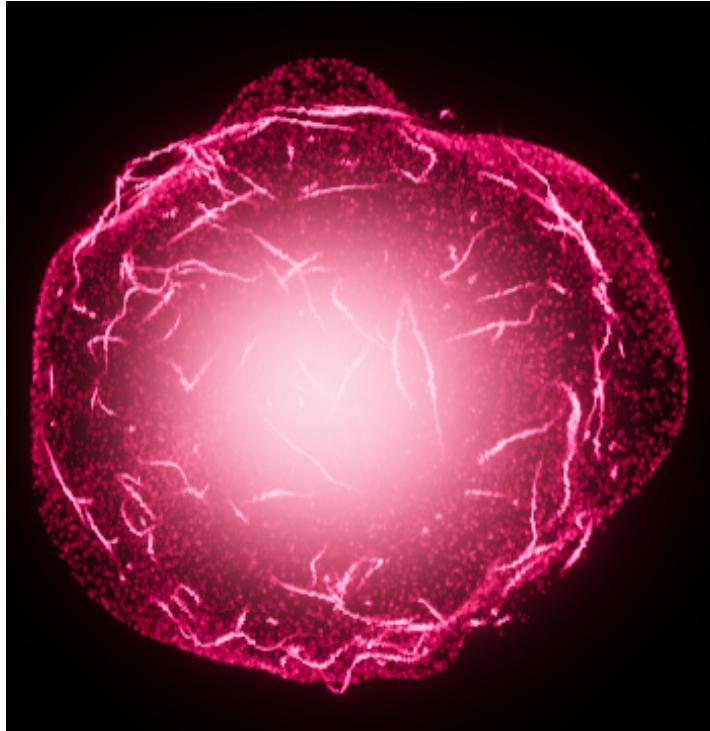


Unity VFX Graphs

Personal Learning + Tutorial Process



Disclaimer/Introduction

This is an informal document of my research into VFX graphs that is meant to serve as a place for me to document my findings for ease of reference and to effectively track my learning process with this Unity tool. This repository can be found [here](#).

The inspiration

One of the first things that got me interested in learning VFX graphs was having Unity's video titled [Unity Visual Effect Graph - Spaceship demo](#) shared with me. Having only used Unity's default particle systems up to this point, seeing these gorgeous effects struck me, which is what has driven me to start learning more about it.

The following are some screenshots I took from the video on VFX graph effects which impressed and inspired me the most.



The research process

The first portion of my research will focus on understanding what VFX is alongside Unity's Visual Effect Graphs, and finding existing examples and tutorials on VFX graph effects. This will allow me to gather material that I can follow and use for this learning process. Once I understand how

to manipulate and create these graphs, I will try and create custom effects that either use some or none of the tutorial material.

What are Unity VFX Graphs?

Resources:

- [Create beautiful and complex effects with the Visual Effect Graph: Verified in 2019.3](#)
- [Visual Effect Graph | Visual Effect Graph | 8.2.0](#)

First of all, Unity's Visual Effect Graphs is a node-based way of creating and designing a wide range of things, including simple and common particles, to more complicated ones. What VFX manages to achieve is that it allows for the creation of complex and common effects in an easy and user friendly way. As a result, despite making this a viable tool for artists, it also makes this a viable tool for my own learning of Unity's VFX graphs. Visual effect graphs can also be created as Visual Effect Asset which makes it reusable.

Why I should learn VFX Graphs for creating effects

Resources:

- [Creating explosive visuals with the Visual Effect Graph](#)
- [Create beautiful and complex effects with the Visual Effect Graph: Verified in 2019.3](#)
- [Introduction to the Visual Effect Graph | raywenderlich.com](#)

Although particle systems can be used to achieve similar effects, VFX graphs are appealing as it has an intuitive and easy to use interface for designing beautiful particle effects, something that I feel could greatly contribute to my future projects as up to this point my projects have only ever used the default particle system which are sometimes difficult to properly customize for specialized effects.

Based on [John O'Reilly's article](#) and the image he included which gives a quick breakdown of key differences between the particles, the primary and most important difference is that VFX graphs use the GPU, contributing to shader calculations, while the particle system uses the CPU for simulated effects.

CPU vs GPU Particles		
	CPU Particle System	GPU Visual Effect Graph
Particle Count	Thousands	Millions
Simulation	Simple	Complex
Physics	Underlying Physics System	Primitives, Depth Buffer, Scene Representation (E.g. SDF)
Gameplay Readback	Yes	No*
Other	-	Can read frame buffers

This fundamentally changes how particles work and interact with the rest of the Unity environment, as GPU particles won't be affected by the physics engine, while the default particle system does. Overall the Visual Effect Graph may be more performance intensive, but it opens

up for a lot of potential when needing to create advanced and creative effects that don't rely on scene physics.

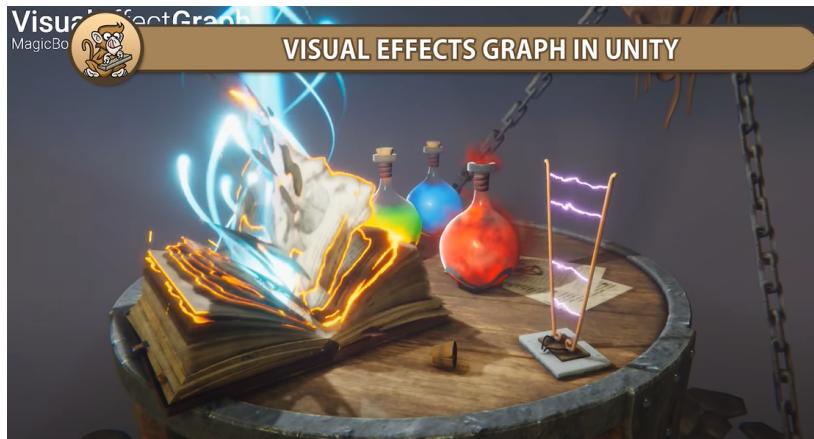
Although there are clear benefits to learning how to create VFX graphs, the authors of these sources, as well as many users across different discussion threads, make it abundantly clear that it is important to carefully choose when you implement either system, as it will not always be necessary or plausible to use one over the other.

Setup & Introduction Tutorial

Resources:

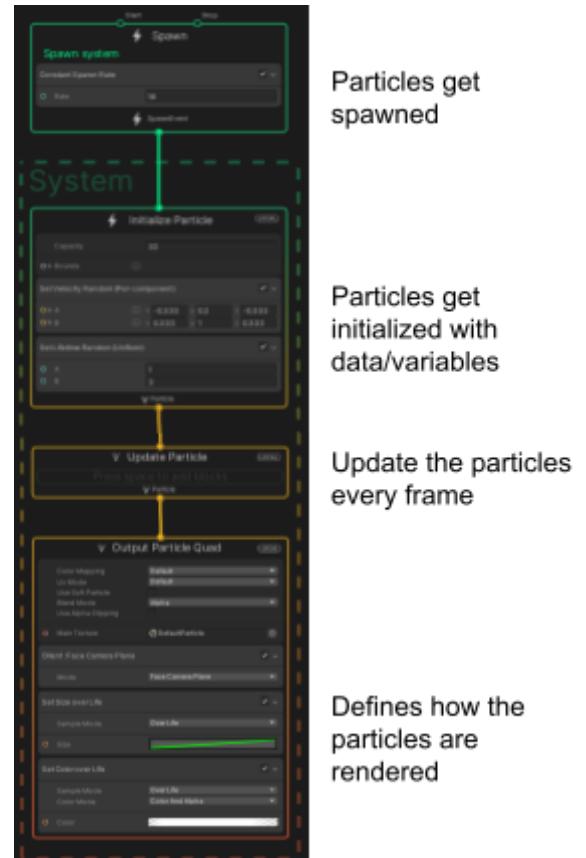
- [Make Awesome Effects with Visual Effect Graph in Unity!](#)
- [MILLIONS OF PARTICLES! - Unity VFX Graph](#)

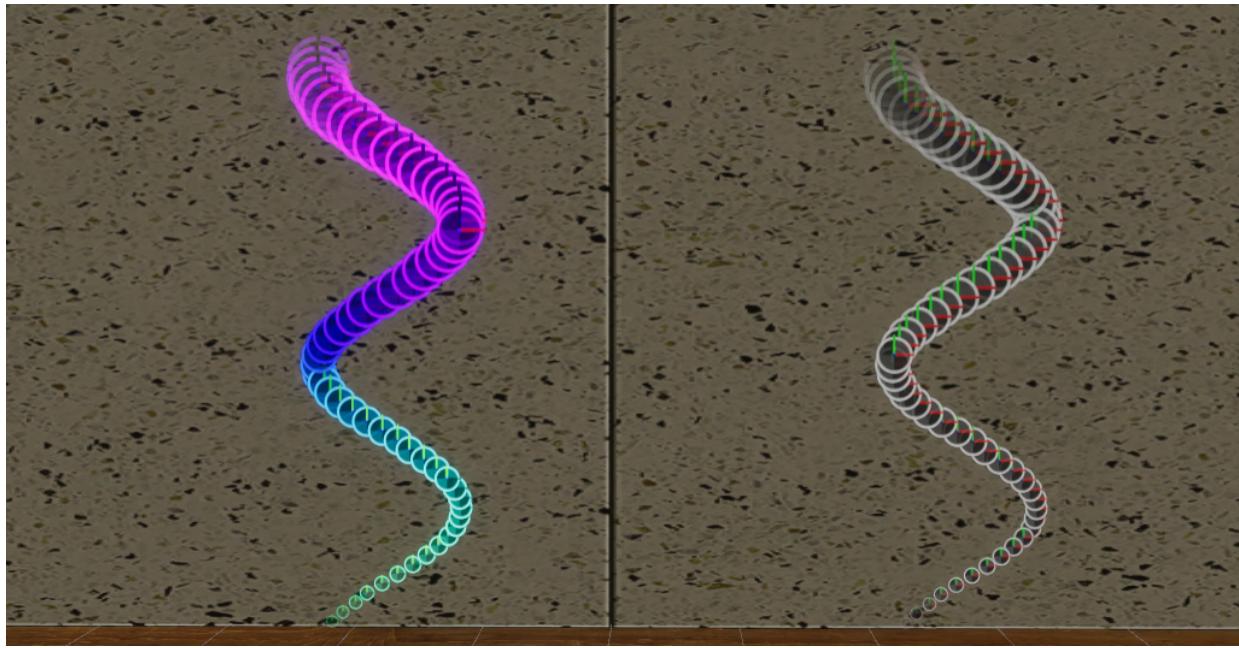
Other than using Unity's [Introduction to the VFX Graph](#) tutorial to begin setting up my work, I also decided to follow another interesting introduction and tutorial by the online user [CodeMonkey](#), a professional indie game developer who creates freely available tutorials on a wide variety of Unity features and functionalities.



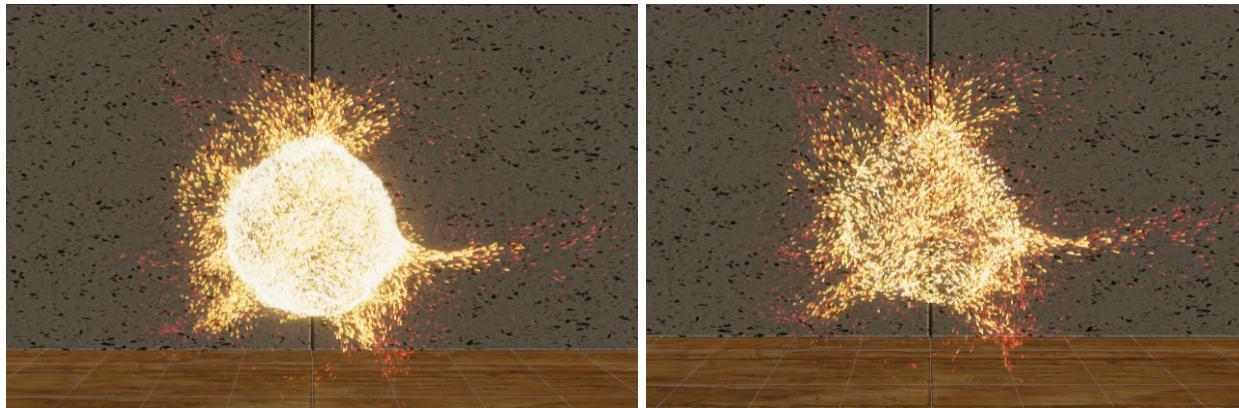
This tutorial proved to be very useful in learning and understanding how to set up your own effects, as well as with understanding how to use this feature's effect design interface. The primary thing this taught is how to add or remove different blocks across the contexts, expose variables and how to connect these nodes to each other. Below is the basic structure of the default VFX graph with some summary of what they do:

By the end of this tutorial I was able to create the following effect. This is created by using a time based sine input for the starting position when initializing the particles, and then by applying a color gradient to the output quads color.





Another quick introduction tutorial that I decided to reference and follow was one created by Brackeys, another popular professional developer who creates freely available tutorials. Although he covered most of the same introductory information, some extra insight was shown into some other small VFX graph functionalities. One of the interesting effects that was shown was how to create a simple pulsating effect. In addition to the pulsating effect, I applied some extra nodes just to see how I could manipulate it.



Smoke and Fire Tutorial

Resources:

- [!\[\]\(36f8637baaa56c4be44b454435949289_img.jpg\) FIRE AND SMOKE with Unity VFX Graph!](#)
- [Free VFX Image Sequences & Flipbooks](#)

The next tutorial I looked at was Brackeys fire and smoke tutorial, and with the use of extra assets from [Thomas ICHÉ](#), managed to create some interesting effects.

Despite being an easy to follow tutorial, one issue I encountered was with some of the particles taking on a green color. Fixing this issue proved to be more difficult than hoped for, as most suggested fixes did not work, or referenced settings that no longer existed, as Unity has been making big changes to where these types of things may be found. However, eventually I found a fix that suggested adding in a new diffusion profile and linking that up to the VFX effect, allowing it to appear as intended.



The following image is the end result of me completing and following this tutorial, which includes the customizable smoke particles, as well as some sparks which could be below a fire.

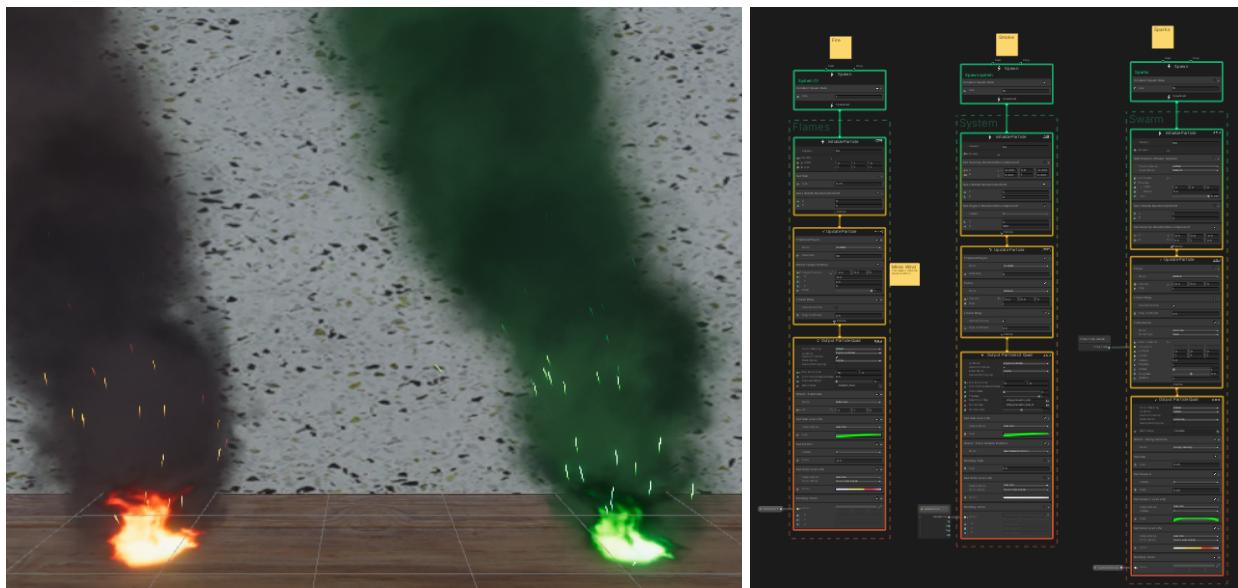


Fire Effect

Resources:

- [Free VFX Image Sequences & Flipbooks](#)

After having created the smoke effect, I felt like it could be further improved upon by adding an actual flames effect below the smoke, so that I could try and create something without the help of tutorials. So after having gotten an effects flipbook of flames from [Thomas ICHÉ](#), I began experimenting on creating a flames effect. Although it took a little bit of experimentation to get the flame to work correctly, I managed to create the fire effect below the smoke and added in some more exposed parameters that made it easier to change the color of the flames, sparks and smoke. The following images show the results of this work as well as how the VFX graphs were structured.

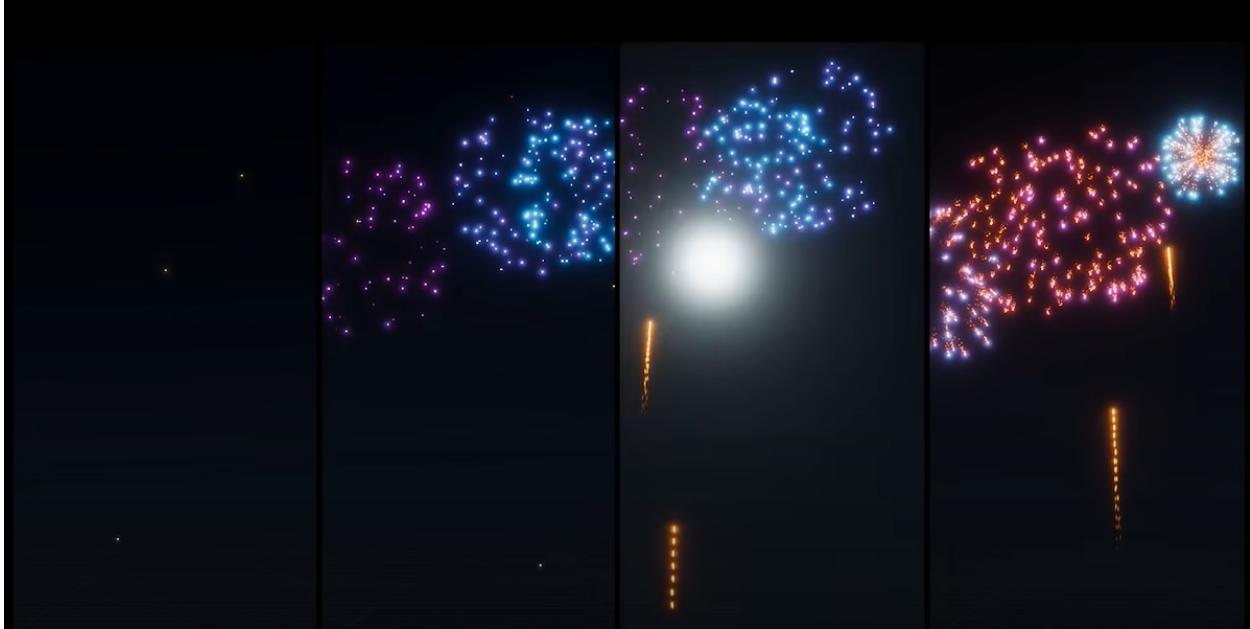


Fireworks Tutorial

Resources:

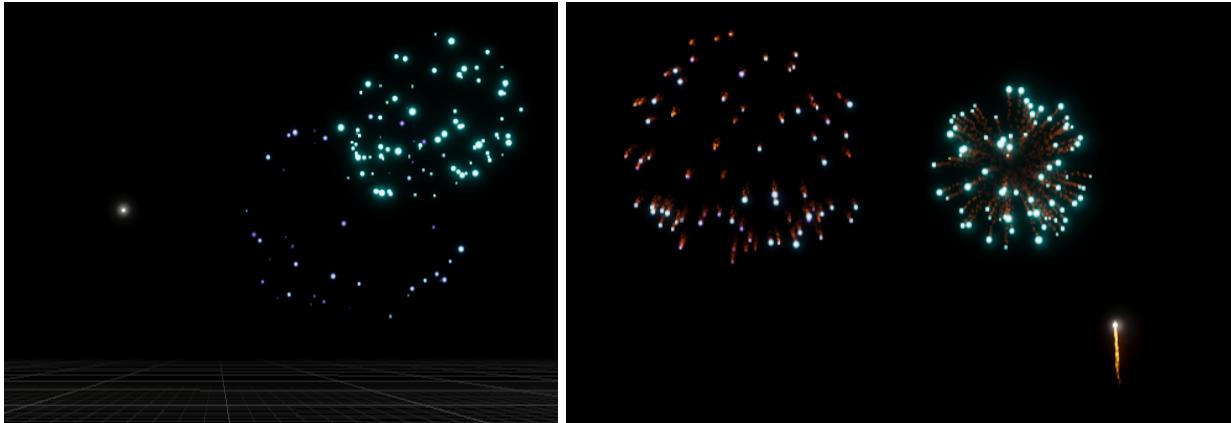
- [FIREWORKS in Unity using VFX Graph!](#)

The next tutorial I looked at was Brackeys fireworks tutorial, which was easily broken down into the following four primary stages: the rockets, the explosion, trails on the rockets, and then trails on the explosion particles. The following image from Brackeys tutorial visualizes this.



Overall this tutorial proved to be very insightful and useful as it serves as an easy way to practice various VFX features in a very dynamic way. Alongside this, one of the important things that it teaches is how to use events within the effect graphs, allowing for specific triggers for other parts of the effect.

The following is the result of me following this tutorial. The first shows the rocket (white) and explosion (blue/purple) without their trails, and the second shows these with the trails.

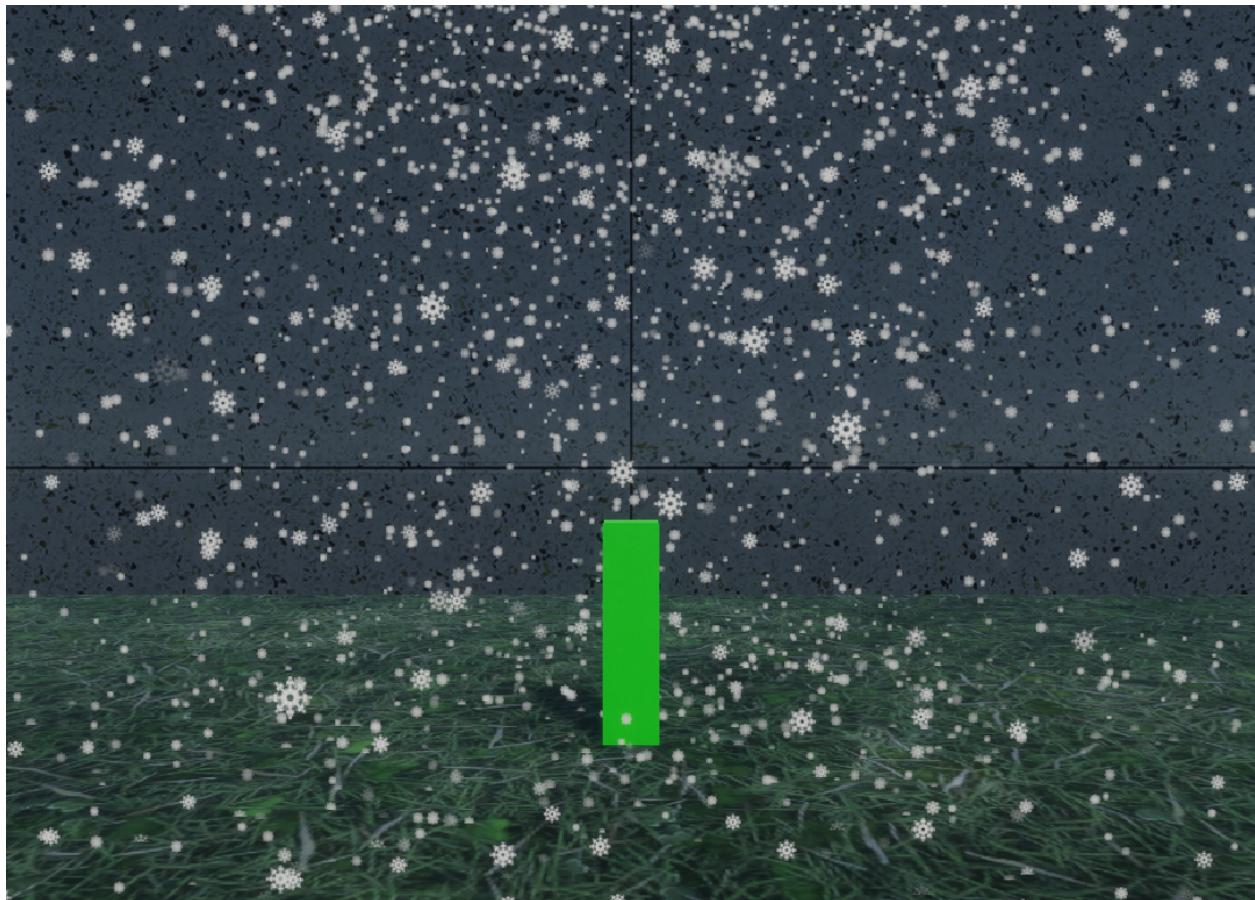


Fireworks Tutorial

Resources:

- [Snowflakes. Sprites | 2D Textures & Materials](#)

After having followed these tutorials, I decided to try and create a simple effect without the help of tutorials, as I felt more comfortable with the node based creation process of VFX graphs. After a bit of thought, I decided on creating a snow effect. It slowly falls from the sky and will settle on the ground before disappearing. The radius of this effect, the starting height and spawn rate of the snowflakes is all customizable from the inspector. Although this took a little bit of time to create, I am quite happy with the end result, as shown in the following image.



Weapon Effect Tutorial

Resources:

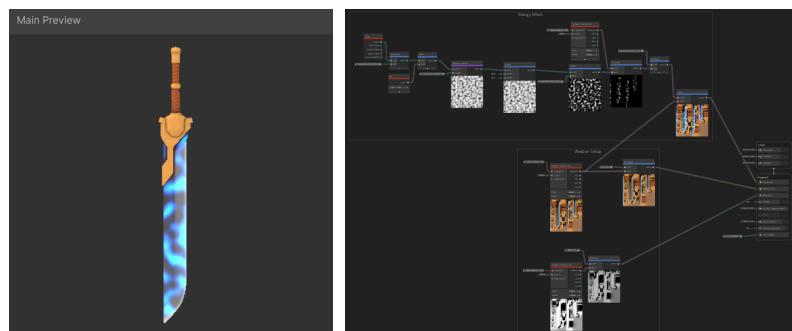
- [Unity VFX Graph - Weapon Effect Tutorial](#)
- [SciFantasy Sword Pack | 3D Weapons](#)

The next tutorial I decided to follow was a weapon effect tutorial created by Gabriel Aguiar Prod, which creates both a custom shader for the weapon, as well as a VFX particle for the blades of the swords. In order to achieve the best results, I decided I would also use the weapons pack that Gabriel originally used for this tutorial as well, so that I could properly compare my results to his results.

Overall this tutorial was very insightful as it not only taught me more interesting features about VFX graphs, but also introduced me into shader graphs and how these can be mixed with VFX graphs to create interesting and exciting visuals. The image on the left is what Gabriel Aguiar had created during his tutorial, meanwhile the image on the right is what I managed to create.



Despite having created the shader effect, it did not appear to function in the game scene and so this will be a part of future research into shaders and how to correctly implement them.

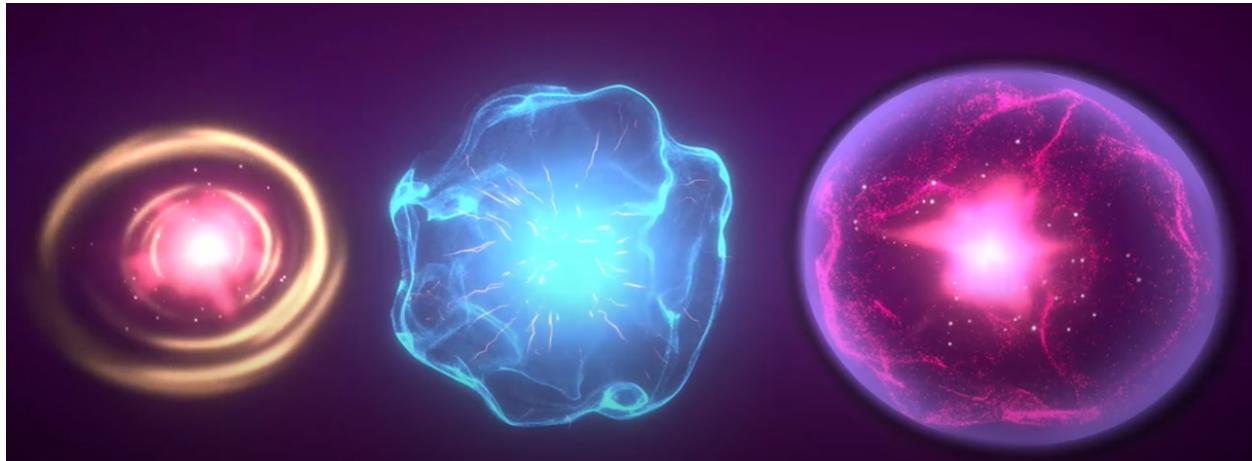


Magic Orb Tutorial

Resources:

- [Unity VFX Graph - Magic Orb Effect Tutorial](#)

The next tutorial I decided to follow was a magic orb tutorial created by Gabriel Aguiar Prod. The following image from this user's video shows different variations and versions that they have created, however this tutorial only covers how to create the centermost orb.



Overall this tutorial proved to be very useful, as it taught me about other functions that the VFX graph offers, such as perlin curl noise generators and how to creatively apply them to things such as position and turbulence, allowing for interesting and waving effects. The following images are what I managed to achieve following this tutorial:

