SpragueInSanity Create Snow

Contents

Introduction

This all started off as a crazy idea to create snow on my current terrain and game objects in the Unity game environment. I wanted to have the ability to apply snow without having to have snow specific game objects or terrain versions. Just flip a switch on the current game scene and I have a winter wonderland. Then, flip a switch and I have all my original scene back as it was before.

I started off with some research into using Shader Graph. It did a decent job and provided the appearance of snow-covered objects and terrain. But I soon hit limitations:

- For game objects, no depth or thickness to the snow. While it did look like snow on the top of objects, it was very uniform and looks painted on.
- For the terrain, the shader did not handle the SpeedTrees nor the paint detail layers for grass and grass flowers. Looked like it snowed in the Spring.

Then, I went down the rabbit hole when I asked myself: "How hard could it be to add snow to tops of objects and the terrain such that it has thickness? It can be manipulated to not be a square block and handle miscellaneous terrain objects?" Well, for a newbie like me, it was a fun learning experience but at times a frustrating trial and error endeavor.

I wanted to share what I created just in case anyone else could benefit from some of the implementation details. Not just snow specific stuff, but other techniques that I implemented. In other words, I want to pay it forward. I learned so much from the Unity community of developers and leveraged lots of existing solutions. I often hit the wall and luckily, gracious folks have shared their knowledge openly and freely. Please see the References section at the end of this document for the resources that I used. If you find the code lacking, maybe my consolidated list of resources helps with research for various topics including those specific to this code sample.

Some techniques that I have implemented in this code sample are:

- Ability to find a section of game object mesh based on normal direction and angle to copy that piece of mesh. This is used to find "TOP" or "UP" for tops of game objects.
- Ability to take this mesh, perform extrusion and build a whole new object that matches the top of the original game object. This is my thick snow.
- Ability to perform subdivision on the new mesh object to provide more detail. Often, the mesh created has very few vertices and triangles. Subdivision is necessary if I want to smooth out some sharp edges to soften the look of the new snow object I just created.
- Ability to apply Laplacian smoothing algorithm to the new mesh to soften up the hard edges. This is where doing subdivision prior to this helps.
- Ability to apply snow material with parameter for amount to both the terrain and to all SpeedTrees using a Shader Graph. And, reset it back since the shader material is a permanent change to the SpeedTree material. It is very easy to set back.
- Ability to define an area of Terrain in the vicinity of your player to give appearance of thick snow on the ground. In addition, it removes paint details for grass, grass flowers, etc. Best part is this is done to the clone of terrain so these changes are memory only, not permanent.

I hope folks find the code sample helpful. If for nothing else, but for learning and implementation for other purposes. I am sure some of the stuff I did is not the most efficient and clever, but for an old hacker like me, good enough. Some of the appearance stuff can use some tuning.

One thing that I hope this code provides is a solution for the non-generic cases. You know, not the well-defined simple cube with known dimensions and sizes. But crazy objects with nooks and crannies. Well, at least somewhat crazy. Future stuff I may eventually get around to implementing is deforming the snow based on player footprints. Also, enhancing the object creation to look better than it does now. But, not bad for an initial stab at solving my need for snow cover.

Assumptions:

This code sample assumes the following environmental set-ups in Unity:

- Project is URP, not 3D. This has not been tested using the older default render engine. It has only been tested and used using the Universal Rendering Pipeline rendering engine.
- Standard Assets package has been installed for SpeedTrees under default Assets folder.
- This was created and tested on Unity version 2021.1.16f1

Disclaimer

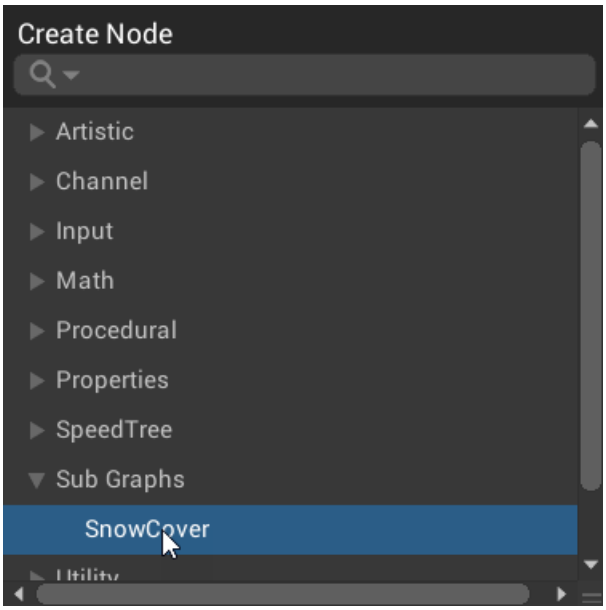
I take no responsibility for bugs and issues resulting from my code. Use at your own risk. My suggestion is to set-up a sandbox and try it out. Works, fantastic! Doesn't work? Well, maybe provide solutions and pay-it-forward. Bottom line is to have fun. Please read carefully the How It Works section below for set-up if you decide to try to run the code as-is. This also has been tested on SpeedTrees using on the Conifers trees.

How It Works

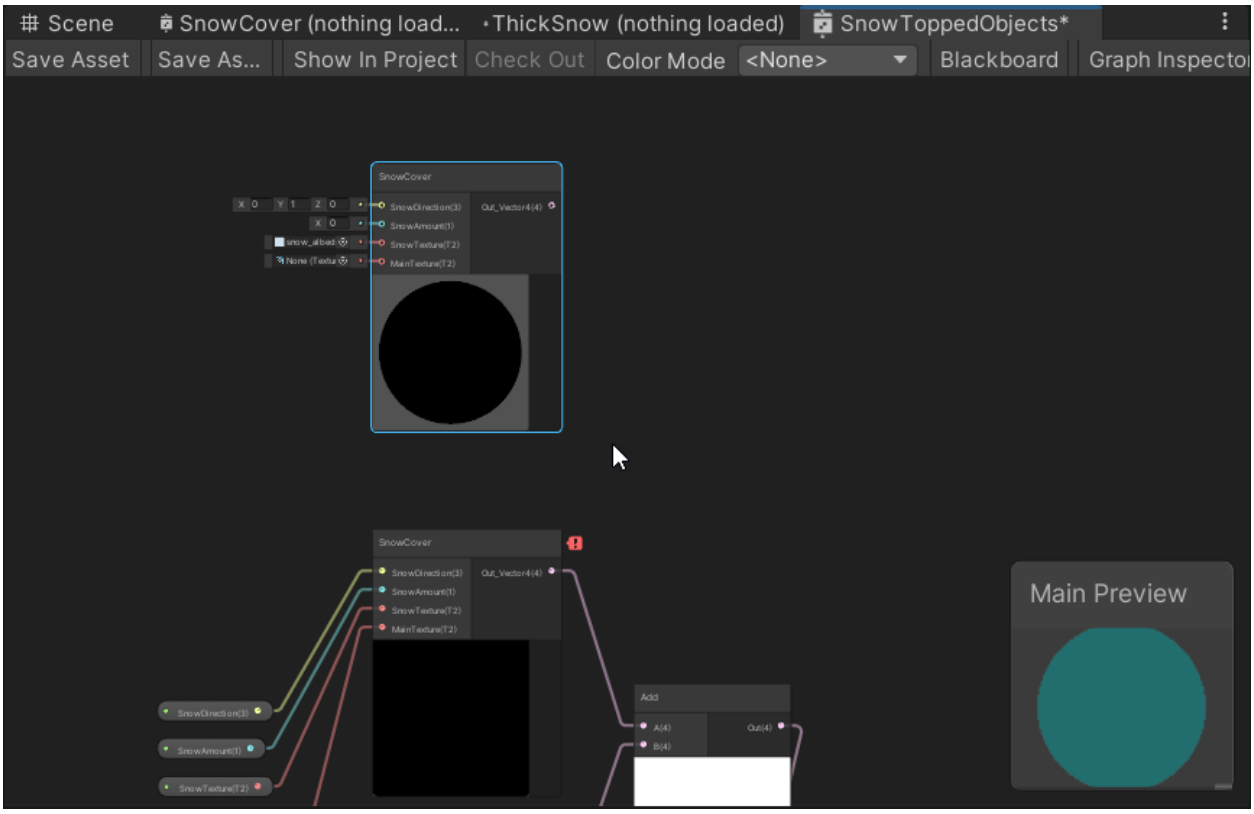
I will try to give a high-level explanation of how this sample code works. We will see how successful I am.

Download the zip file and unzip the files. Copy the files/folders into Assets folder of your sandbox URP project. I provided a demo script called DemoCreateSnow.cs. Just create an Empty Game Object, copy this script to it. Then, drag your Terrain object to parameter on the Inspector.

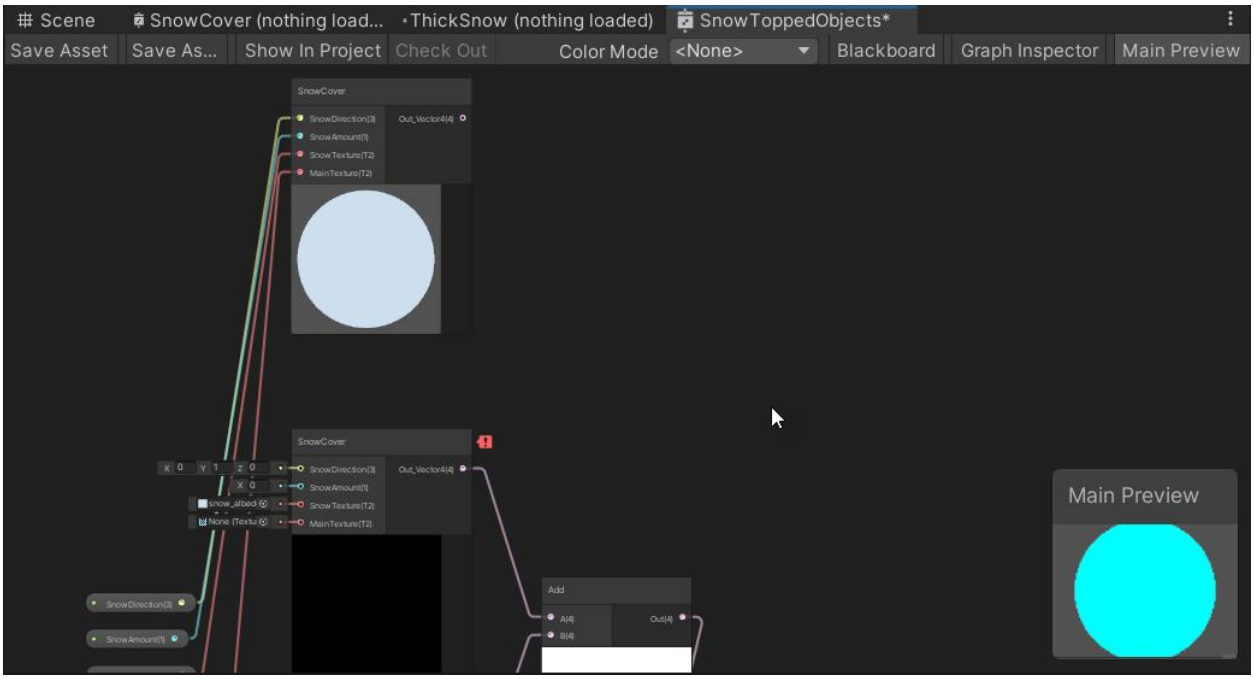
The other change that needs to be made is correction of the SnowToppedObjects and SnowToppedTerrain shader graphs. Both use a subgraph. Unfortunately, the subgraph has some unique UID that changes when you import into another project. The material will be pink and an exclamation mark will appear on the SnowCover node in the Shader Graph editor. So, to fix this, you need to remap. Easy enough to do. Double click on both shadersgraph objects in Assets/Sprague InSanity/Shaders folder. This should open editor window for both. In individual window for each shader graph, right mouse click in open area, chose Create Node. Click on Sub Graphs, select SnowCover. For example:

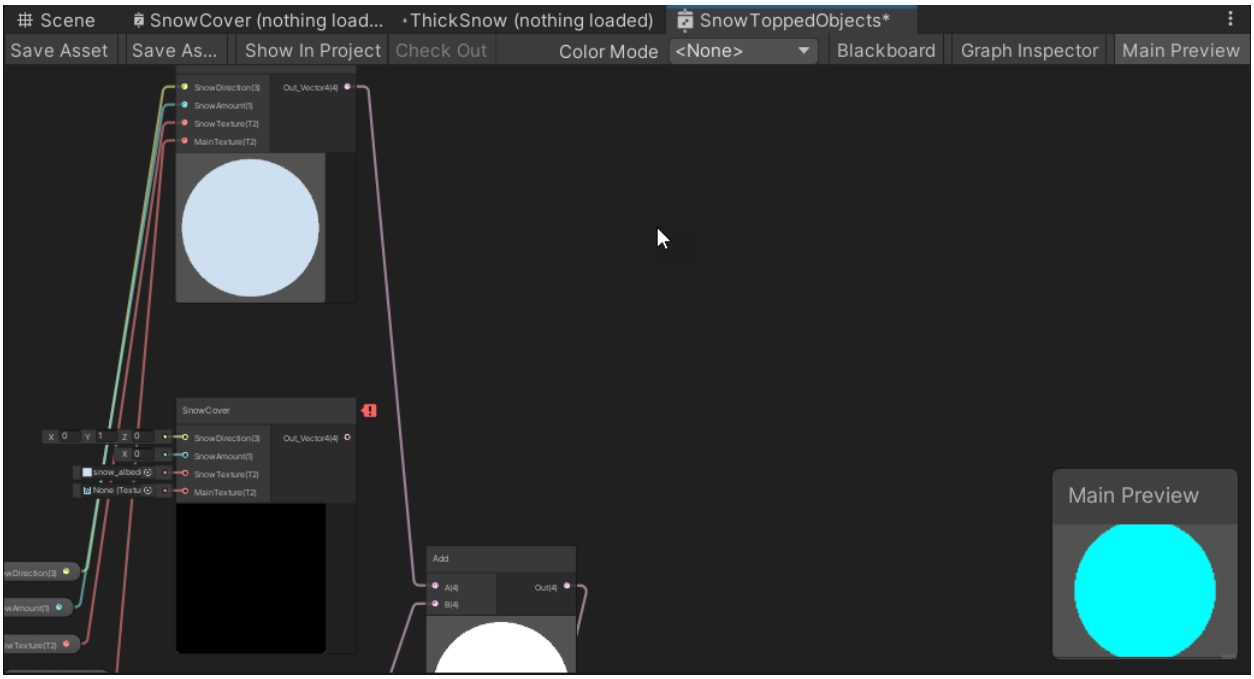


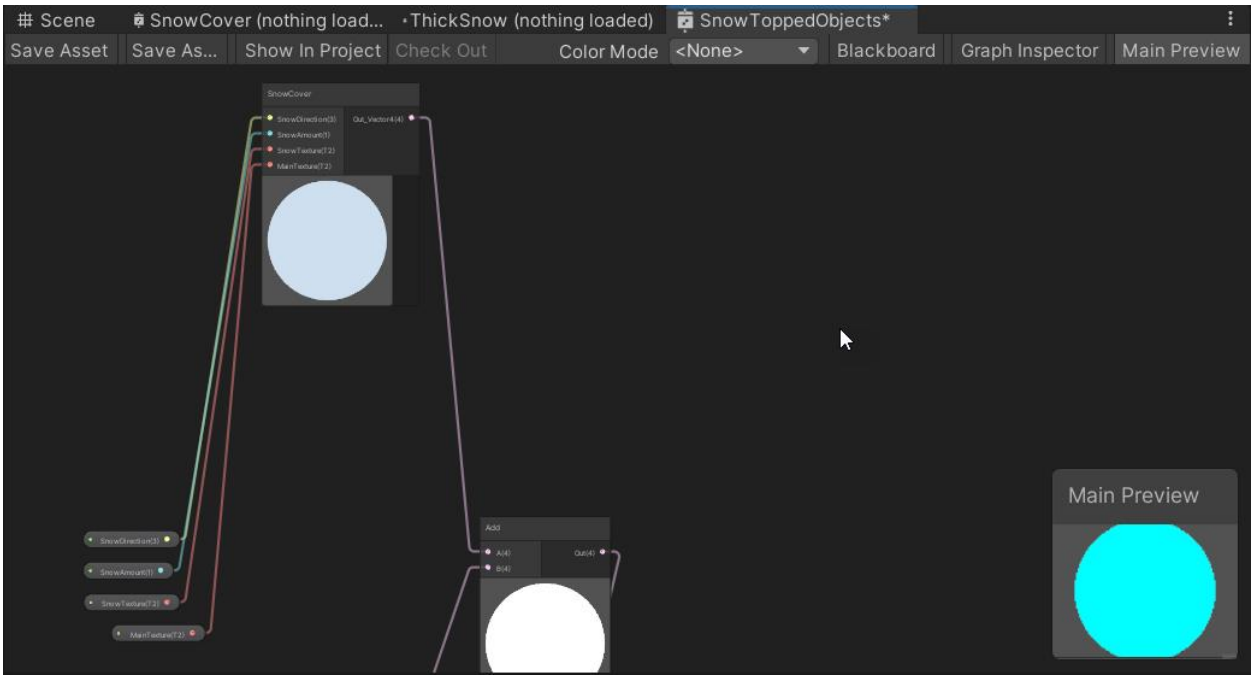
You should get a new node added.



Remap input parameters from object with exclamation to new object, delete old object. Save. Do this for both shaders.







Save both shaders clicking on Save Asset button in window editor. Now, the material must be remapped:

- Go into folder Assets/Resources. Click on SnowToppedObjects.mat. In Inspector, click on Shader drop down, select Shader Graphs, select the custom SnowToppedObjects.
- Do the same for SnowToppedTerrain.mat. Select Snow Topped Terrain.

This should fix any pink issues in the scene when the game is run.

Upon execution of the game scene, this script will look for a JSON file, UISnowOptionsSettings.json, for configuration/run-time information in the top-level Assets folder of your game environment. I use this for a crude UI config screen. You can edit it directly using your favorite editor. This file sets the following run-time parameters:

- startPosition: world location to create the Terrain thick snow mesh. Edit this for your environment immediate scene.
- Width/depth: size of Terrain thick snow mesh. Edit the size you want.
- farOffSnowCoverage: Controls terrain snow shader snow amount. 0-1
- treeSnowAmount: Controls SpeedTree snow shader snow amount: 0-1
- snowShader: This is the Shader Graph to apply snow effect to the Terrain and SpeedTrees
- defaultTreeShader: IMPORTANT: SET TO YOUR ENVIRONMENT. This is the SpeedTree default material that gets reapplied when turning off snow. Depending on the type of trees created in your Terrain using Terrain Tools, look at folder Assets/Standard Assets/Environment/SpeedTree. In my case, I use Conifer. So, continue down Conifer/Conifer_Desktop Materials/ then all subfolders. Each material will have the shader. Make sure that is configured in this field. NOTE: ASSUMPTION IS ONLY ONE IS USED FOR ALL TREES. THIS SAMPLE HAS NOT BEEN SET-UP TO HANDLE VARIOUS SHADERS TODAY. FUTURE ENHANCEMENT. THIS HAS BEEN TESTED ONLY FOR CONIFER.
- SnowObjectsFile: this is a JSON file that lists each game object to apply snow. I really wanted something more automated but had to compromise with this per object implementation. However, this implementation of a per-object configuration has its advantages. First, this implementation allowed me to narrow the range of objects to apply snow. For a large scene, cuts down on execution time to create snow. Also, it allowed me to exclude complex, multiple LOD meshes. Second, gave me more control over different objects where a one-size-fits-all does not work. Such as definition of up, angle to use, rather to use subdivision or smoothing. This file is located in Assets default folder. Config for your specific scene objects.

The script then instantiates CreateSnow.cs and passes the parameters found to each member function:

- CreateSnowForTerrain – This applies the snowshader for areas not covered by the mesh. Creates the mesh for snow in the area defined.
- CreateSnowForObjects – This opens the SnowObjectsFile and processes each game object configured if active is true. This is where the items I mentioned in the introduction are applied. Mesh extraction, extrusion, subdivision and finally smoothing.
- CreateSnowForSpeedTrees – Applies shader graph to the SpeedTrees material to give snow covered appearance on all SpeedTrees
- RemoveTerrainPaintDetails – Clones terrain and removes paint details of grass, grass flowers from in memory copy.

Then, when the game is stopped, OnDestroy is called:

- ResetSpeedTrees – puts back the original material/shader. See above configuration file and caveats. IF FOR ANY REASON YOUR SPEEDTREES STAY SNOW COVERED, JUST CREATE A DUMMY SCRIPT TO APPLY ResetSpeedTrees AND HAVE IT PASS IN THE ORIGINAL MATERIAL/SHADER.
- ResetTerrain – cleanup of in memory stuff – back to original terrain

- DestroySnowObjects – removes temp snow cover objects.

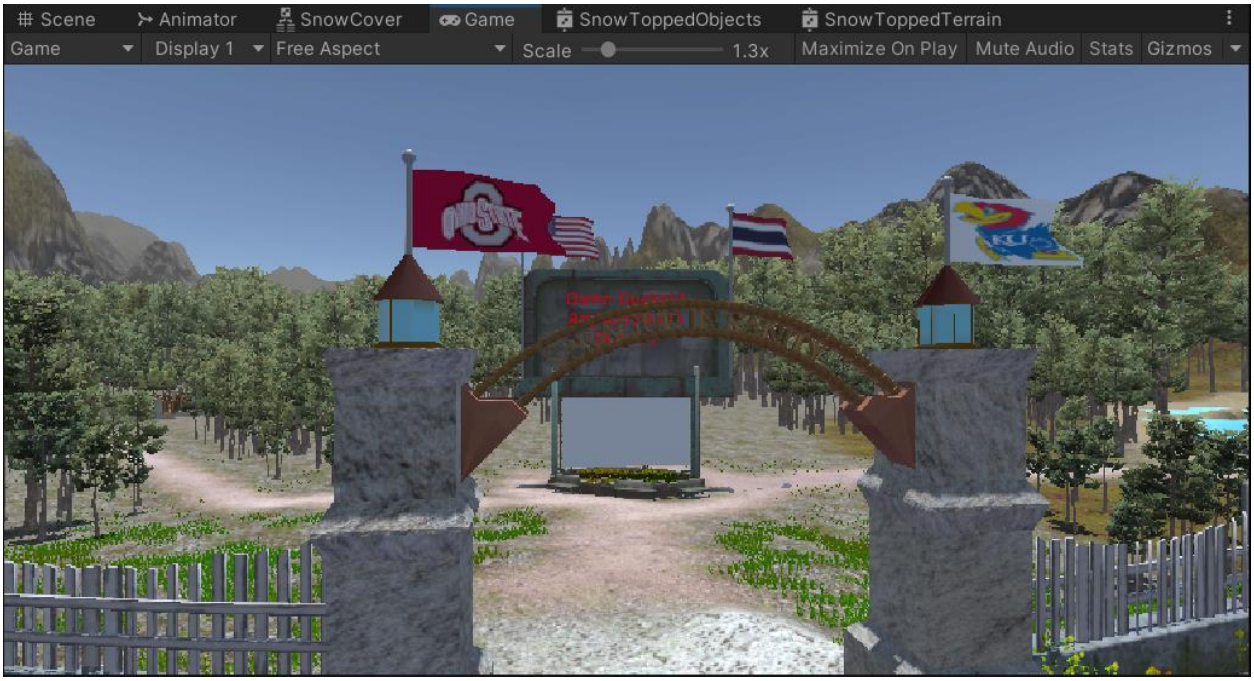
SnowObjects.json

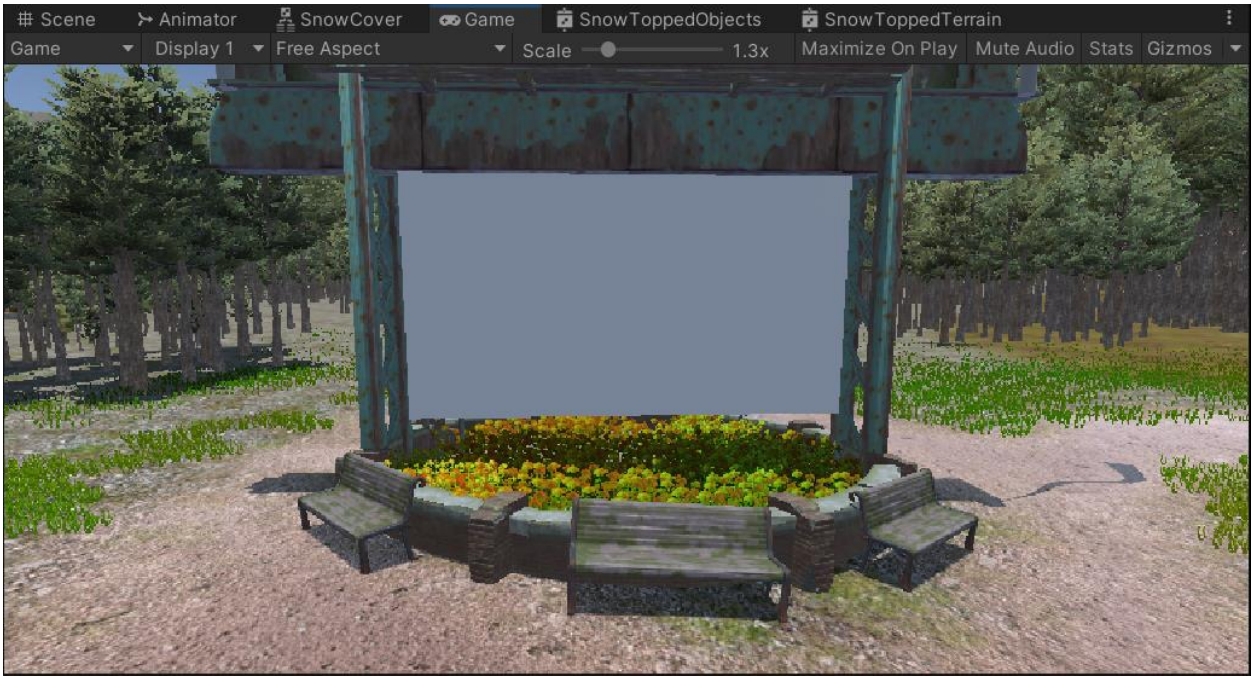
Here is the explanation of this configuration file. This controls which game objects get snow and how much snow.

- gameObjectTopLevelName: This is the first level hierarchy name of the game object in Unity.
- gameObjectPartName: This is the actual game object mesh to find top, extrude, etc. If this is a simple object that only has one level, then this name should be set-up with the same value in gameObjectTopLevelName. The combination of both levels helps uniquely address individual objects that may have been duped in the scene.
- extractAngle: This is angle of search to find the mesh area to extract.
- extractDirectionX, extractDirectionY, extractDirectionZ: Represents direction of search. Corresponds to Vector3.up, down, forward, back, left, right. NOTE: ONLY UP HAS BEEN TESTED ALTHOUGH OTHERS SHOULD WORK.
- ExtrudeDistance: How big to make snow object on top of original object.
- subDivisionPasses: How many times to subdivide for detailed mesh. Limit max is 6.
- smoothnessPasses: How many times to apply Laplacian algorithm
- smoothnessFactor: Provides ability to constrain amount of vertex movement by Laplacian algorithm. I added this in case I wanted to play with make crazy looking stuff.
- adjustXPivot, adjustYPivot, adjustZPivot: I found some game objects that did not have a pivot point at the 0,0,0 local object coordinates nor in the center. This allows it to be adjusted in memory to allow angle and direction to find current mesh segment using pivot point as anchor.
- debugVertexRaysOn: this ties into above parameters. If desired mesh is not found, set this to true to see which way the angle, direction and pivot is looking for the mesh part. This will help diagnose object issues and use the adjustment parameters to compensate. Maybe have to play with direction as well if coordinate of local object is not correct.
- Active: set to true to apply snow.

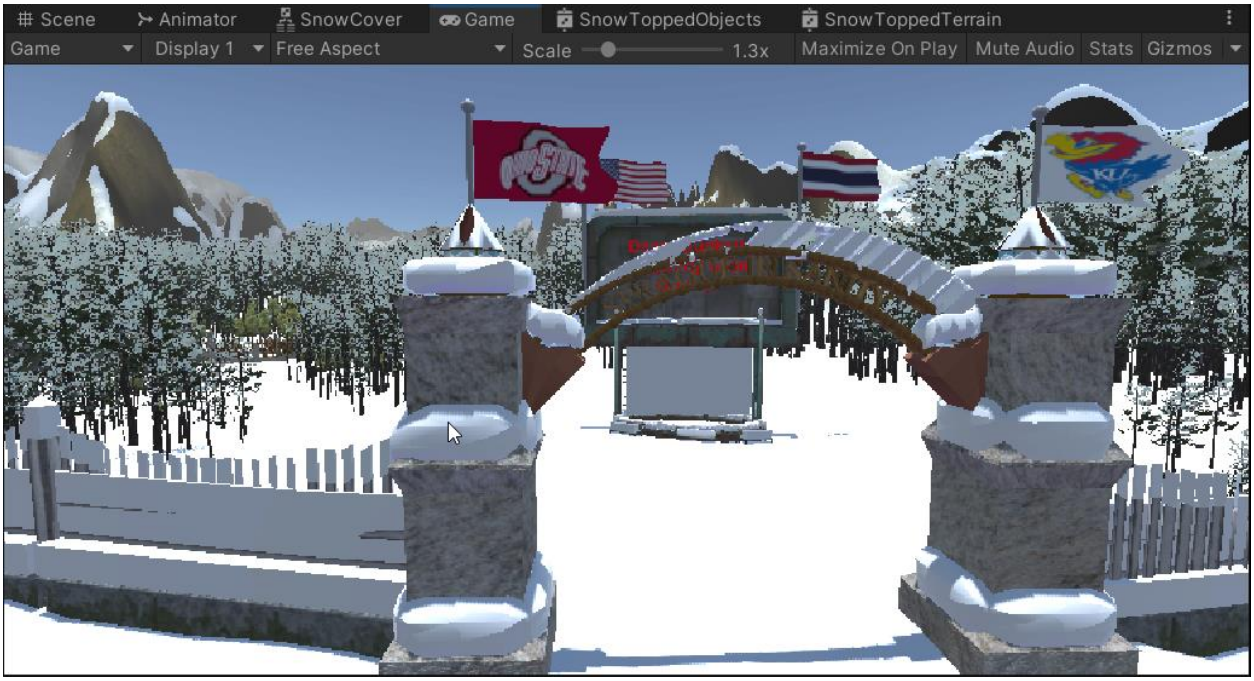
Examples

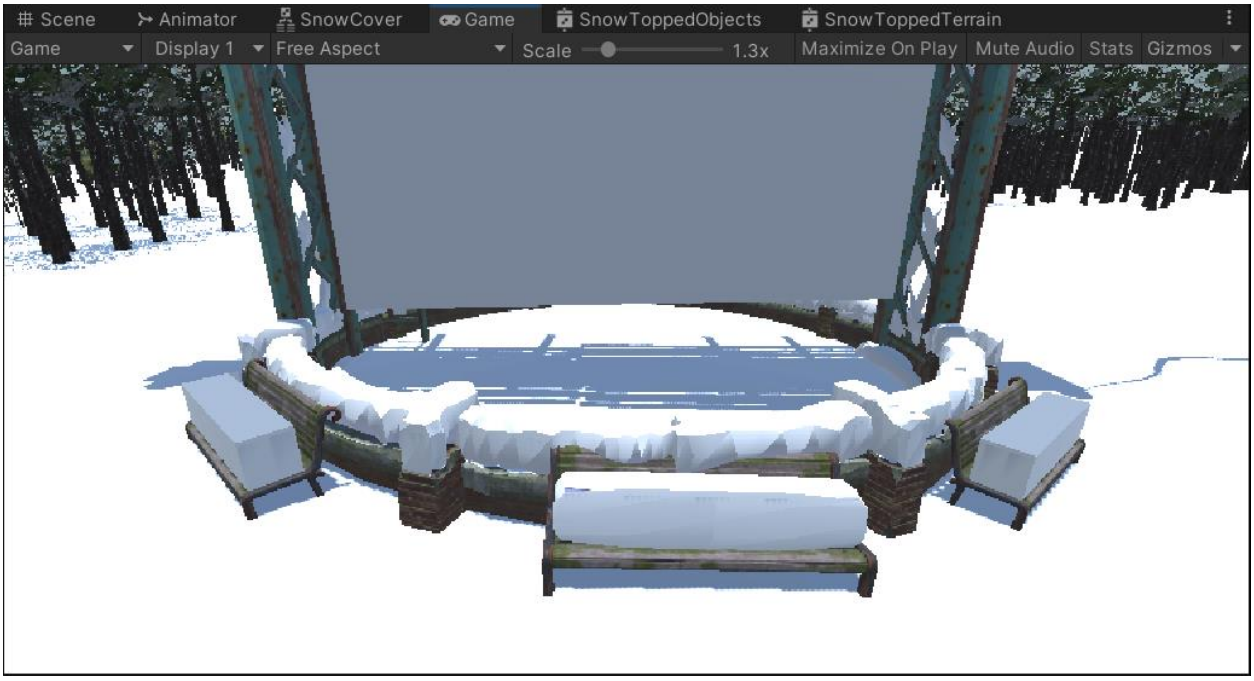
Here is a snapshot of my environment before Create Snow:





And here is after snapshot with exaggerated extrusion size for demonstration of the ridiculous. Also, it includes one bench with subdivision/smoothing while the other two do not. You can see the difference in snow objects.





References/Acknowledgements

This section lists all the resources used to created Sprague InSanity. Everything from instructional videos, posts, published shared code, shared game objects and helpful solutions. While there will be overlap, the following sections try to categorize solutions by problem area. Keep in mind that some of the videos are older and done on older versions of the software. But the concepts and techniques still apply.

I apologize for any missed references. In the heat of battle with my environment trying to get things to work, I did not save all links for valuable information I found. Many, many folks have provided great solutions for newbies like me on Stackoverflow, Unity Community and forums.

Blender Introductory/Intermediate Tutorials

- https://youtu.be/TPrnSACiTJ4 : Very detailed tutorial by Blender Guru on making a donut and coffee cup. Lots of techniques and tricks. This is more geared to a model look fantastic in Blender and not geared towards a game object for Unity. But, great for learning Blender.
- https://youtu.be/elUJCEC06r8 and https://youtu.be/fZSD7pVIUkY : This is another set of videos that might be easier to digest and gives good information on Blender. This video series by tutor4u shows how to make a hammer with metal and wood effects.
- https://youtu.be/kejQ8nX5YZA , https://youtu.be/_0sMYvk1o-8 , https://youtu.be/yffWd4kI51Q : Top tools/techniques to use for all modeling by CG Cookie.
- https://youtu.be/JMBMHSca_j0 : This is an awesome video by CG Boost that lists 100+ tips/tricks and shows how to model a complex object. The video also has a link to sign-up for free resources from Google Drive (free resources - Google Drive) and model downloads. Well worth the sign-up.
- https://youtu.be/clzstqtN6YQ : Life saver video by Blender Base Camp on how to fix bad normals. I had this issue more than once and got stuck until I found this.
- Other videos illustrating how to perform certain tricks:
    - https://youtu.be/D0CPw_rvgHc : Curved text by blended.
    - https://youtu.be/oHkzyH9dpv4 : Curved objects by Gamemaker Game Programming Course
    - https://youtu.be/nBMWKxh84Tg : Sculpting, curved objects such as gravestones by Blender Tutorial.
    - https://youtu.be/xM2qXaU1NyM : Creating cracks and imperfects such as cracks in gravestones by CG Cookie.
    - https://youtu.be/EpZa0l_aCuA : Create awesome looking pumpkins by Ryan King Art.
    - https://youtu.be/HTHefemOMLM : Shows how to create Halloween pumpkins by Blender Cubed. I combined techniques from both previous videos for my Halloween pumpkins in Unity.
    - https://youtu.be/Bc52fcrHvAA : Cutting holes in mesh by Evermotion.
    - https://youtu.be/x_zNrolENjU : Realistic dirt particles by CG Cookie. I used this as part of creating my dirt hole for Unity.
    - https://youtu.be/_owcNpxp8h4 : Creating bumps, deformations in mesh using Displacement Modifier by CG Cookie. I used this as part of creating my dirt hole for Unity.
    - https://youtu.be/nYLWUAm2vqs: Creating a cave by Derek Elliott. I used this as part of creating my dirt hole for Unity.

Blender Modeling/Animation/UV Mapping/Export to Unity

- https://youtu.be/UAami_DhnTA : Multipart series by Imphenzia on creating an animated character in Blender for use in Unity. Very good tutorial to learn about modeling, animation and UV mapping trick using a small color palette. The only caveat is that this is 2D animation in place so does not work for 3D world to make a character move. You must incrementally move

the character forward with each step animation, so it moves in Unity. Other than that, everything applies. Very good tutorial.

- https://youtu.be/_C2ClFO3FAY : Good animation techniques given by CG Geek. He also provides reference to walking video for reference. https://youtu.be/HEoUhlesN9E by Kevin Parry.
- Baking PBR Materials to Textures from Blender to Unity | NODRAGEM GAMES and GitHub - danielenger/Principled-Baker: Blender Add-on: Bake PBR textures with a few clicks to automatically bake and export needed textures to Unity. I use it all the time.
- https://youtu.be/4JzmH1DpQYA : While not specific to Blender, good tips from Nick Senske for handling material/textures, etc. import to Unity.
- https://youtu.be/Ap4SoG-KfKM : Immersive Limit covers important techniques to properly set rotation, scale for models that are to be imported to Unity.
- https://youtu.be/wTzk9T06gdw : Very cool video on how to create marble effects by Ryan King Art.

Unity Set-up

I originally created my environment using 3D template. This eventually caused me issues because I wanted to implement Shader Graph stuff. It requires the URP rendering engine which is installed and configured as part of the URP template. So, if you run into the same issue, these videos cover the migration steps to move from 3D to URP.

- https://youtu.be/9CdIw87cH0g : By My Deep Guide. However, I still lost rendering for in-game video display.
- https://youtu.be/TvkvI2S1mp0 : Fixes video by fast tech.

Unity Introductory Tutorials

Some of these videos may be beyond newbie level but are very good to watch to get an understanding of the environment, techniques, tricks.

- https://youtu.be/pwZpJzpE2lQ : Good intro for newbies to Unity by Imphenzia. Covers player movement, physics, scripting and simple game objects.
- https://youtu.be/Lu76c85LhGY : This video by Jason Weimann shows how to build an Angry Birds like game and builds on what you learn in the previous video.
- https://youtu.be/RCUC5-lbb2g : Another good video by Jason Weimann doing a live tutorial of building a 3D game.

Unity Terrain

- https://youtu.be/ddy12WHqt-M : Making Terrain, dirt, grass, trees by UGuruz. Covers installing and using Terrain Tools and Terrain Tools Sample packages by Unity.
- https://youtu.be/MWQv2Bagwgk : Making more detailed terrain features including importing heightmaps.
- https://youtu.be/9rSP-ozPs0A : Good video on understanding Terrain and using a shader to change the texture of the terrain. By World of Zero. I started off with this to play with putting snow effect on the terrain. Good learning exercise and introduces important concepts.

- https://youtu.be/kmPj2GmoMWo : By Léo Chaumartin. This builds/expands the concepts above and creates a soup-to-nuts shader for the Terrain. I implemented this as-is in Sprague InSanity including the next video.
- https://youtu.be/bY7r6blL1K8: Excellent video by World of Zero to build a snow shader that covers the top of game objects. I took his concepts and plugged the snow coverage into the shader above to create a big ole shader for Terrain. I use this in SprageInSanity to create snow coverage for far off stuff like mountains, etc. I implemented my own crazy Create Snow C# stuff to create thick snow coverage in the immediate view of the player. This shader is awesome. But I had issues with grass, flowers still showing and bleed through of underlying Terrain colors. Also, any game objects with incorrect UP/LEFT/RIGHT coordinates caused snow to render on the side of the object. Thus, my C# stuff. Also, I wanted snow to have thickness on objects.
- https://youtu.be/z7kQpUZXXhw : Another video by World of Zero that goes into detail of how to build dynamic snow. This shows how to deform the mesh to make it look like tracks in the snow.
- I used the information provided by the following sources for manipulating the Terrain paint details (grass, flowers) at run-time. Created code to temporarily remove them during my snow coverage scenes.
    - Edit Terrain foliage/texture at runtime - Unity Answers : Excellent explanation by duck on Terrain data detail.
    - Can I modify grass or details on the terrain at runtime? - Unity Answers : Also, more information from duck.
    - Create/Remove Unity grass at runtime with C# | The Game Designers Journey (the-game-designers-journey.blogspot.com) : Code solution for the concepts above by Mads. For my code, I added cloning for the TerrainData object so the changes were not permanently done in case of crash or stop of game before resetting.
    - https://stackoverflow.com/questions/62442429/best-way-to-get-shape-of-terrain-at-1x1-size-for-road-system : derHugo gave a great sample on how to extract terrain height data so you can create a mesh to mirrors the terrain. I used this to build my code to build a terrain mesh, extrude it, and lay it on top of the existing terrain to give a nice thick snow look.

Unity Assets

- Free Footsteps System: From the asset store and gives a quick start to apply sounds to footsteps. FREE. Used it to build my footsteps FP player as well as First Person and Third Person movement controls.
- Flooded Grounds: Used many assets from this asset store package. Lots of cool props, buildings, etc. Great asset to get a head start on game object stuff to plop into a game environment. FREE.
- Zombie: Fully animated zombie. Great asset to get a head start on understanding how things work. Very detailed Zombie model including separated limbs. FREE. You can import this back into Blender to create other characters or types of animation. This has to be my favorite package since a lot of work was done and shared freely. Very, very cool!!!
- FREE Rigged Skeleton and Bone Collection: Fun stuff for Halloween themed props. FREE.

- Living Birds: Another cool asset store package that gives living birds in a scene. FREE. Used this to have birds attacking the Zombie from Flooded Grounds.
- Rock Package. Nice set of large rocks to decorate the landscape. FREE.
- Terrain Tools and Terrain Tools Sample Pack: By Unity and FREE. Use this to add SpeedTrees, grass, flowered grass.
- Unity Particle Pack: Good samples

Unity Concepts/Tips/Tricks:

- How to Rotate in Unity (complete beginner's guide) - Game Dev Beginner : This article John French is a very good introduction into rotation and difference between Euler Angles and Quaternions. Must have background to understand how to use the various tools to rotate a game object.
- Change Material and its properties at runtime: Unity Tutorial - Gyanendu Shekhar's Blog : Change material at run-time by Gyanendu Shekhar.
- https://youtu.be/eJEpeUH1EMg : Mesh generation basics by Brackeys
- Runtime Mesh Manipulation With Unity | raywenderlich.com : Excellent look at how a mesh can be manipulated by Sean Duffy. Good to read for concepts and information.
- https://github.com/renangalves/MeshExtrusion : Used this to understand the concepts of extrusion by renangalves. It a wired for a simple cube, but I was able to extend the concepts to more complex objects.
- https://youtu.be/BVCNDUcnE1o : Video introduction by Tvtig to his actual code. This an awesome set of code to show how to slice an object (mesh) and create two objects. Very cool stuff. I used the code and concepts to extract a piece of mesh I identify as top for my Create Snow C# stuff. Code location in video description and here at: GitHub - Tvtig/UnityLightsaber: A lighsaber modelled in blender and brough to life in unity
- Rounded Cube, a Unity C# Tutorial (catlikecoding.com): Excellent tutorial by Catlike Coding for understanding how to create procedure cube and how to smooth it. Covers important concepts such as Clockwise Triangle creation so that the faces render correctly with the correct normals. This tripped me up early on. I started with the video to learn how I can build new faces on my extruded object. Word of caution: this tutorial has a lot of built- in assumptions and shortcuts for the smoothing aspect. It relies on the fact you that have a known cube with known sizing, dimensions and local coordinates starting at 0,0,0. So, don't try to apply the smoothing stuff in this video to other objects. But the concepts introduced are invaluable.
- MeshSubdivision - Unify Community Wiki (unity3d.com) : Subdivision algorithm by realm_1. Used this code to build my subdivision C# routine that goes into my smoothing routine.
- Smoothing code examples I found. I used the concepts from all these guys to write mine.
    - MeshSmoother - Unify Community Wiki (unity3d.com) : Implements Laplacian algorithm. By MarkGX. Lots of looping, but the concepts apply.
    - GitHub - mattatz/unity-mesh-smoothing: Mesh smoothing algorithm for Unity. : Another implementation of Laplacian algorithm by mattatz
- https://gist.github.com/unitycoder/a9ca74a8f84162ab64b6: Sorting algorithm to sort vector2 by unitycoder. I used this for many key sorts, etc.

Unity Player Controls

- https://youtu.be/_QajrabyTJc : First Person movement by Brackeys.
- https://youtu.be/BTxqErZ1JqE : Flashlight mechanism by Jimmy Vegas

Unity Animation/Collision/Triggers

- https://youtu.be/Bc9lmHjqLZc : Good detailed look at how to handle collisions(colliders) vs. triggers by Code Monkey.
- https://youtu.be/ReauId6jFFI : Using Animator to control different animations of character by Single Sapling Games
- https://youtu.be/Xx21y9eJq1U : Very old video by Unity on the Animator tool and capabilities. But, worth a look to show how things work conceptually.

Unity Enemy Controller

- https://youtu.be/4Wh22ynlLyk : Press Start shows how to make enemies follow the player.

Unity Day/Night Cycle

- https://youtu.be/33RL196x4LI : Very good tutorial to create a complete day/night cycle with sun and moon. Very configurable. This video is by Zenva. This is used as-is in Sprague InSanity to simulate night or day.

Unity Precipitation – Snow, Rain, Fog (Particle System)

- https://youtu.be/hyBbcFCvDR8 : Awesome set of assets, samples and tutorial by Unity on using the Particle System. The assets package installation is worth it (https://bit.ly/particlesystems-assets). They have fire, flames, rain, rockets, etc. Very cool stuff. I used this to create snow and rain.
- https://youtu.be/Ph3FvxJJ8AA : Cool use of particle system. By Gabriel Aguiar Prod. I used the Unity stuff above but added his splashes for rain.

Unity Clouds ,Flags, Water(Shader Graph)

- https://youtu.be/pzo4mitkY8k : Making a flag wave with Shader Graph by Unity.
- Introduction to ShaderGraph - Unity Learn : Introduction to Shader Graph to create a waving flag covered in video above.
- https://youtu.be/oV59qJYTlok : Makes flag waving more realistic motion by Walking Studios Project.
- https://youtu.be/gRq-IdShxpU : Making water feature by Unity. I could not find the asset package but followed tutorial to help make lake in SprageInSanity.
- https://youtu.be/xxhvUyvIH6s : Clouds by Gabriel Aguiar Prod. Used in SprageInSanity.

Unity Game UI

- https://youtu.be/-2Z7UzhYyTA : GameGrind shows how to create a game UI for interaction during gameplay. I used this to create my crude UI for setting weather, time of day, snow, precipitation during my in-game walking around demo.