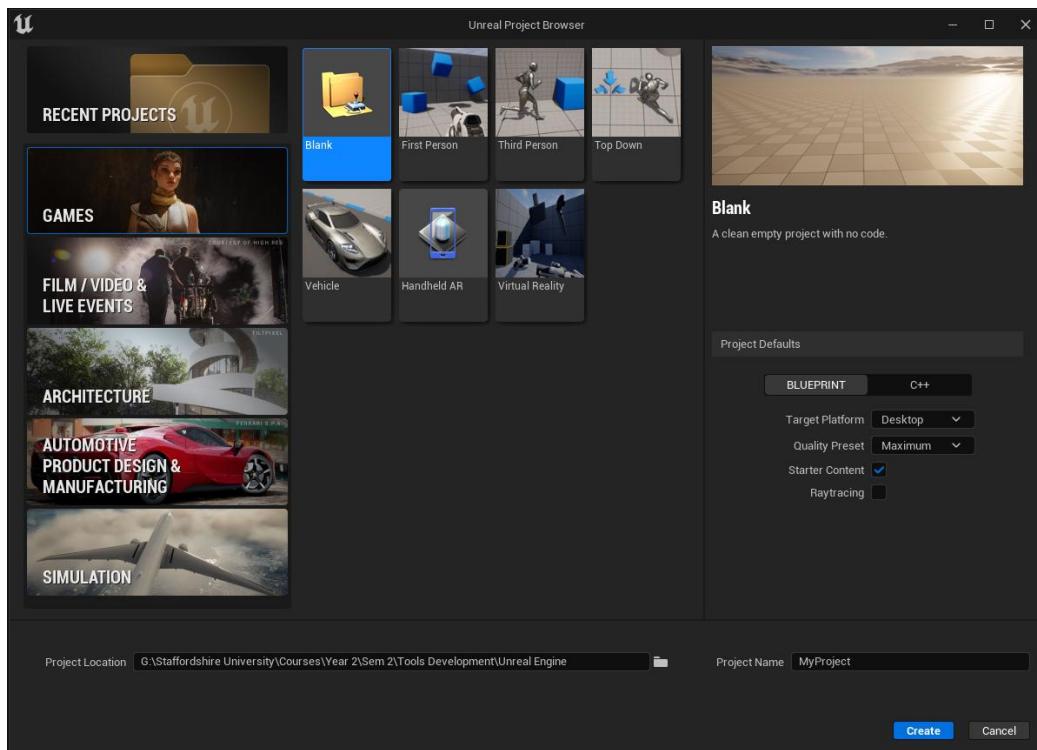
Recommended Engine Version: Unreal Engine 5.2.1

## User Setup Guide

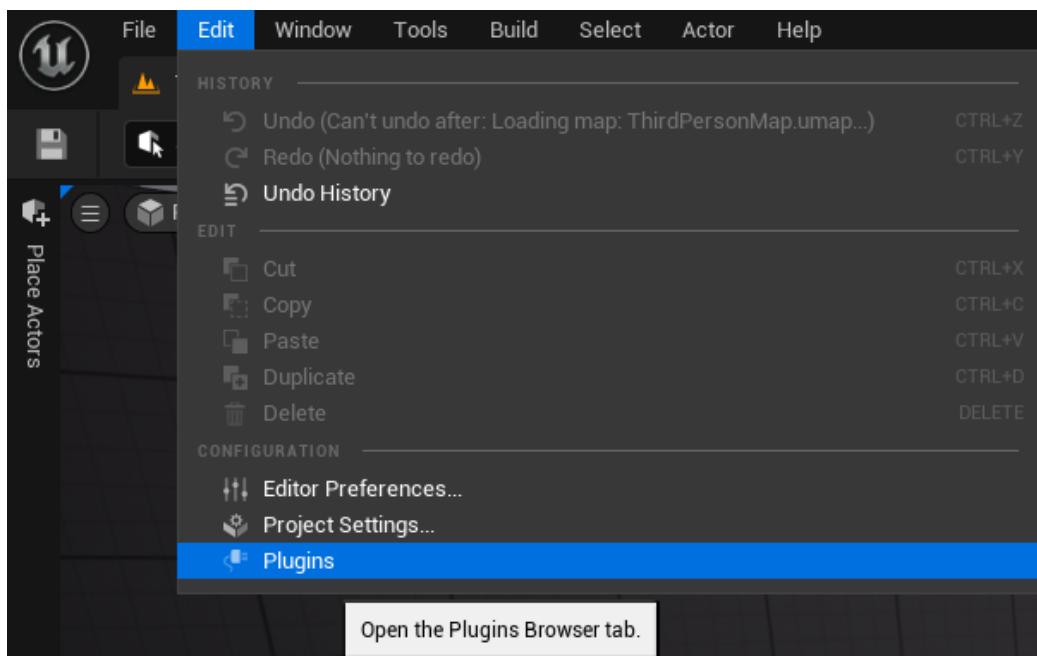
1. Launch your preferred engine version (Refer to the Engine Requirements section to avoid incompatibility issues).



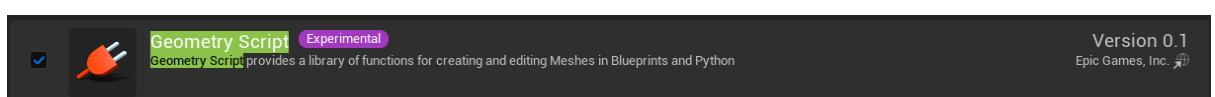
2. Create a new C++ project. You can use any template or create a blank project.



3. Once in the engine, then click on Edit > Plugins.



4. In the plugins browser tab, search for and enable the “Geometry Script” Plugin.



5. Restart the engine.

## Minimum System Requirements

- Operating system: Windows 10 64-bit.
- Processor: Quad-core Intel or AMD 2.5 GHz or superior.
- Memory: 8GB RAM.
- Graphics Card: Any DirectX 11 or 12 compatible card.

## UI of the Tool/Artefact/System

Snow VFX UI:

The screenshot shows a user interface for 'User Parameters' with the following settings:

Parameter	Value
Curl Noise Frequency	50.0
Curl Noise Strength	100.0
Snow Area Radius	500.0
Snow Intensity	300.0
Wind Direction	100.0
Wind Speed Scale	1.0

Snow Cover Generator UI:

The screenshot shows a user interface for 'Snowy Actors' and 'Snow Settings' with the following configurations:

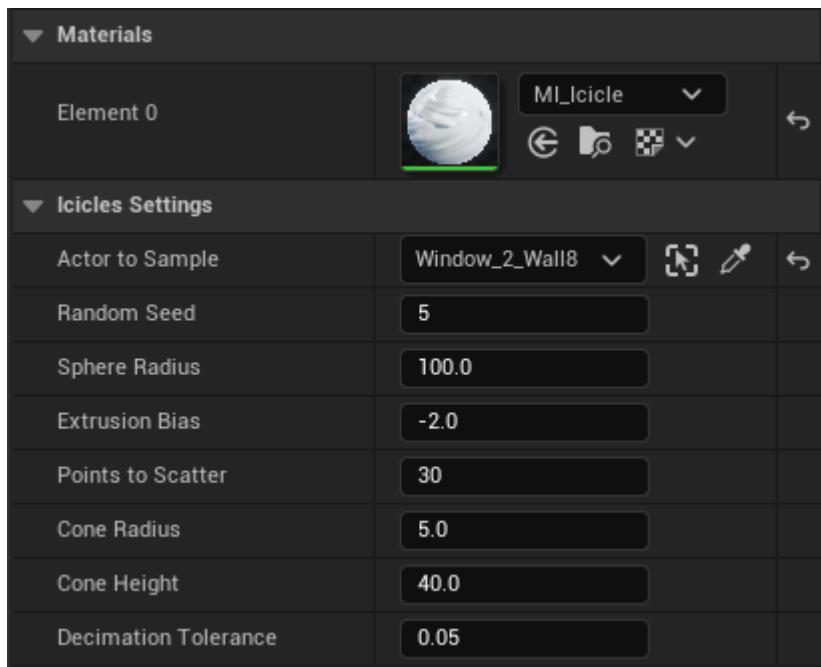
**Snowy Actors:**

Actor	Index [ 0 ]
SM_Cube	

**Snow Settings:**

Setting	Value
Snow Thickness	10.0
Perlin Noise Magnitude	50.0
Perlin Noise Frequency	0.2
Perlin Noise Seed	0
Remesh Type	Triangle Count
Remesh Iterations	0
Remesh Target Triangle Count	7500
Remesh Target Edge Length	6.0
Reject Actors Bigger Than	5000.0

Icicles Generator UI:



## External Library Dependencies

Geometry Script Plugin (Blueprints)

GeometryFramework (C++)

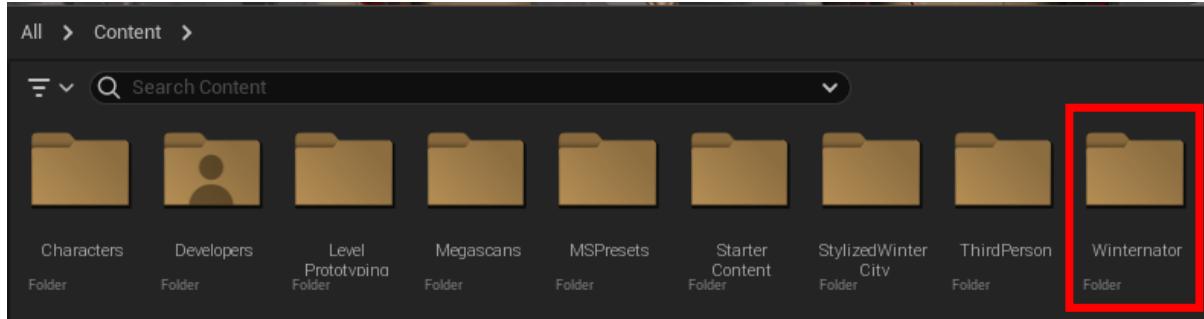
ProceduralMeshComponent (C++)

GeometryScriptingCore (C++)

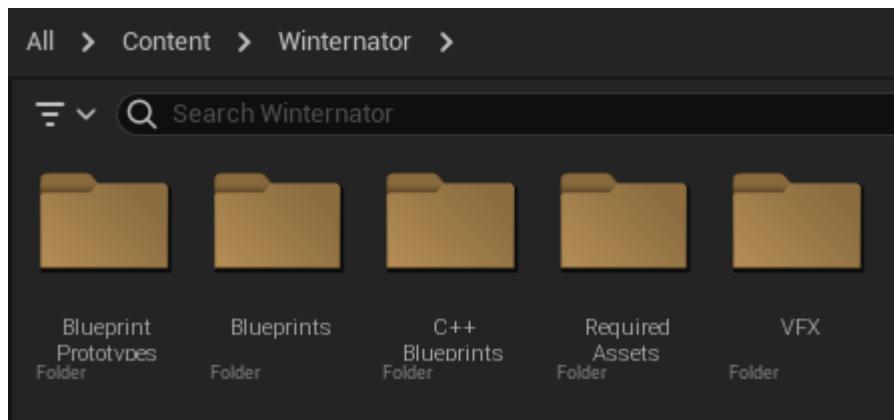
DynamicMesh (C++)

## Asset Pipeline

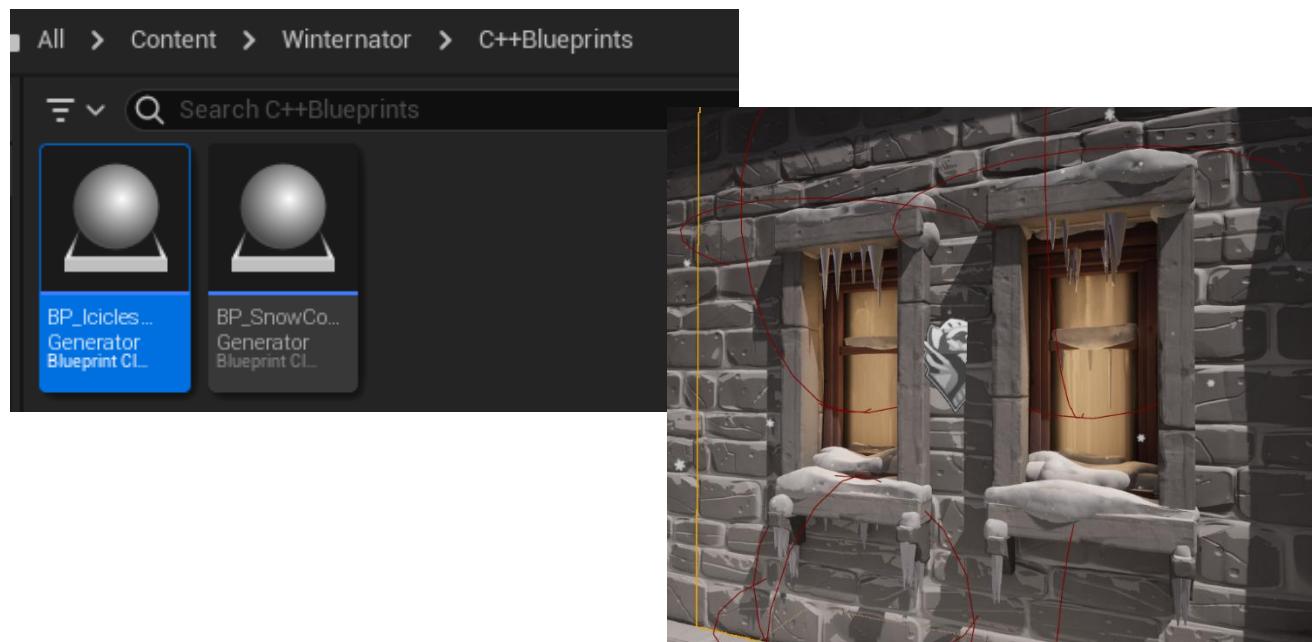
All of the assets included with the tool can be found within the Winternator folder under Content.



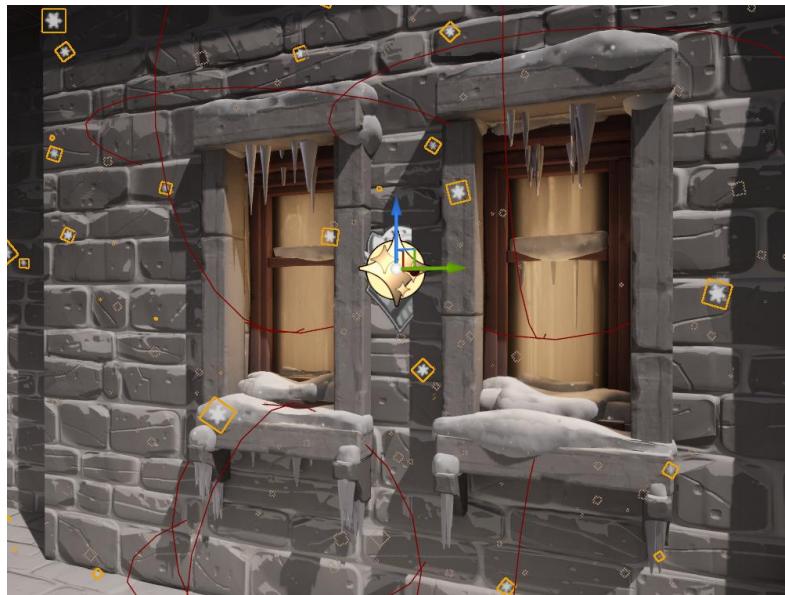
This folder includes the Blueprints and C++ Blueprints that can be dragged into the world. The VFX folder contains the Snow VFX that can be attached to the player. The Required Assets folder contains all the textures and materials used by the tool and can be edited according to the project's needs.



The blueprints can be simply dragged into the world and attached to actors for them to generate snow cover or icicles.



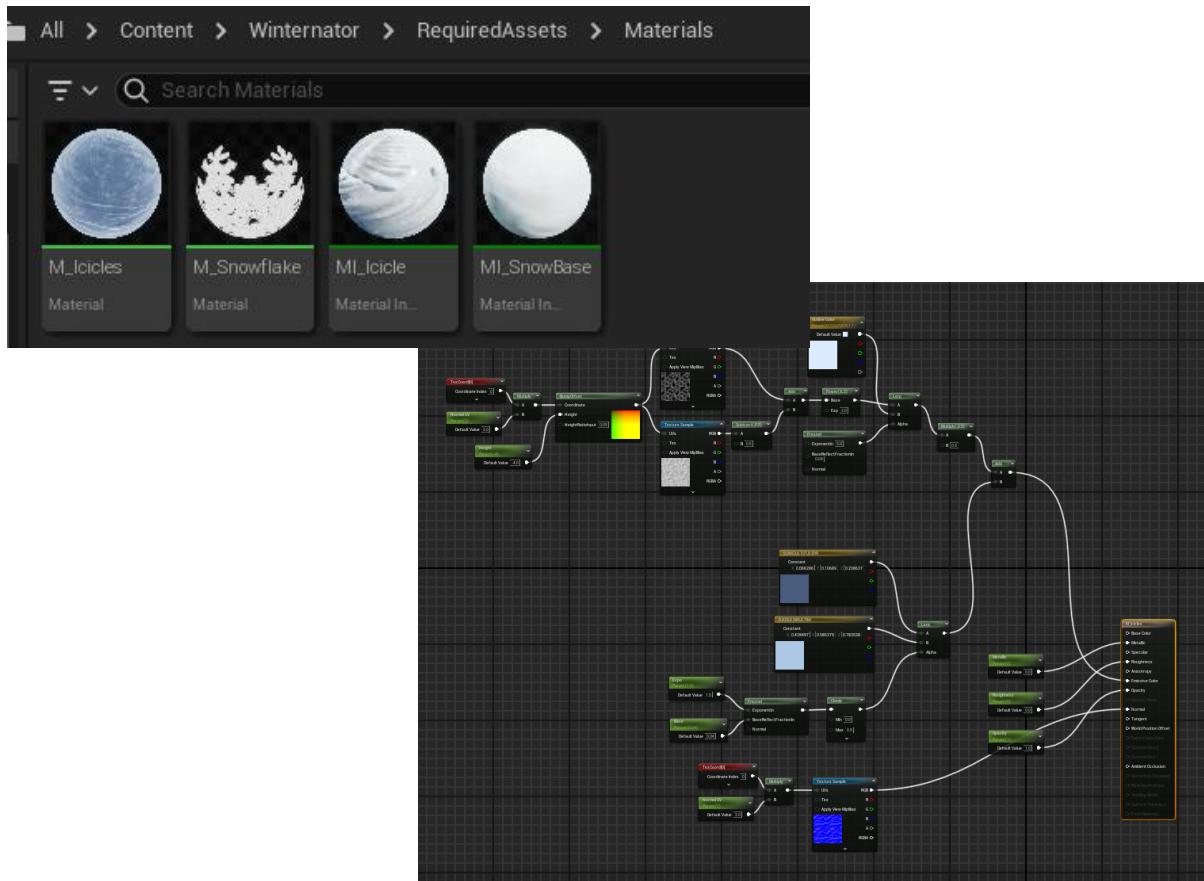
Similarly, the Snow VFX can also be dragged into the world to create a snow effect or also attached to a player so that it follows the camera.



## Material Pipeline

The tool comes with multiple materials and material instances that can be set in the Snow Cover Generator or the Icicles Generator to give the dynamic meshes the desired look.

The materials come with a lot of easily changeable parameters that can be tweaked according to the desired look for the project.



## Assessment Related Requirements to be answered:

- The problem that inspired the tool was that environment artists usually need to hand place each individual icicle and snow cover over on top of small meshes in a map. Now, if the map is the size of a city with countless structures that could potentially grow icicles, it'd take up a long time of the development process.
- The proposed solution for the problem which does not include an existing tool system is modelling multiple versions and hand placing every single icicle and snow cover on all structures and small meshes in the game map.
- Using the icicles generator tool, the environment designer can simply place a volume where they want the icicles or snow cover to generate. The tool will automatically analyse the structure within the volume and procedurally generate realistic icicles or snow cover on it.
- The proposed tool would be used during the environment design and set dressing phase of the development process of a game.

## Horizontal Prototype

- Time of post: 2<sup>nd</sup> of February 2024, Friday at 22:38
- [Link to post.](#)

## Vertical Prototype

- Time of post: 12<sup>nd</sup> of February 2024, Monday at 19:33
- [Link to post.](#)

## Polished System

- Time of post: 22<sup>nd</sup> of February 2024, Thursday at 17:29
- [Link to post.](#)

# Test Plan

## Snow VFX:

Test Item	Expected Result	Actual	Fix
Snow VFX > Curl Noise Frequency	Curl noise frequency of the snow particles changes	Frequency changes successfully	No fix needed
Snow VFX > Curl Noise Strength	Curl noise strength of the snow particles changes	Strength changes successfully	No fix needed
Snow VFX > Snow Area Radius	Radius of the effect changes	Effect radius changes successfully	No fix needed
Snow VFX > Snow Intensity	Intensity of the snow particles changes	Spawn rate changes successfully	No fix needed
Snow VFX > Wind Direction	Direction of flow of the snow particles changes	Wind direction vector changes successfully	No fix needed
Snow VFX > Wind Speed Scale	Wind Strength affecting the snow particles changes	Wind speed scale changes successfully	No fix needed

## Snow Cover Generator BP:

Test Item	Expected Result	Actual	Fix
Snow Cover BP > Actors Selection	Actors can be added and removed to and from the `ActorsToProcess` Array. Nothing would happen if the array is empty	Actors can be successfully added or removed to and from the array. Nothing happens if array is empty	No fix needed
Snow Cover BP > Snow Thickness	Thickness of the snow can be changed	Snow thickness changes accordingly	No fix needed
Snow Cover BP > Adding variation noise to the snow mesh	Noisy mesh deformation	Flat mesh extrusion	Applied perlin noise to mesh and added variables to control the strength and frequency of the noise.
Snow Cover BP > Remesh Type	Change remesh method	Remesh method can be changed at runtime and is applied properly	No fix needed
Snow Cover BP > Performance	Dynamic mesh generates as quick as possible	Dynamic mesh generates very slow on large meshes	Create a new variable to be set by a designer that rejects actors bigger than a certain size in order to prevent accidentally clicking on a large mesh like a landscape

## Icicles Generator BP:

Test Item	Expected Result	Actual	Fix
Icicles BP > Actor Selection	Actors can be selected or deselected to be processed. Nothing would happen if the array is empty	Actors can successfully be selected and deselected to be processed. Nothing happens if the array is empty	No fix needed
Icicles BP > Actor Selection	Actors doesn't need to be within the volume to be selected	The actor doesn't need to be in the volume while the ActorToProcess is assigned	No fix needed
Icicles BP > Sphere Radius	The sphere volume can be changed and is reflected	The sphere volume can be successfully changed and is reflected by the visualizer in the world	No fix needed
Icicles BP > Extrusion Bias	The icicles spawn on the mesh and are flush with the surface	Dependent on the mesh, sometimes the icicles spawn slightly off of the surface	A variable ExtrusionBias is introduced to allow the designer to set a custom offset for edge cases
Icicles BP > Points to Scatter	Points are randomly scattered on the mesh to append the icicles to	The PointsToScatter variable isn't an accurate reflection of the points being used on the mesh to scatter the icicles	No fix found
Icicles BP > Cone Radius	The base radius of the icicles can be changed	The base radius of the icicle cones can successfully be changed	No fix needed
Icicles BP > Cone Height	The height of the icicles can be changed	The height of the icicle cones can successfully be changed	No fix needed
Icicles BP > Performance	Icicles are generated as quickly as possible	Time taken to generate is proportional to the poly count of the mesh	A variable DecimationTolerance is introduced to decimate the original mesh to reduce generation time

## Snow Cover Generator C++ BP:

Test Item	Expected Result	Actual	Fix
Snow Cover CPP BP > Actors Selection	Actors can be added and removed to and from the 'ActorsToProcess' Array. Nothing would happen if the array is empty	Engine crashes when the array is empty	Check if array is filled, else return
Snow Cover CPP BP > Snow Thickness	Thickness of the snow can be changed	Snow thickness changes accordingly	No fix needed
Snow Cover CPP BP > Adding variation noise to the snow mesh	Noisy mesh deformation	Noise is applied to the snow mesh	No fix needed
Snow Cover CPP BP > Remesh Type	Change remesh method	Remesh method defaults back to triangles mode	Use an enum to define the method and set it accordingly in the ApplyUniformRemesh function
Snow Cover CPP BP > Mesh Smoothing	Mesh surface gets subdivided and smoothed	Mesh surface stays blocky and low poly	Set the dynamesh variable for each post process
Snow Cover CPP BP > Performance	Dynamic mesh generates as quick as possible	Dynamic mesh generates faster than the blueprint prototype and has significantly better performance and less resource costs	No fix needed

## Icicles Generator C++ BP:

Test Item	Expected Result	Actual	Fix
Icicles CPP BP > Actor Selection	Actors can be selected or deselected to be processed. Nothing would happen if the array is empty	Engine crashes if the array is empty	Check if array is filled, else return
Icicles CPP BP > Actor Selection	Actors doesn't need to be within the volume to be selected	Engine crashes if the selected actor is outside the volume	Check if the selected actor's distance is greater than the sphere radius to make sure the actor is within the volume, else return an error message
Icicles CPP BP > Sphere Radius	The sphere volume can be changed and is reflected by the visualizer in the world	The sphere volume didn't appear in the editor	Set the sphere radius in the constructor
Icicles CPP BP > Extrusion Bias	The icicles spawn on the mesh and are flush with the surface	Dependent on the mesh, sometimes the icicles spawn slightly off of the surface	A variable ExtrusionBias is introduced to allow the designer to set a custom offset for edge cases
Icicles CPP BP > Points to Scatter	Points are randomly scattered on the mesh to append the icicles to	The PointsToScatter variable isn't an accurate reflection of the points being used on the mesh to scatter the icicles	No fix found
Icicles CPP BP > Cone Radius	The base radius of the icicles can be changed	The base radius of the icicle cones can successfully be changed	No fix needed
Icicles CPP BP > Cone Height	The height of the icicles can be changed	Icicles are spawned as flat circles	The math was incorrect and the cone height was being set as 0. Fixed by using the random stream to variate the cone height between a range above and below the set cone height by the designer
Icicles CPP BP > Performance	Icicles are generated as quickly as possible	DecimationTolerance variable decimates the mesh to reduce generation time	No fix needed

# System User Manual

## System brief

The Winternator tool is a collection of procedural systems that aids in the development of designing a winter themed environment. The tool would be used by environment designers to significantly speed up the development process by generating dynamic and procedural snow cover and icicles on any given mesh to quickly decorate or set dress massive maps.

## System output

**Snow VFX:** The Snow VFX is a procedural Niagara Particle System that has various exposed user parameters to customize the system according to your project.



**Snow Cover System:** The Snow Cover System is a procedural dynamic mesh generation system that harvests the power of the experimental Unreal Engine plugin Geometry Script. It creates realistic snow cover on top of any given mesh(es).



**Icicles Generator:** The icicles generator is also a procedural dynamic mesh generation system that uses the experimental Unreal Engine plugin Geometry Script. It scatters realistic icicles on any given mesh within a volume.



## Development cycle application

This tool would be used by the design team, specifically environment designers to quickly and procedurally generate Snow VFX, snow cover and icicles to decorate and set dress a map saving valuable development time by automating the process instead of having to 3D model and hand place every single asset in a massive map.

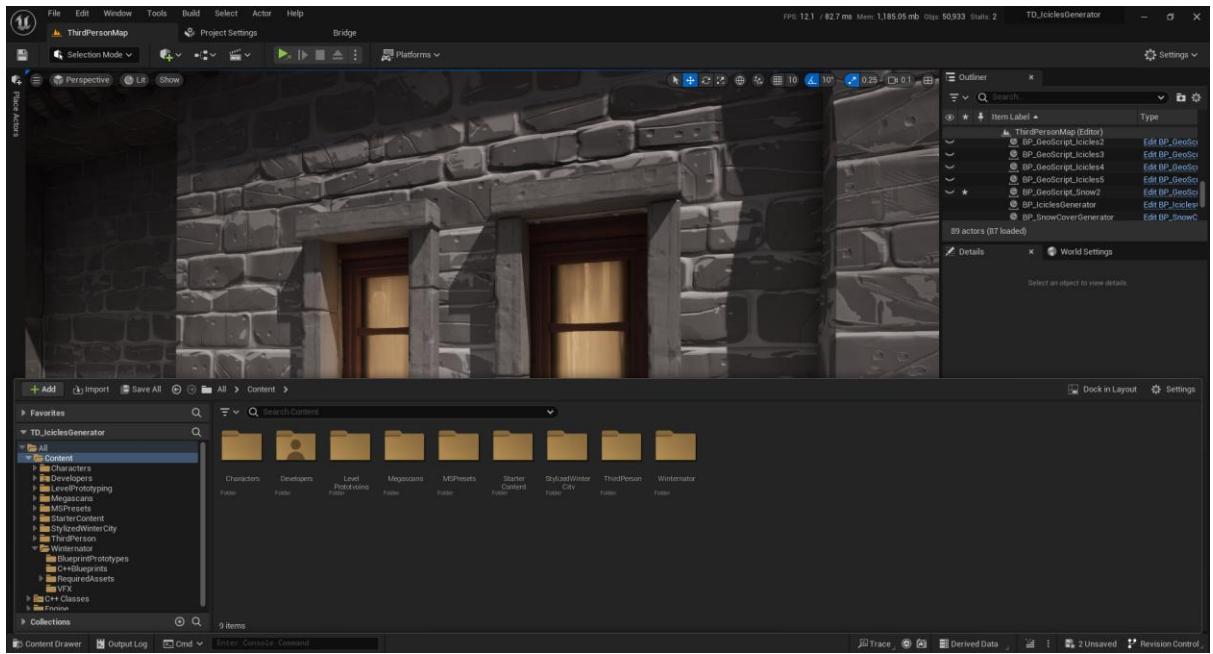
## System Requirements

- Operating system: Windows 10 64-bit.
- Processor: Quad-core Intel or AMD 2.5 GHz or superior.
- Memory: 8GB RAM.
- Graphics Card: Any DirectX 11 or 12 compatible card.

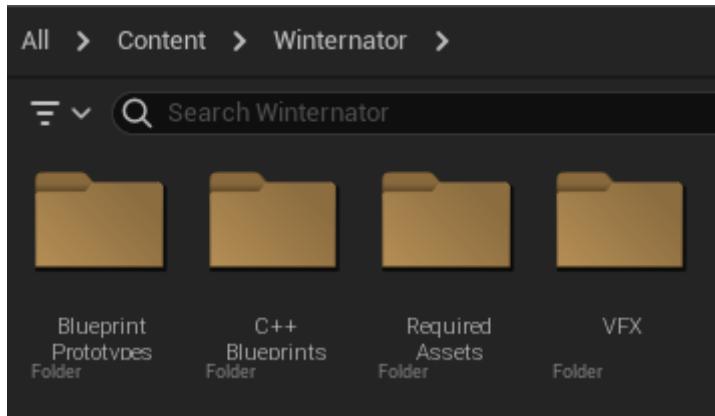
## User Guide

This guide will walk you through using the tool and explain every component and its variables in detail.

1. Start a new project or open an existing project that you want to use the tool in.
2. Extract and import the tool folder into the project directory under the Content folder.



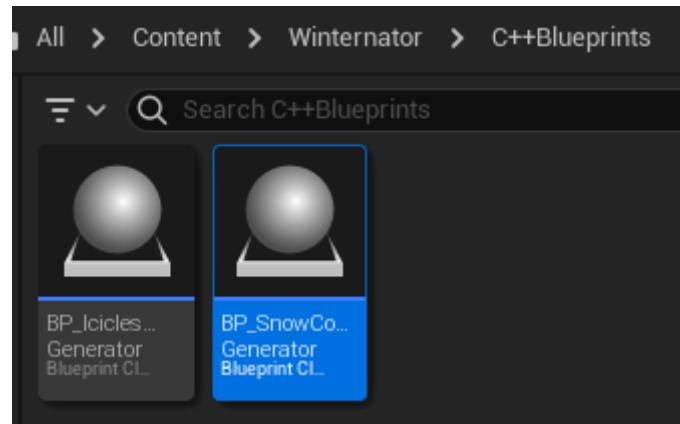
3. Open the Winternator folder and you will be greeted with more folders that contain the main systems included in the tool.



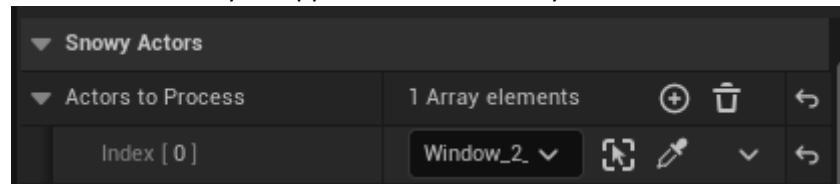
- Blueprint Prototypes* are the initial prototypes that I made purely in blueprint. These are slightly inefficient and slower.
- C++ Blueprints* are the blueprints that are derived from C++ classes. These are more efficient and performant than the pure blueprints.
- Require Assets* are all the materials, and textures that are used by the tool itself, you can change these to your liking and according to the looks of your project.
- VFX* contains the Snow Niagara Particle System that can simply be dragged into the world or attached to a player.

### Snow Cover Generator:

1. Open the C++ Blueprints folder and drag in the BP\_SnowCoverGenerator asset. This is the procedural snow cover generation system. It can be placed anywhere in the world really, but it is recommended to place it near the assets you're generating the snow cover as multiple generators can get confusing as to what assets they're associated with.

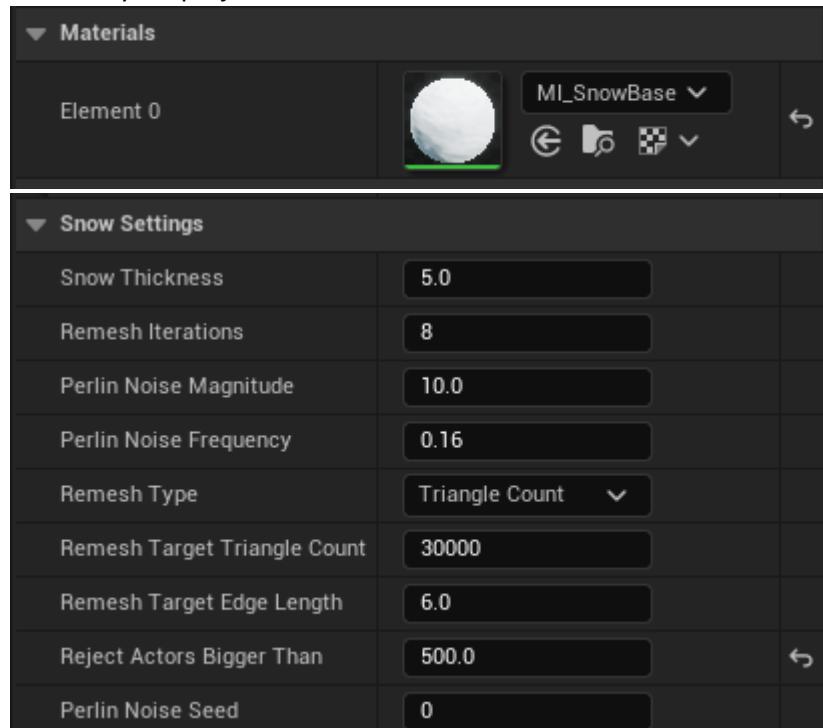


2. Once the actor is in the world, make sure it is selected and look in the details panel.
3. Under the Snowy Actors section, you can select what actors you want to generate the snow cover on top of by clicking the little + icon to add an element to the array then using either the drop down menu or the eyedropper tool to select any actor in the editor viewport.



4. Once you've selected all your actors you want to generate the snow on, we can move onto the Snow Settings section where you can tweak the look of the snow cover. Let's look at what each option in this panel does.

- a. First, you can set the material of the snow cover. There is a material called MI\_SnowBase included with the tool and can even be tweaked accordingly to the needs of your project.



- b. *Snow Thickness*: This controls the thickness or the depth of the snow in cms.
- c. *Remesh Iterations*: The amount of times the snow mesh is remeshed and smoothed.

- d. *Perlin noise magnitude*: The snow mesh is deformed according to a perlin mesh, this option controls the magnitude or the strength of the noise.
  - e. *Perlin noise frequency*: This controls the scale of the perlin noise deformation applied to the snow mesh. Lower values work better between 0 and 1.
  - f. *Remesh type*: The method by which the snow is remeshed. Can be either Triangle Count or Edge Length.
  - g. *Remesh target triangle count*: If Triangle Count remesh type is selected, this will control the target poly count of the dynamic mesh.
  - h. *Remesh target edge length*: If Edge Length remesh type is selected, this will control the target edge length of the dynamic mesh.
  - i. *Reject actors bigger than*: This variable is to prevent accidentally clicking on massive meshes like landscapes as it can freeze the engine for a long time while the dynamic mesh is generated.
  - j. *Perlin noise seed*: This controls the seed of the perlin noise and can be changed to tweak the look of the deformation of the snow mesh.
5. Finally, there is the option to freeze or unfreeze the mesh under the Dynamic Mesh Actor section. This is a simple button that can be used to freeze the dynamic mesh once you're happy with your generated snow cover to be used in the game. This can also be used to save processing power while working on other things in the editor or if you want to move the generated mesh slightly after generation according to your needs.

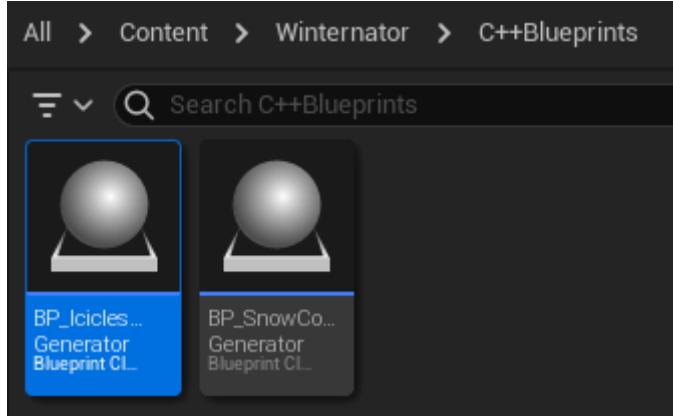


6. You now have your procedurally generated snow cover!

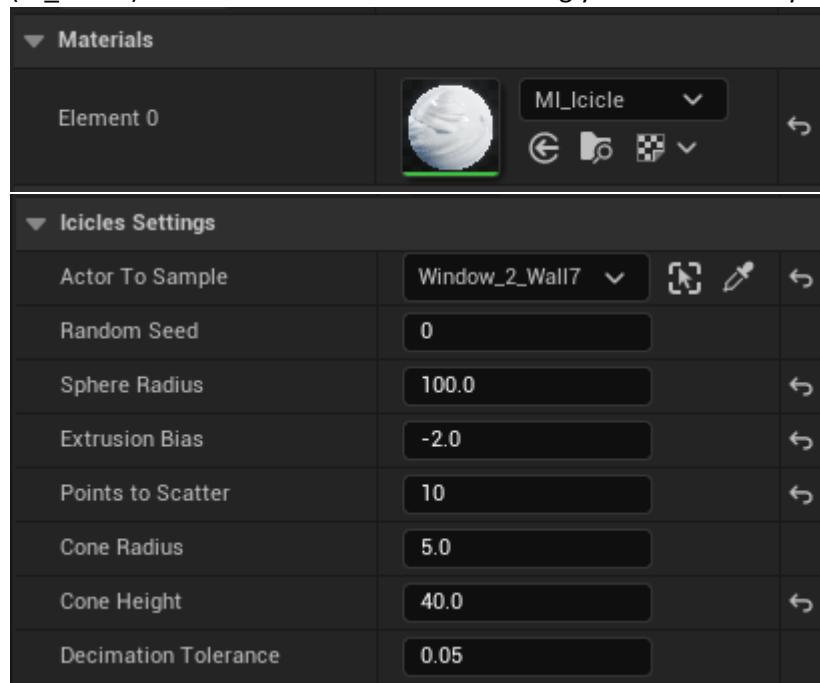


## Icicles Generator:

1. Next in the folder is the icicles generation system. Drag in the BP\_IciclesGenerator asset into the world, make sure it is selected and you'll be greeted with similar options in the details panel.



- a. First, you can set the material of the generated icicles. There are 2 materials included with the tool, one stylized (MI\_Icicle) and the other more realistic (M\_Icicles) and can be even tweaked accordingly to the needs of your project.

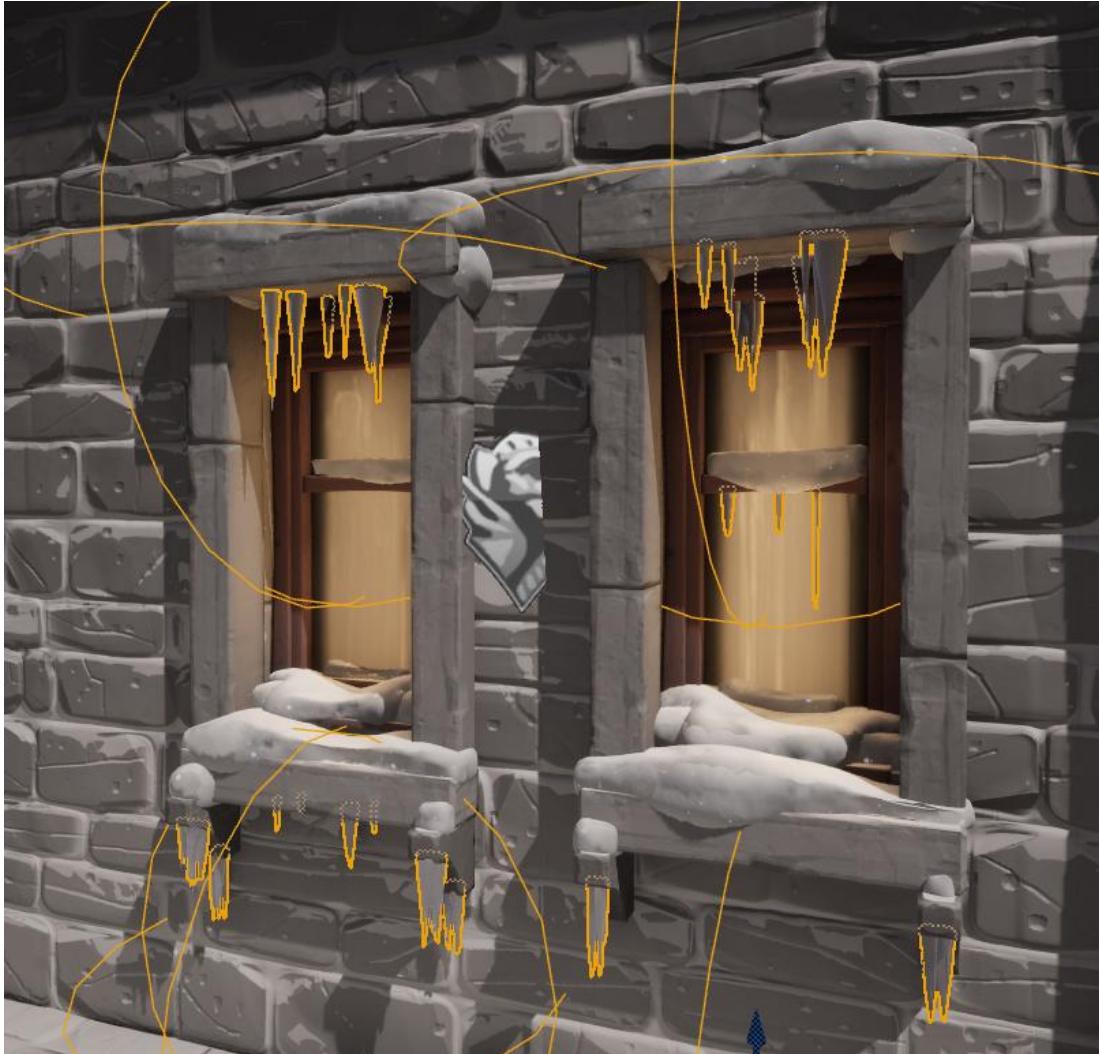


- b. *Actor to Sample*: You can select the actor you want to sample within the processing volume sphere by either selecting it from the dropdown menu or selecting it in the viewport with the eyedropper tool.
- c. *Random seed*: this value is the seed for the height and base radius variation of the icicles.
- d. *Sphere radius*: this controls the radius of the radius visualizer and the processing volume sphere. Change this according to where you want the icicles to generate and make sure only the areas you want the icicles to spawn on are within the volume.
- e. *Extrusion Bias*: All meshes are slightly different so this option allows you to set an offset to the icicles so they're flush with the surface of your mesh.

- f. *Points to scatter*: these are the amount of points scattered on the surface of your mesh and controls the amount of icicles that will be spawned on the surface.
  - g. *Cone Radius*: this is the base radius of the icicles. The spawned icicles will have a slight variation between the base radii but will be around this given radius.
  - h. *Cone Height*: this is the average height of the icicles. The spawned icicles will have a slight variation between the heights but will be around this given height.
  - i. *Decimation tolerance*: This is how much the mesh is decimated in order to obtain the points on which to scatter the icicles.
2. Finally, there is the option to freeze or unfreeze the mesh under the Dynamic Mesh Actor section. This is a simple button that can be used to freeze the dynamic mesh once you're happy with your generated icicles to be used in the game. This can also be used to save processing power while working on other things in the editor or if you want to move the generated mesh slightly after generation according to your needs.

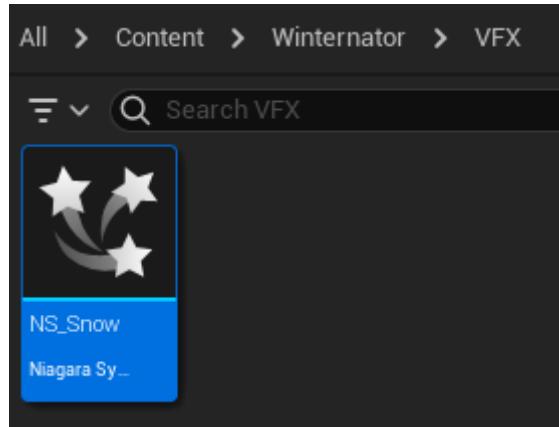


3. You now have your procedurally generated icicles!



## Snow VFX:

Under the VFX folder inside the Winternator directory you can find the Snow Niagara Particle System called NS\_Snow. This can simply be either dragged into the world if you're creating a static scene or be attached to the player so that it follows your character in the game.



It has some user exposed parameters that can be used to change the look and feel of the snow particles:

1. *Curl Noise Frequency*: This is the frequency of the curl noise being applied to the particles, so their falling trajectory is more unpredictable and jittery.
2. *Curl Noise Strength*: This controls the strength of the curl noise affecting the snow particles, the more the jitterier the particles will be.
3. *Snow area radius*: The particles get spawned in a sphere, this option controls the radius of that sphere, the bigger the value, the larger the area of the particles spawn area but lesser particles in the given area.
4. *Snow intensity*: Number of particles being spawned within the snow area radius. Tweak this according to your area radius as the intensity of particles would look less in a bigger radius and vice versa.
5. *Wind direction*: This controls the direction of the wind force being applied to the snow particles. The values correspond with the x, y, and z world axes.
6. *Wind speed scale*: The overall scale of wind speed. The higher the value, the stronger the wind force will be.

