

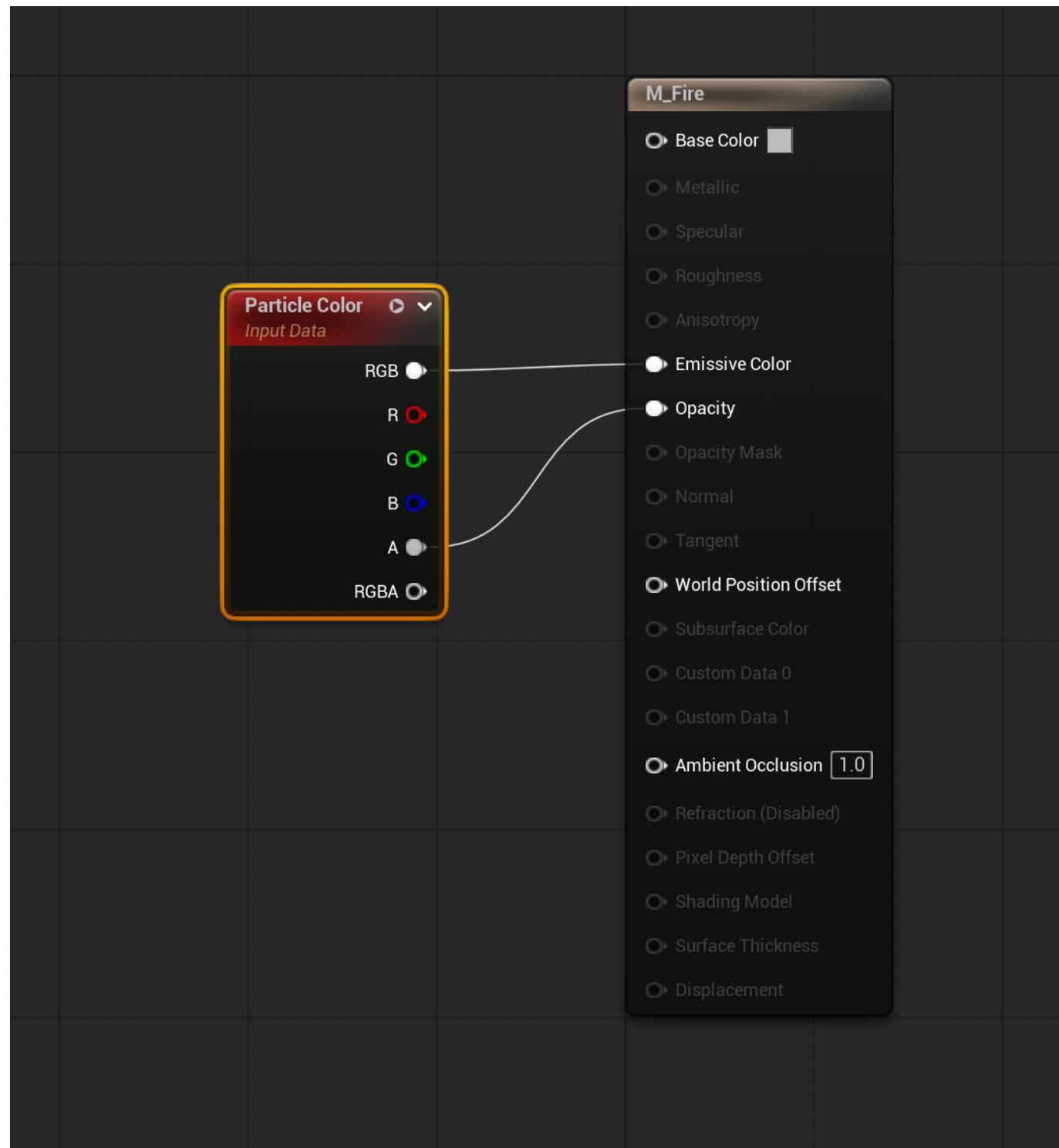
Week 5 Task – Niagara VFX Systems

Callum Wade 2404781 28/10/25

1. Fire

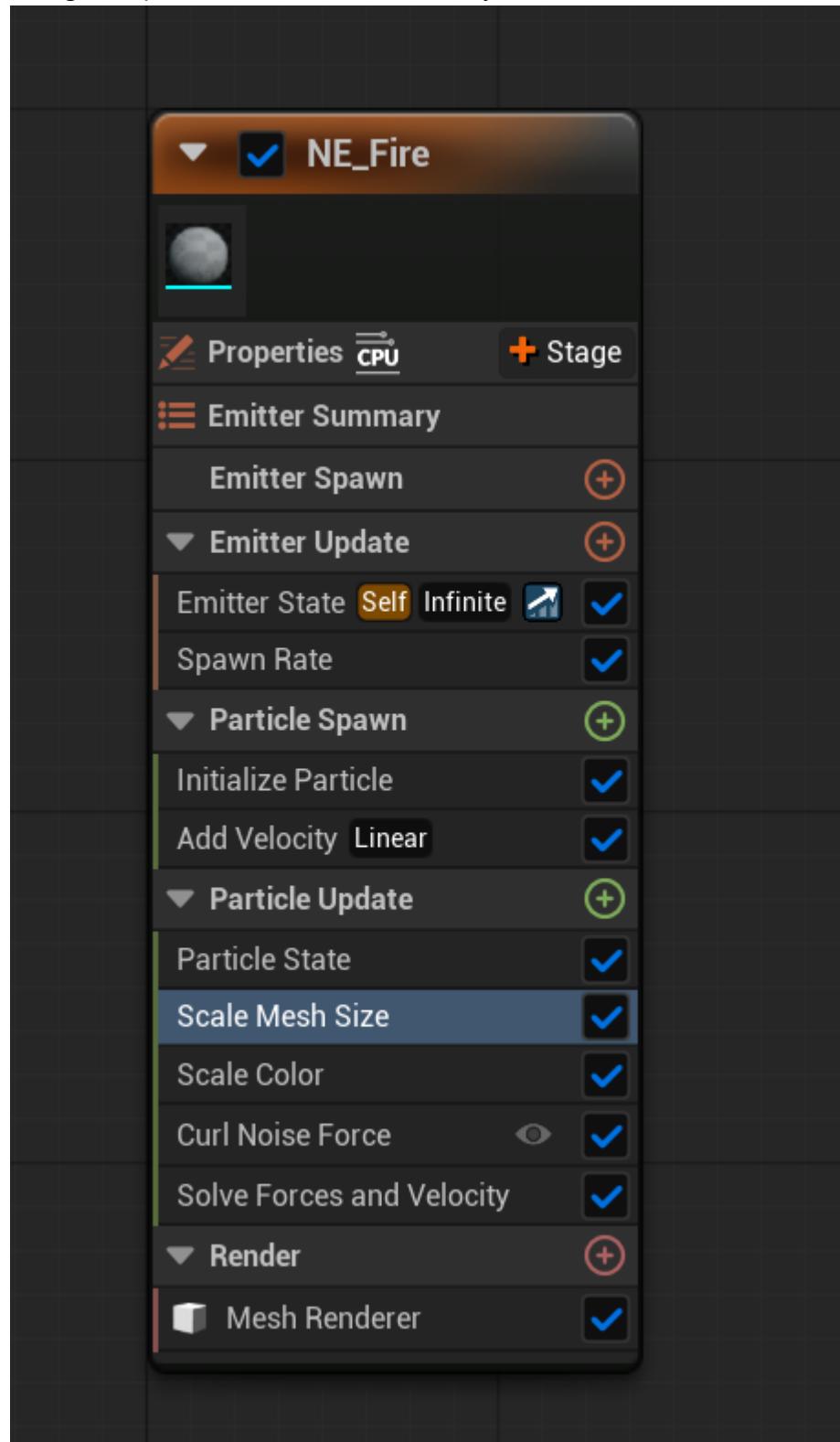
Development Process

Fire Material



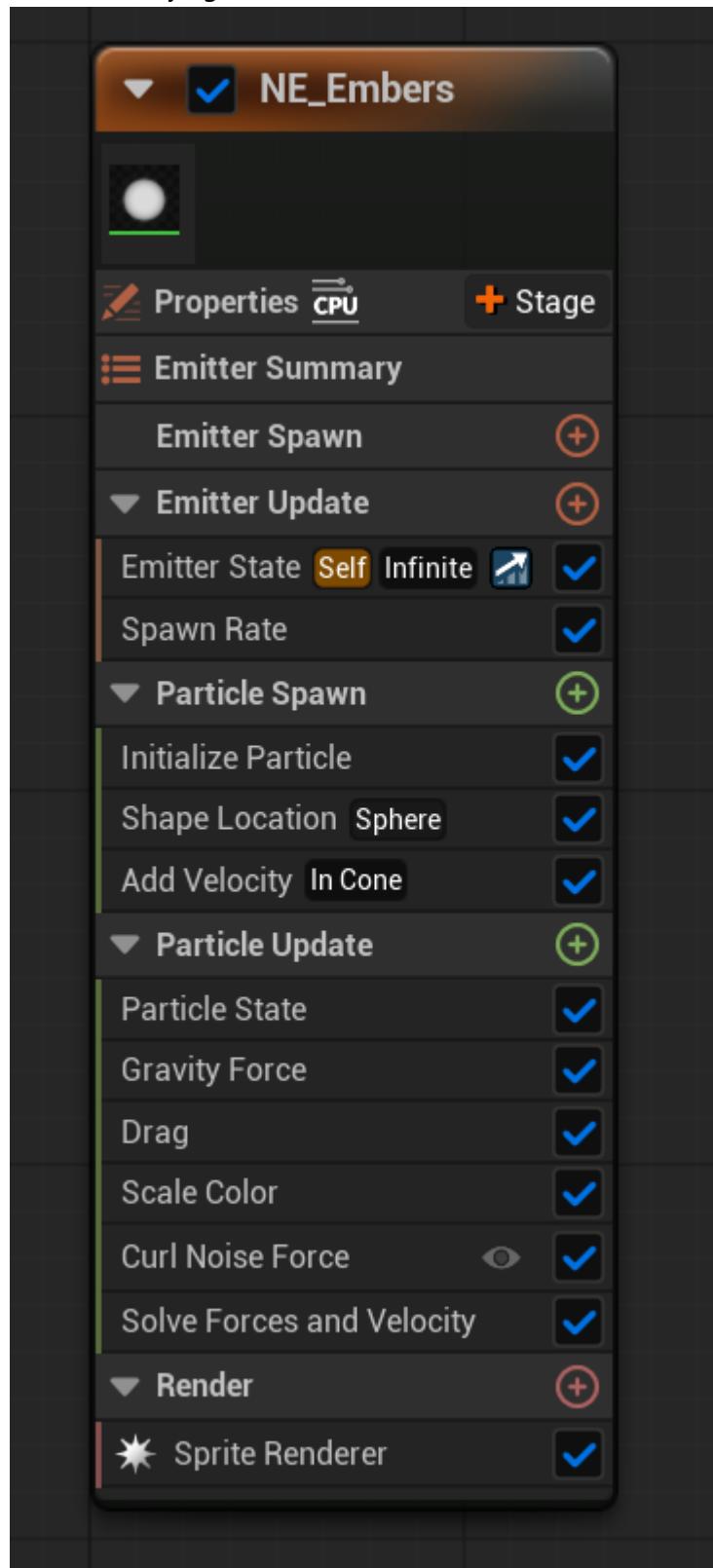
NE Fire

To make the fire work as intended, a spawn rate module and a velocity module were required. The spawn rate controls how often particles spawn over, while setting the Z value of the velocity module to 250 causes the particles to rise upwards. To add variation, the initialize particles module uses randomised lifetime and mass values, and the mesh scale is set to a random range. The only property that isn't randomised in this module is the colour, which is set to a bright red with a red value pushed to 100 to make the particles glow. To stop the effect from looking like identical blobs floating upwards, scale mesh size and scale colour modules are used to change the particle's size over time till they fade out.



NE Embers

Add velocity is also the main module within the embers effect, but this time it works alongside gravity force and drag modules to create particles that rise with an initial upward velocity and then get pulled back down, simulating falling embers. To add randomness to the effect, the lifetime, mass mode, sprite size, and velocity are all set to random values. This ensures that the particles vary in size, last different lengths of time, and fall at different locations depending on the velocity and mass they were set. To give the impression that the embers are dying out, a scale colour module is used to show the colour fading away from each particle.





Niagara System Parameters

NS Fire

Final Result

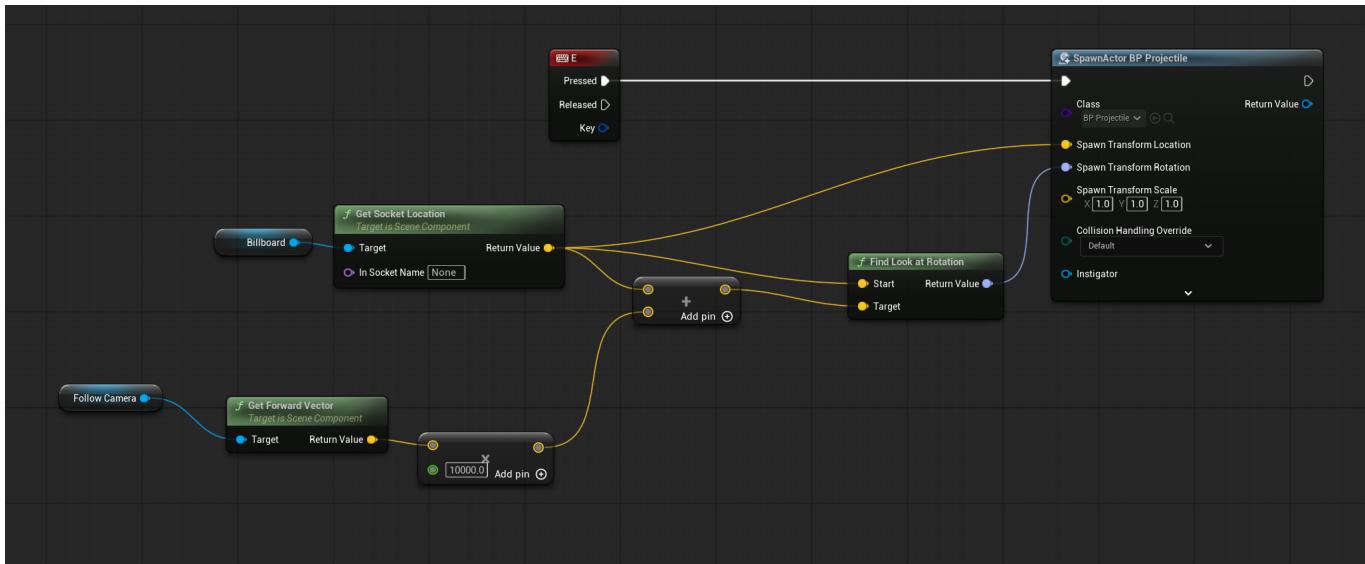
[Fire Showcase](#)

2. Fireball Projectile

Development Process

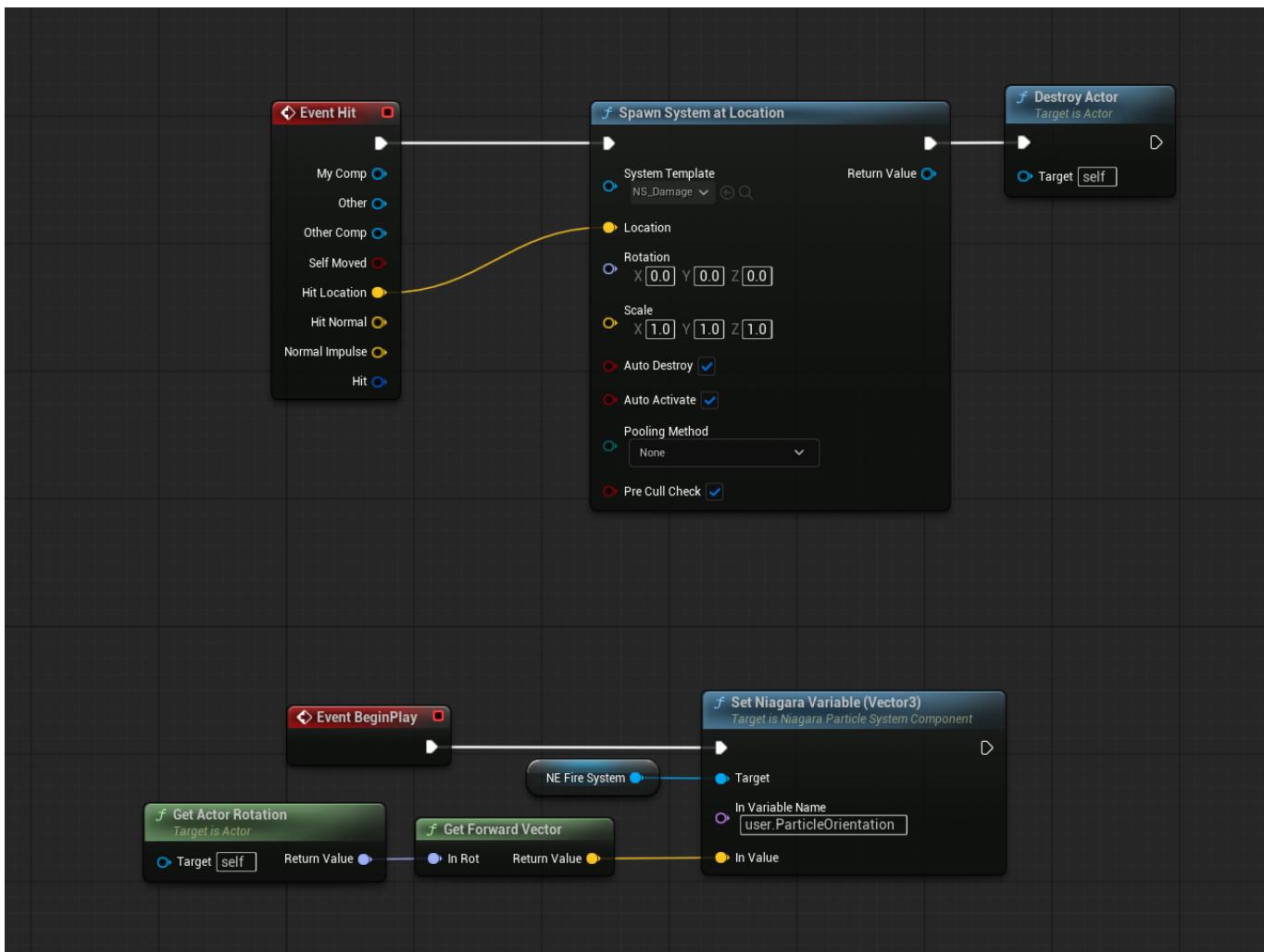
Projectile Spawn

What does it do? This blueprint is set up to spawn a fire projectile when E is pressed, aiming it in the direction the camera is facing and spawning it at the billboard component.



Projectile Particles

What does it do? This blueprint sets the direction of the fire projectile and also spawns a particle system wherever the fire projectile hits.



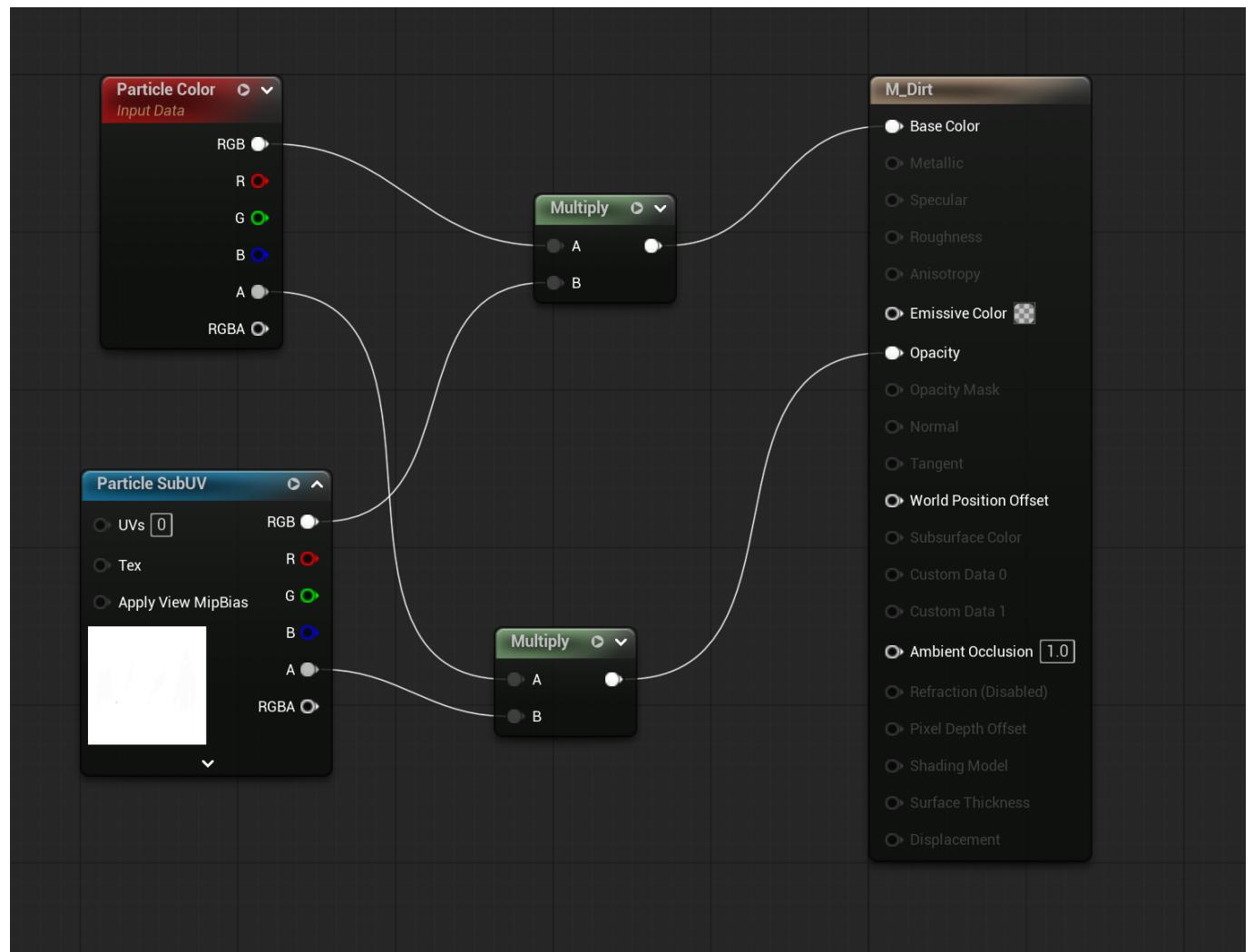
Final Result

[Fireball Showcase](#)

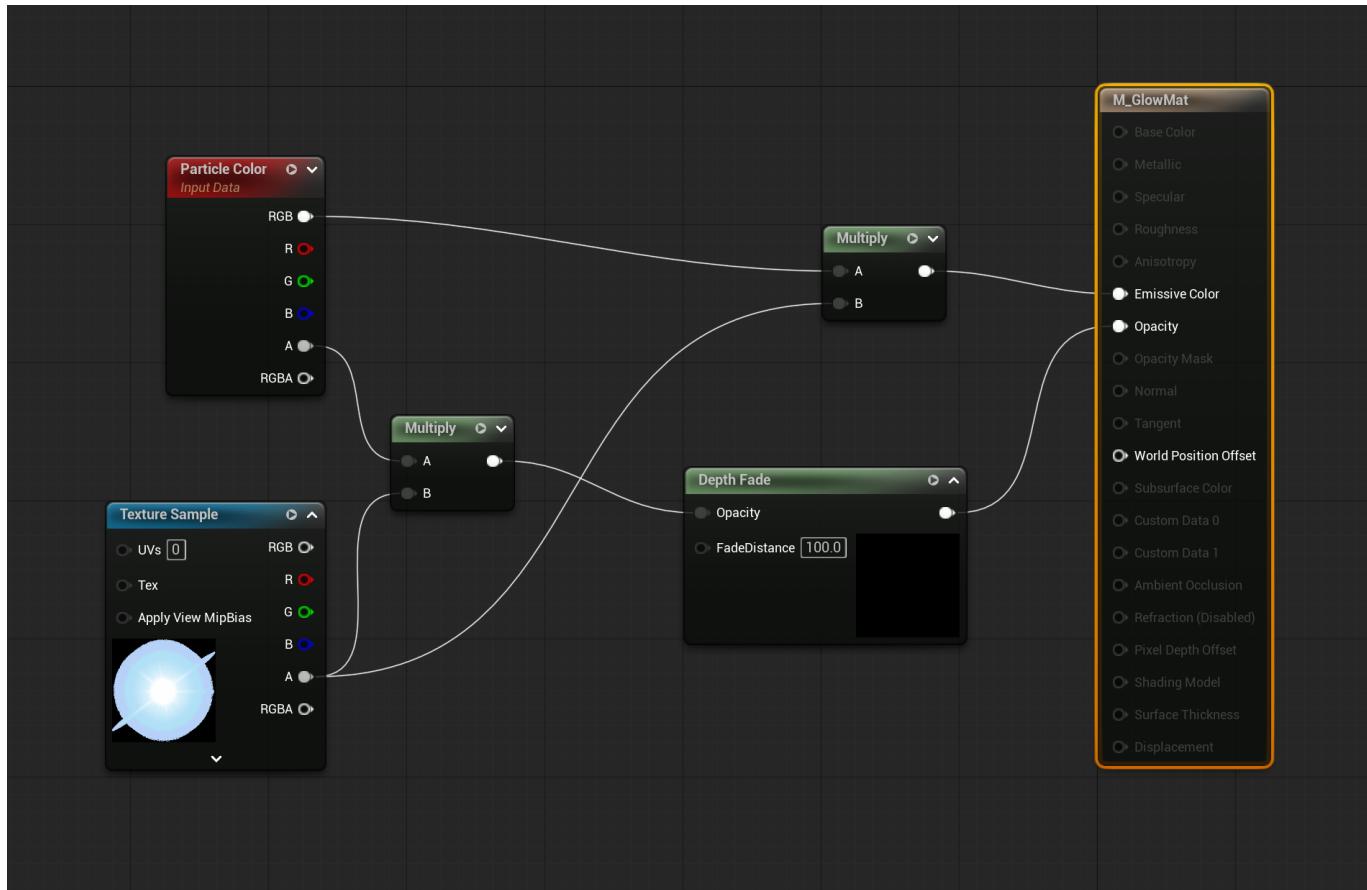
3. Lightning Strike

Development Process

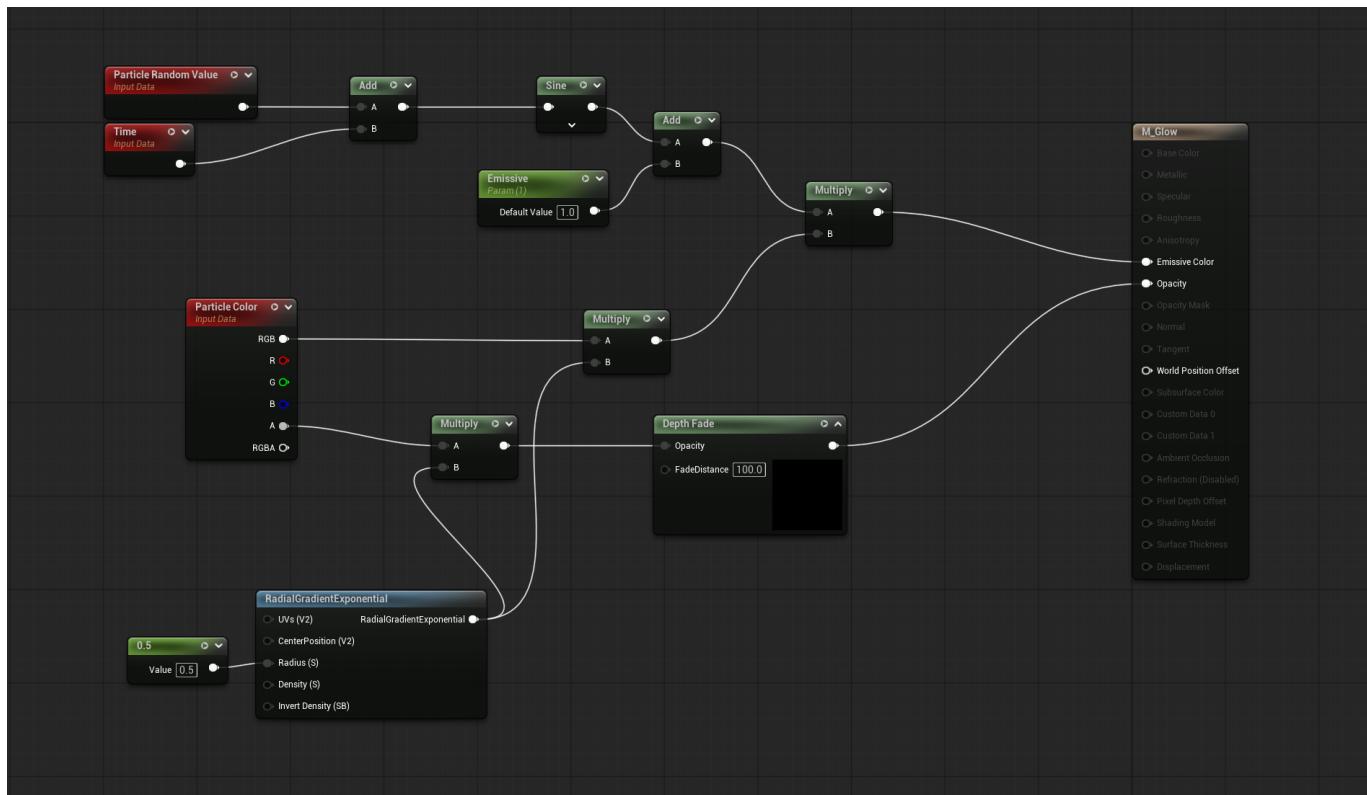
Dirt Material



Glint Material

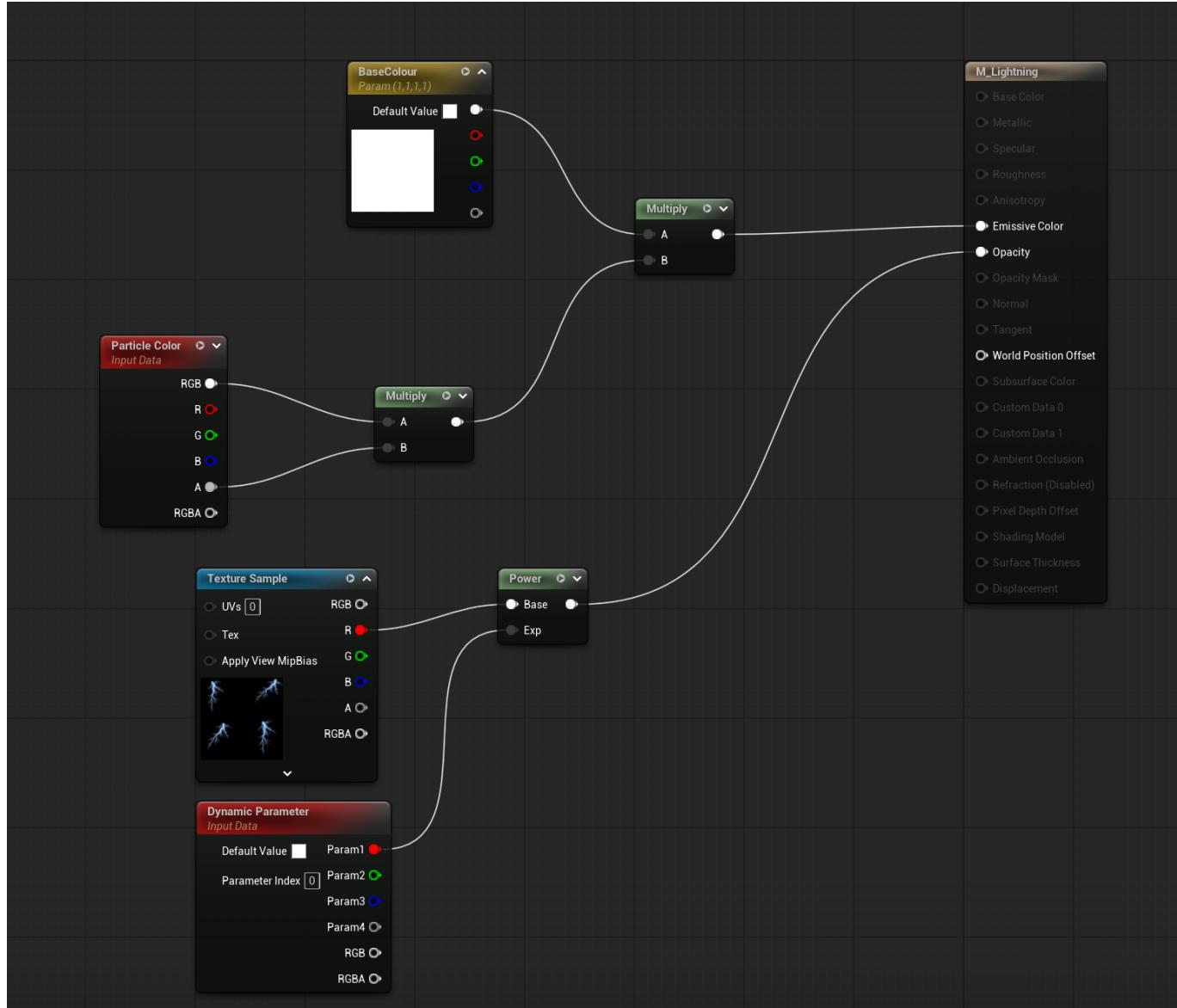


Glow Material



Lightning Material

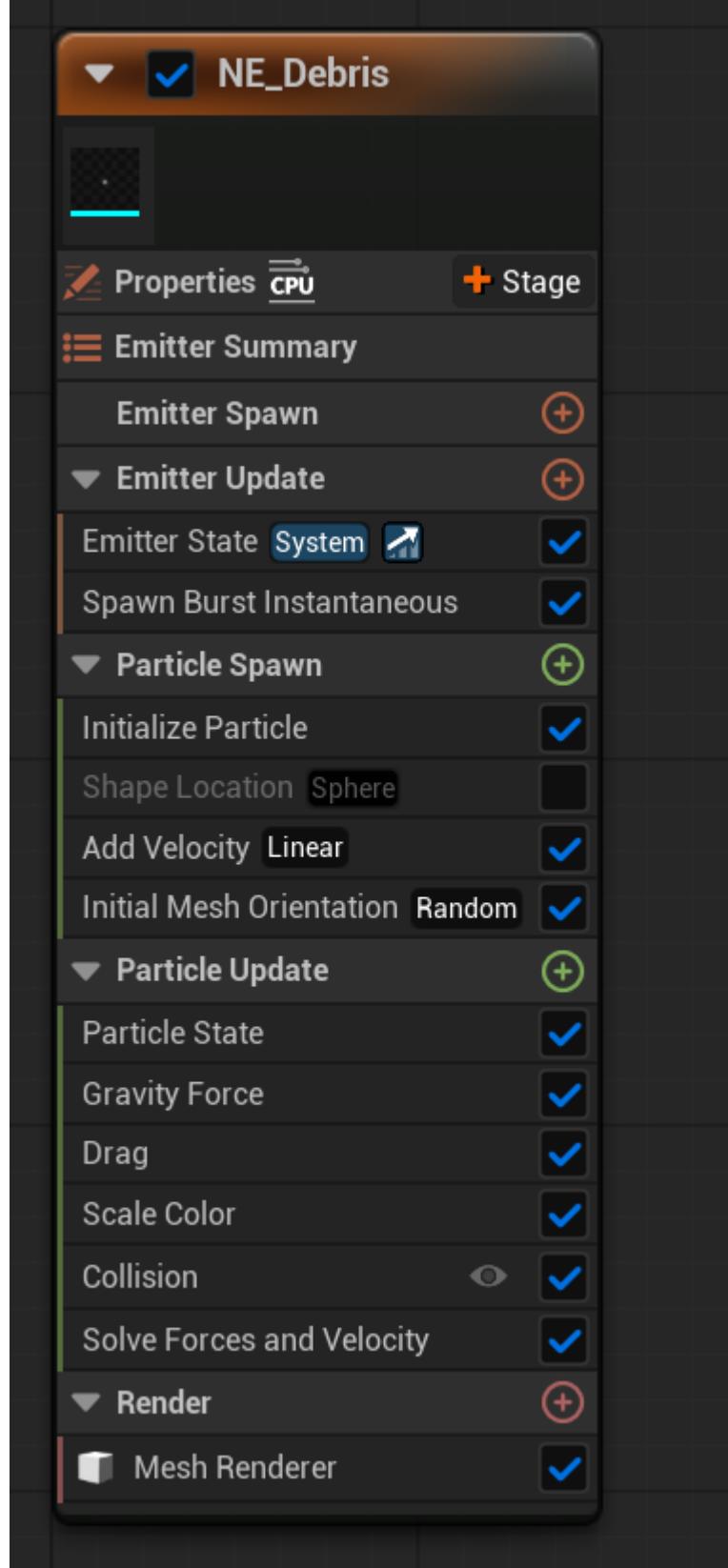
The lightning material uses a texture that I created in Photoshop by combining several lightning PNGs into a single image, which is then used in the particle effect.



NE Debris

Because the debris only needs to spawn once, a spawn burst instantaneous module is used to spawn the particles only once. To add variety to the number of particles that spawn, a random range is used with a minimum of five particles to a maximum of twenty particles. To cause the particles to burst out in different directions, an add velocity module was used. The add velocity module causes the particles to burst outwards by using linear velocity mode with a random range that causes the particles to fly a random distance away. To then pull the particles back to the ground and keep them on the ground, gravity force, drag, and collision modules were used. The gravity force and drag slows the particles and pulls them towards the ground, which then the collision module is used to stop the particles falling through the ground. Within the initialize particle

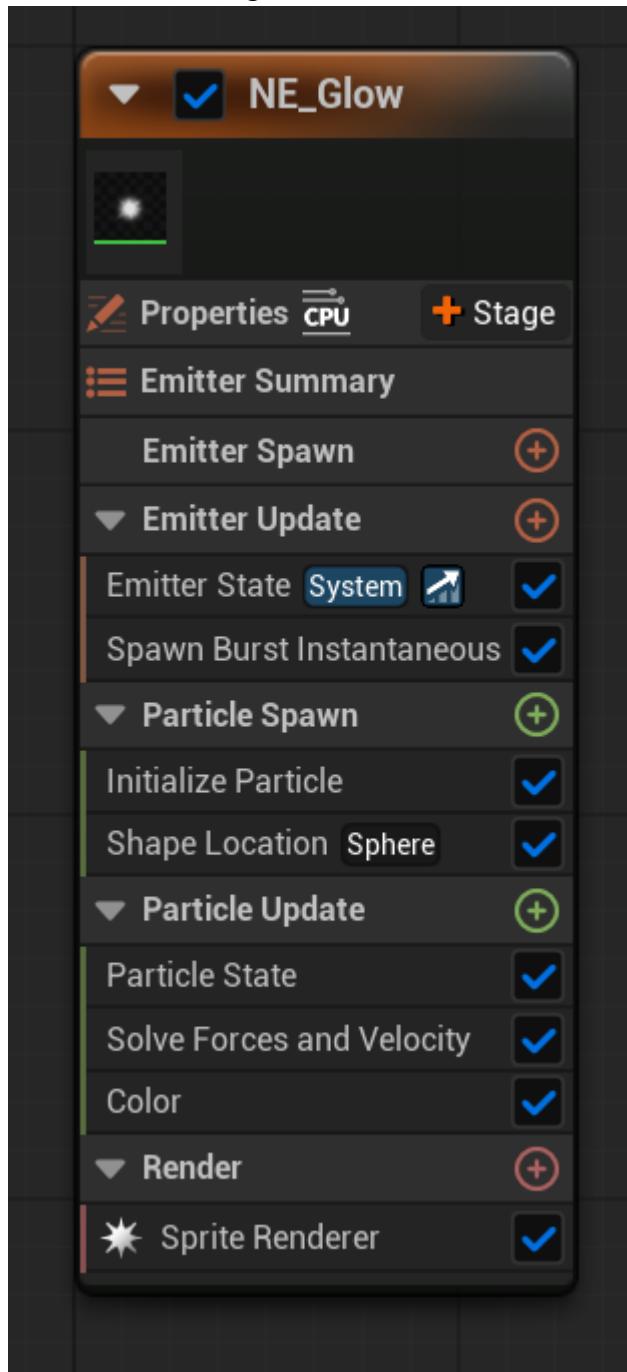
module, the lifetime, mass mode, and mesh scale is set to random to add variety throughout the particles.



NE Glow

To create a glow effect, the glint material is spawned as a quick flash. This is done primarily by using the spawn burst instantaneous and the initialize particle modules. Within the spawn burst instantaneous module, the spawn count is set to a random number between three and six to spawn a random amount of the glints in one go. The lifetime mode, mass mode, and sprite size are all set to random in the initialize particle module to add variety to each glow effect. The final important module used in the glow effect is the colour module

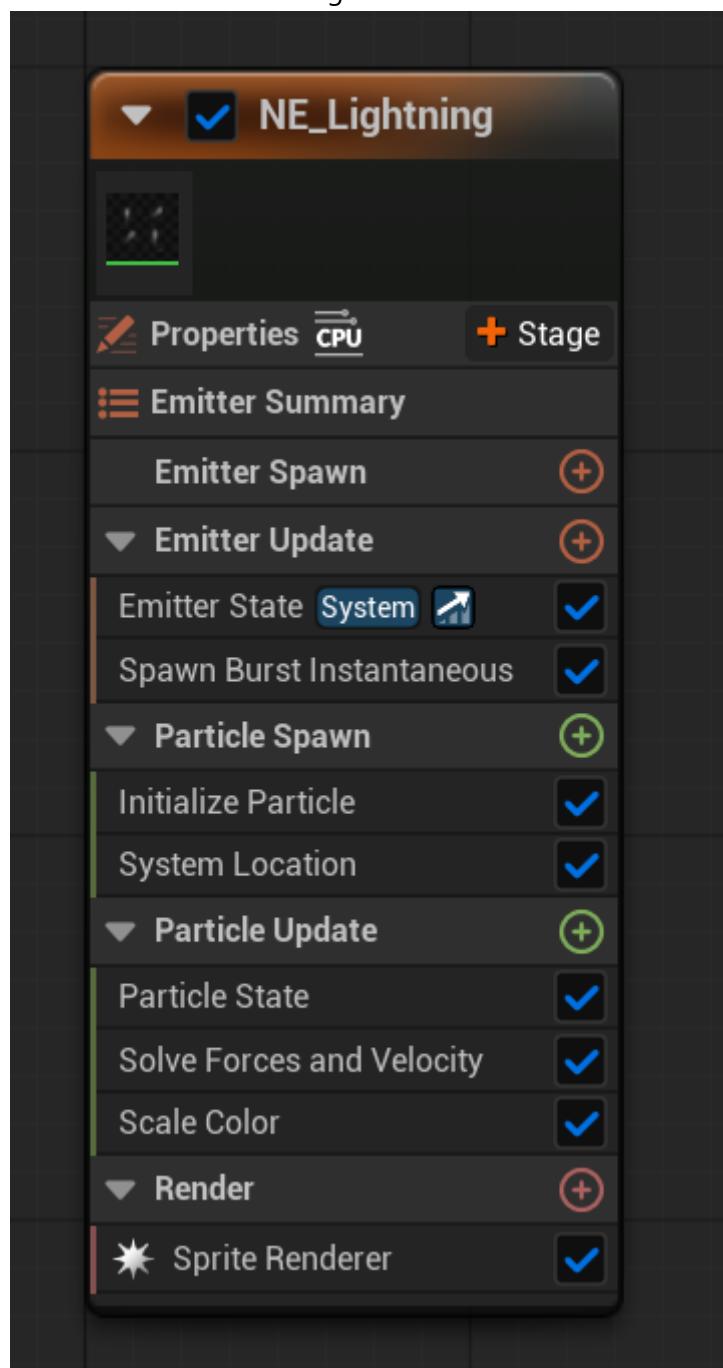
which is set to a bright, metallic blue to create a blinding effect.



NE Lightning

The lightning effect is created by using a sprite renderer module with the lightning material, a spawn burst instantaneous module to spawn a single lightning strike, an initialize particle module with the lifetime and sprite size set to random to add variety to the lightning strike, and a system location module to spawn the

lightning in the air by adjusting the Z value. The colour used on the glow effect is also used with the lightning

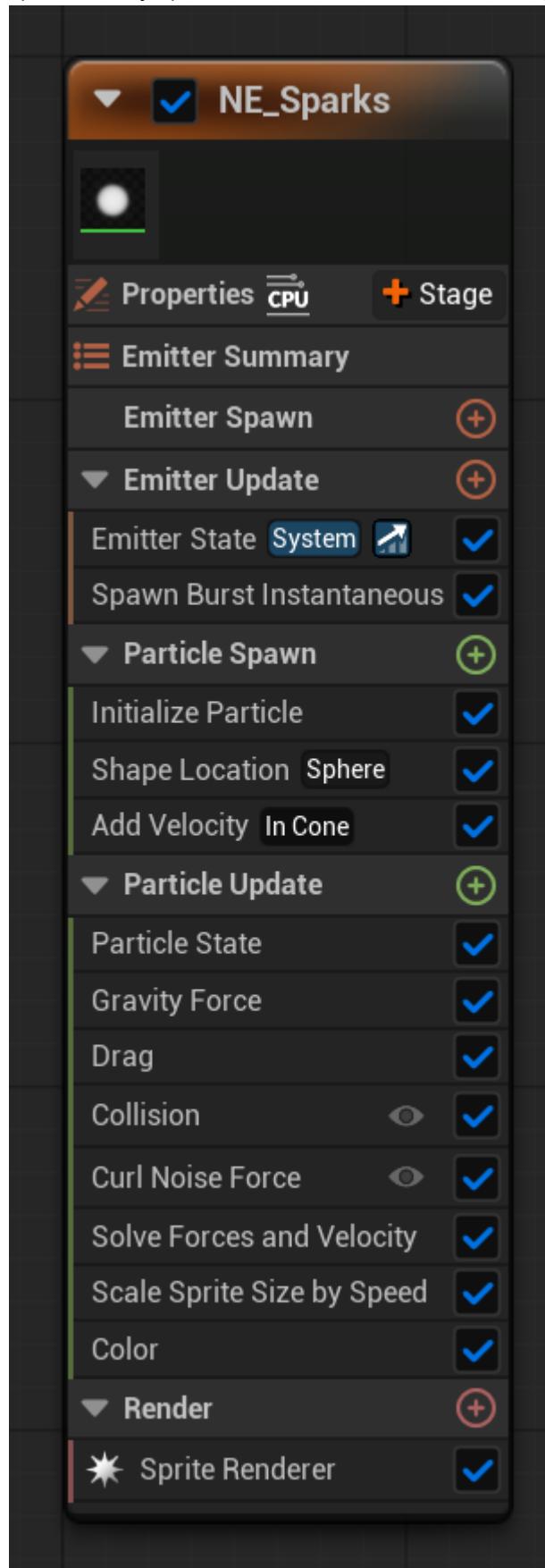


to give it the same blinding, bright effect.

NE Sparks

To create the sparks that appear on the ground when the lightning strikes, multiple modules were required. One of these modules being the spawn burst instantaneous module which has the spawn count set to a random value between fifty and one hundred to add variety to the amount of sparks that spawn. Similarly to the debris effect, add velocity, gravity force, drag, and collision modules are used to fling the particles in the air and then have them slow down before falling to the ground and staying on it till they die out. The scale

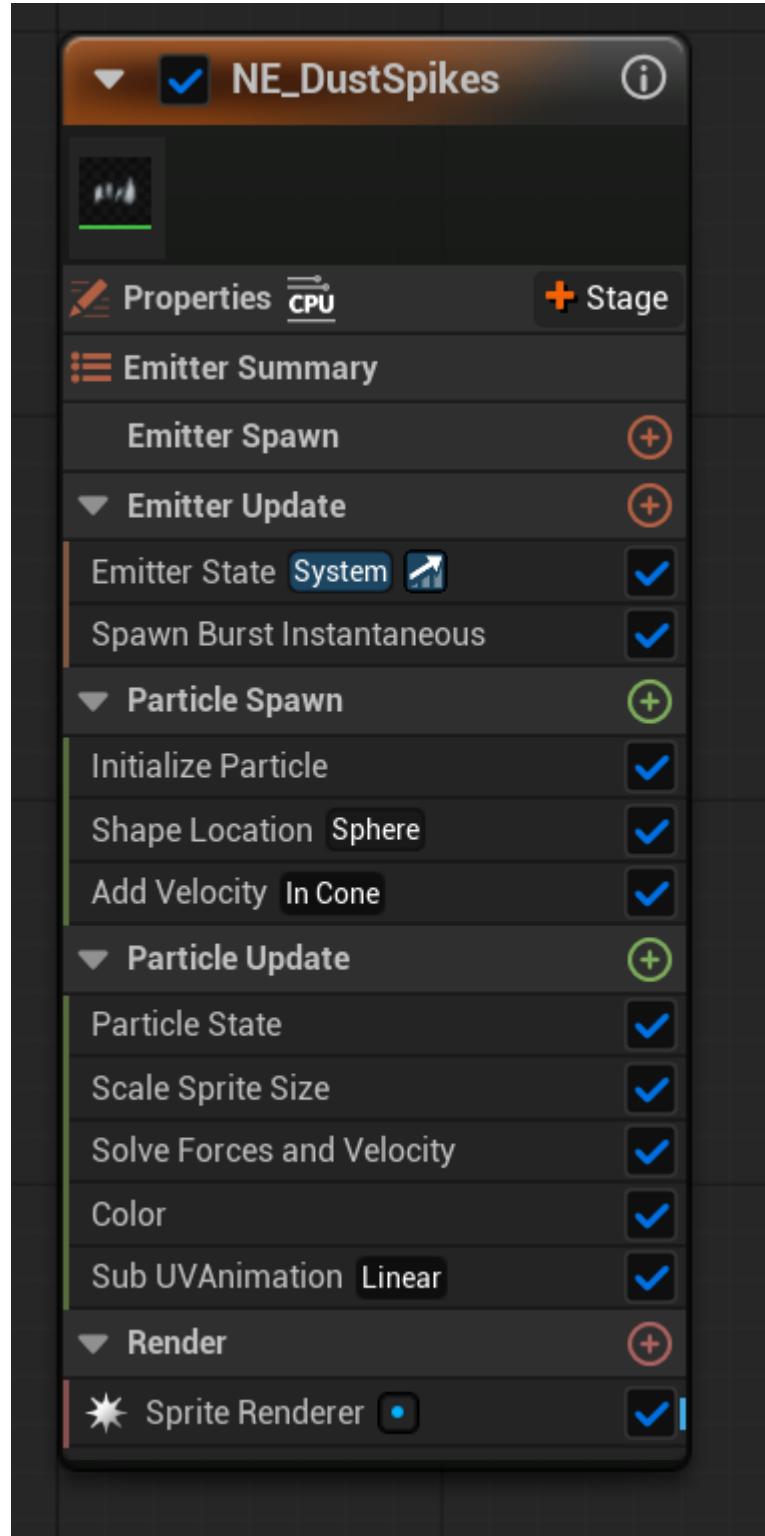
sprite size by speed module is used to show the particles fading out over time.



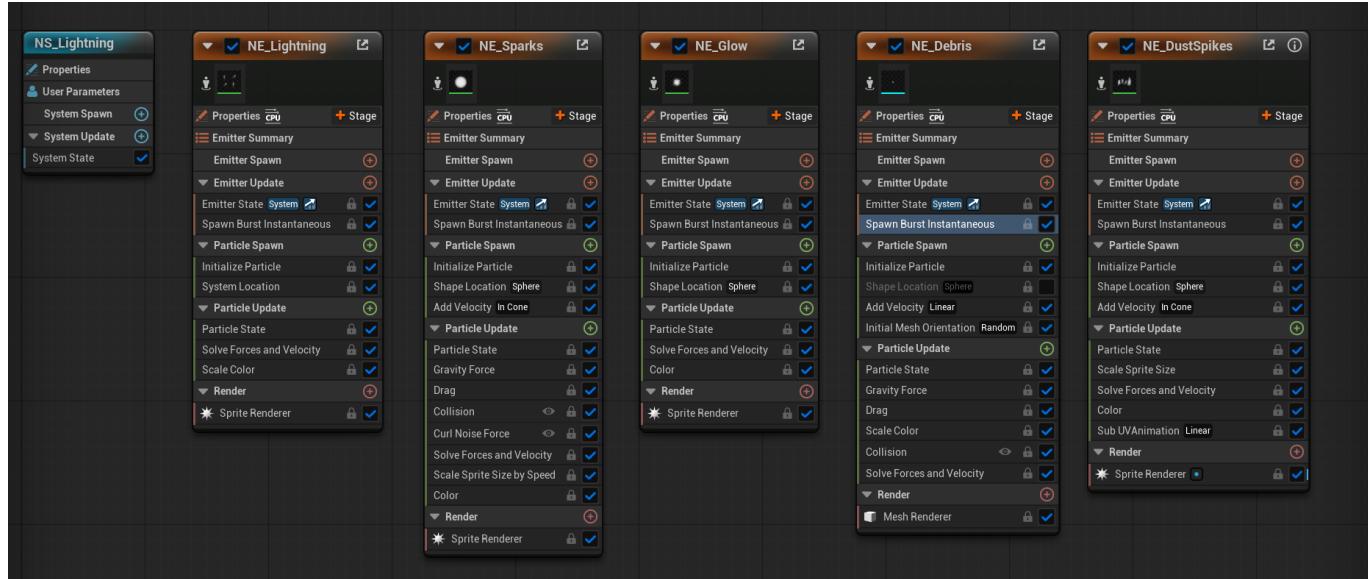
NE Dust Spikes

The dust spike effect uses the dirt material to create an effect that flicks between the drawn dirt smudges to look like dust is coming up from the ground. The main modules used for this effect is the sub UVanimation

module, the spawn burst instantaneous module, and the scale sprite size module. The sub UVanimation module creates the effect of the dirt smudges being displayed in a sequence. The spawn burst instantaneous module is set to spawn eight of the smudges and the scale sprite size module is used to make the dust bigger over time, as if they were coming off the ground and forming a cloud of dust.



NS Lightning



Niagara System Parameters

NS Lightning

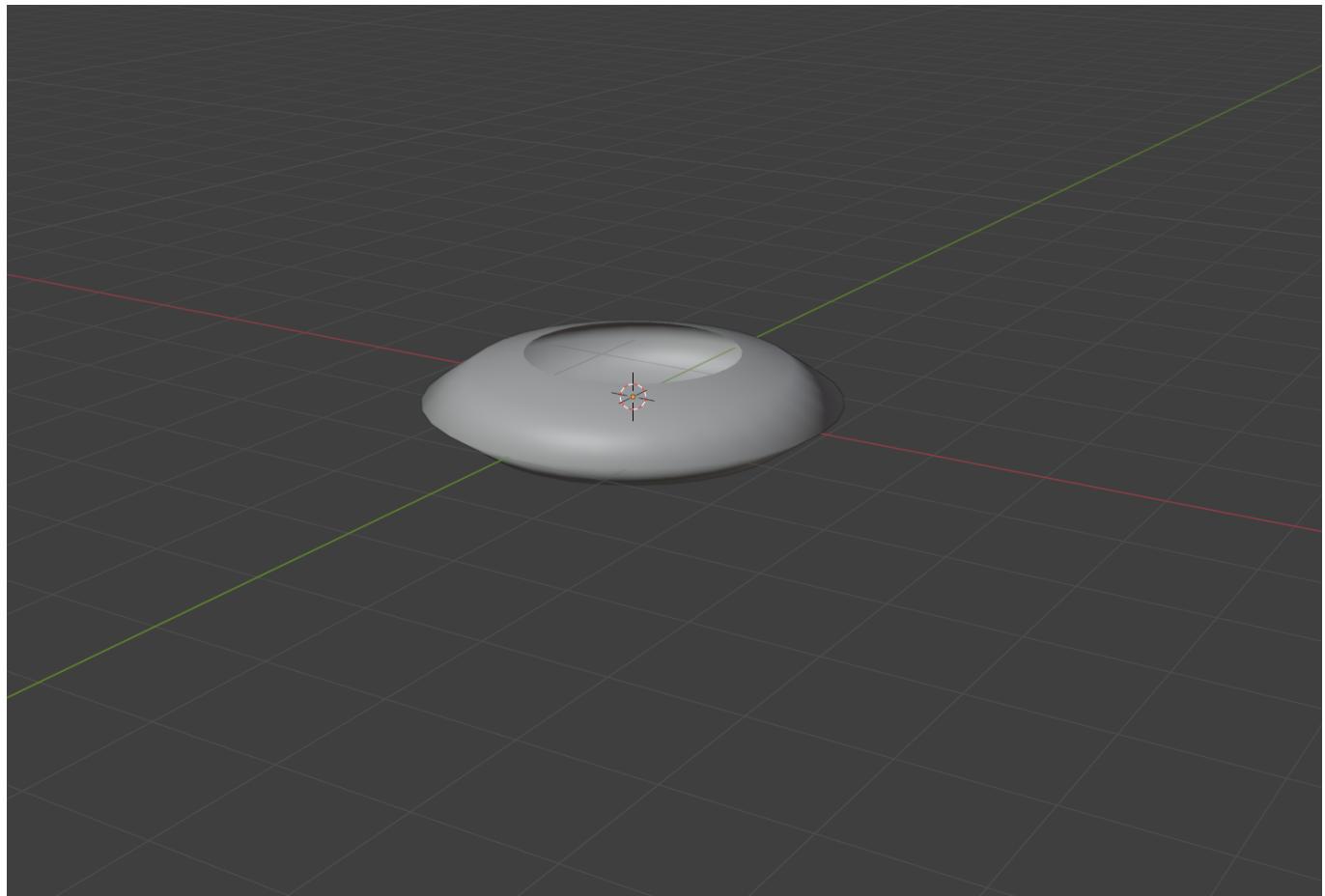
Final Result

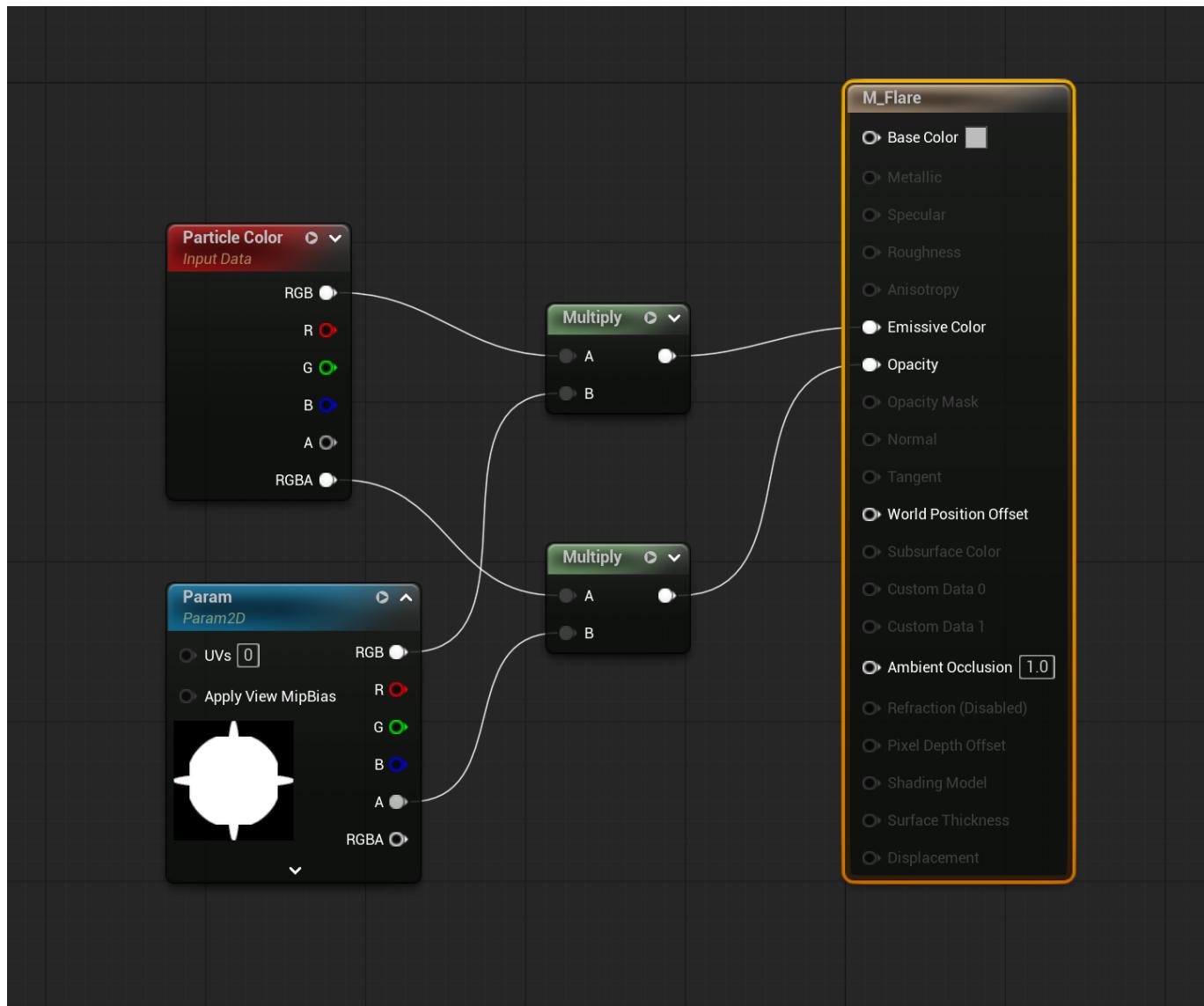
Lightning Showcase

4. Slash Attack VFX

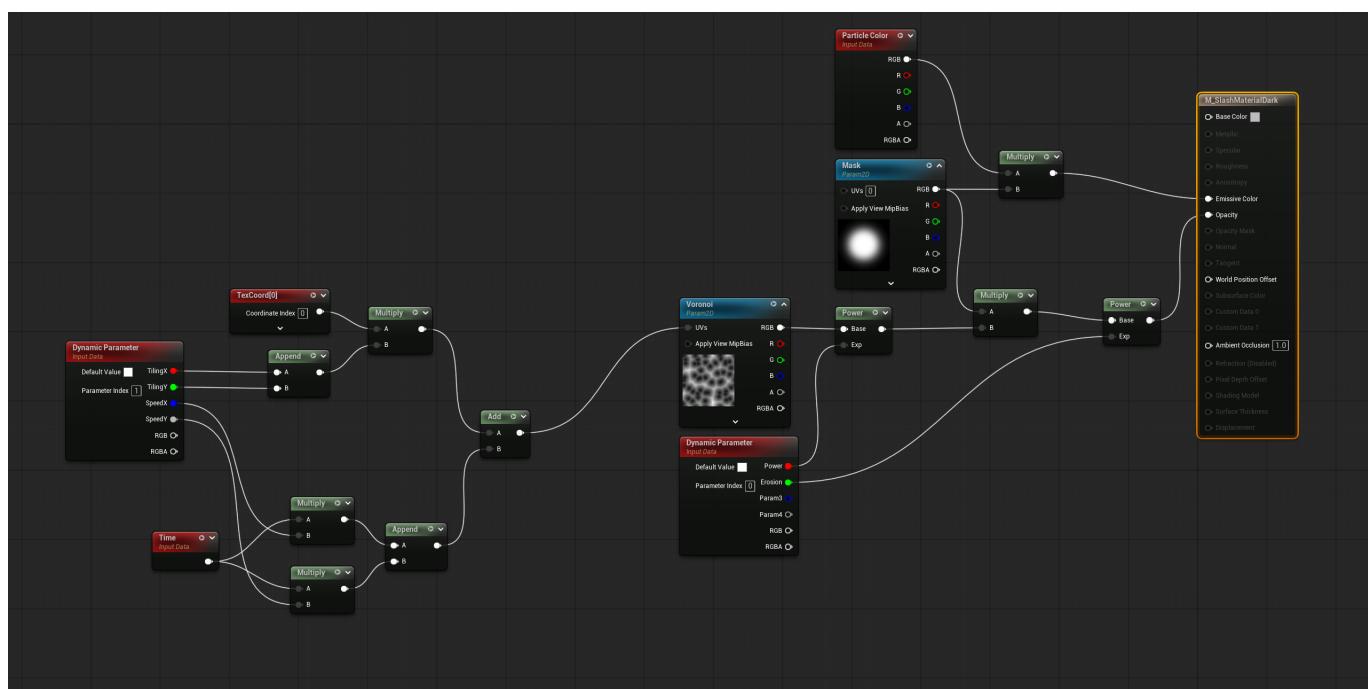
Development Process

Sword Slash 3D Model

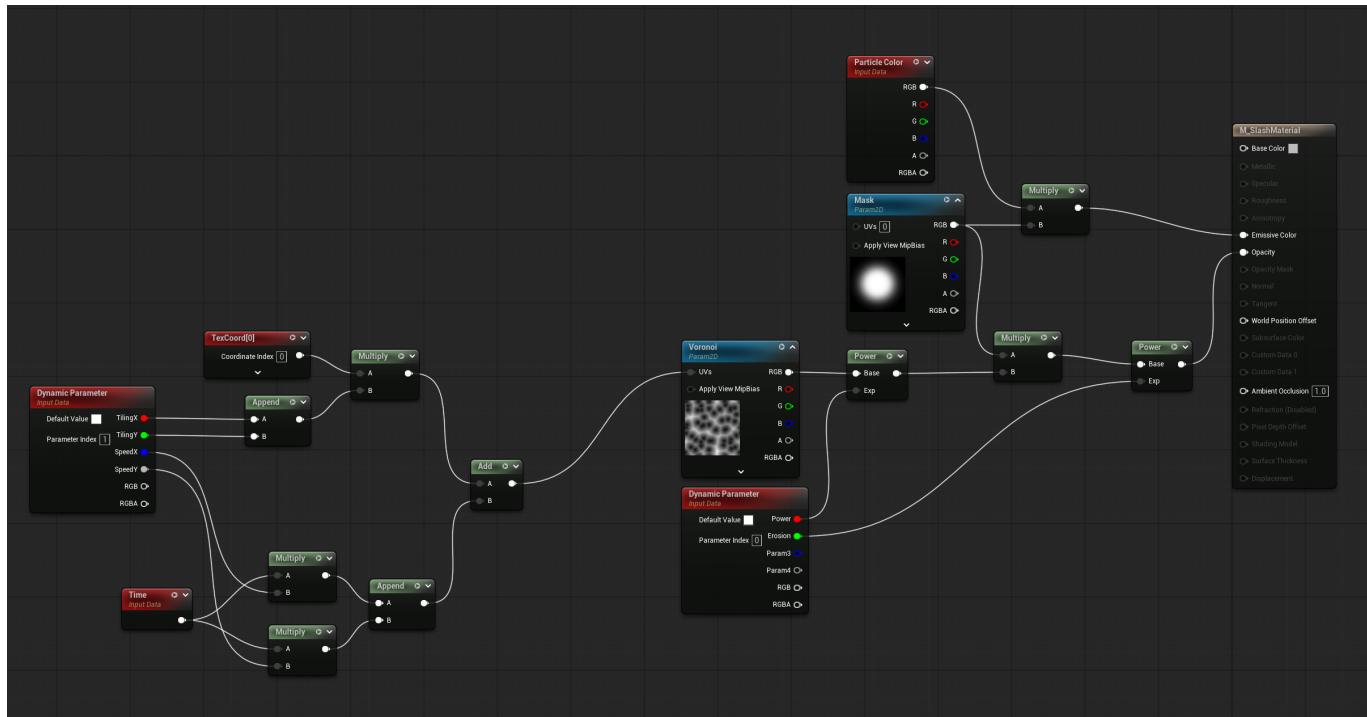
**Flare Material**



Slash Material Dark



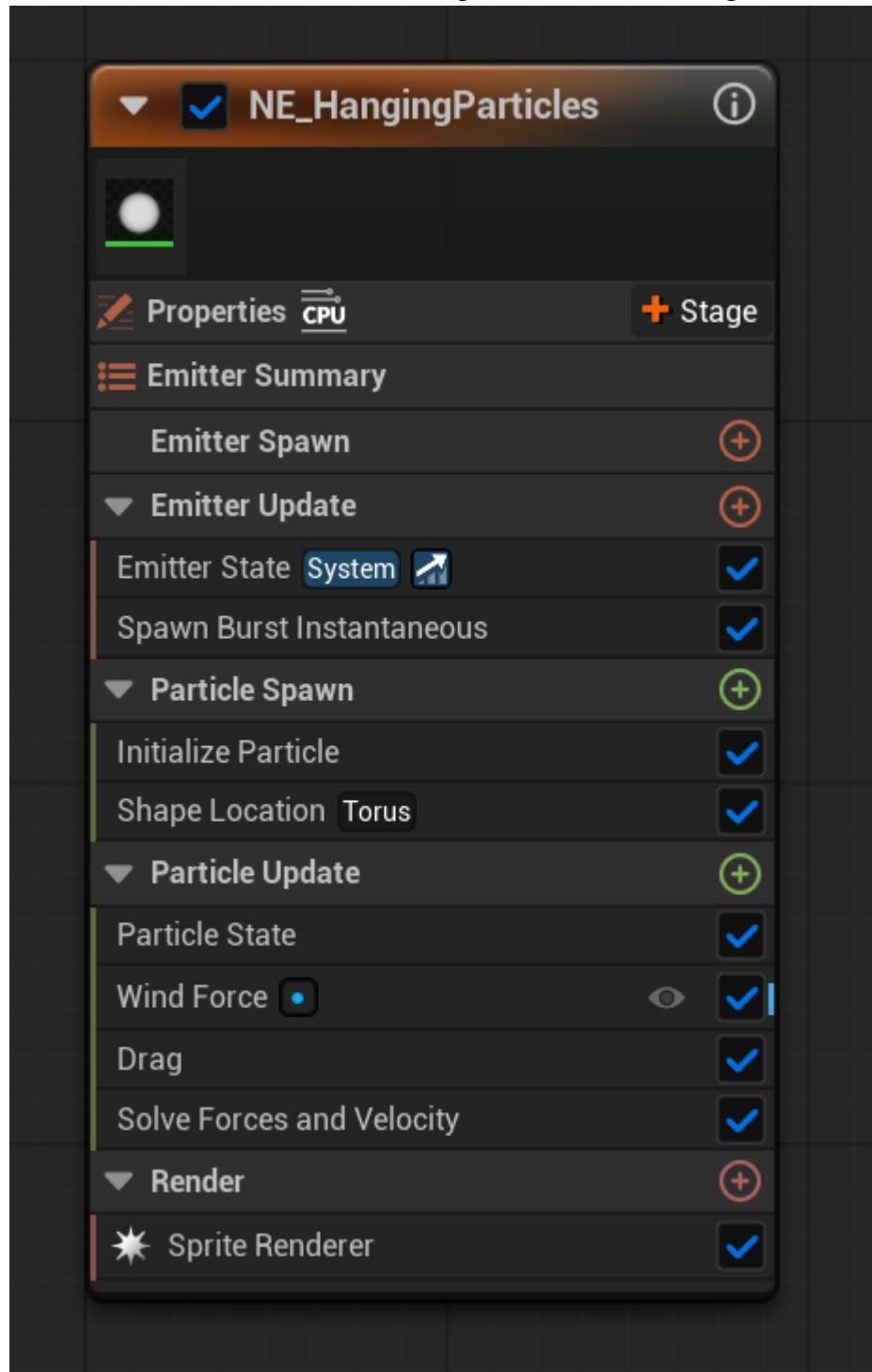
Slash Material



NE Hanging Particles

The hanging particles are created to spawn at the end of the swing to create a lingering magic effect. This is done mainly by the shape location, wind force, drag, and initialize particle module. As the swing is in the shape of a torus, to match that, a shape location module is used with the shape primitive set to torus to spawn the particles in a similar shape to the swing. The wind force and drag allow the particles to float and move in the air. The lifetime and sprite size are set to random within the initialize particles module. This

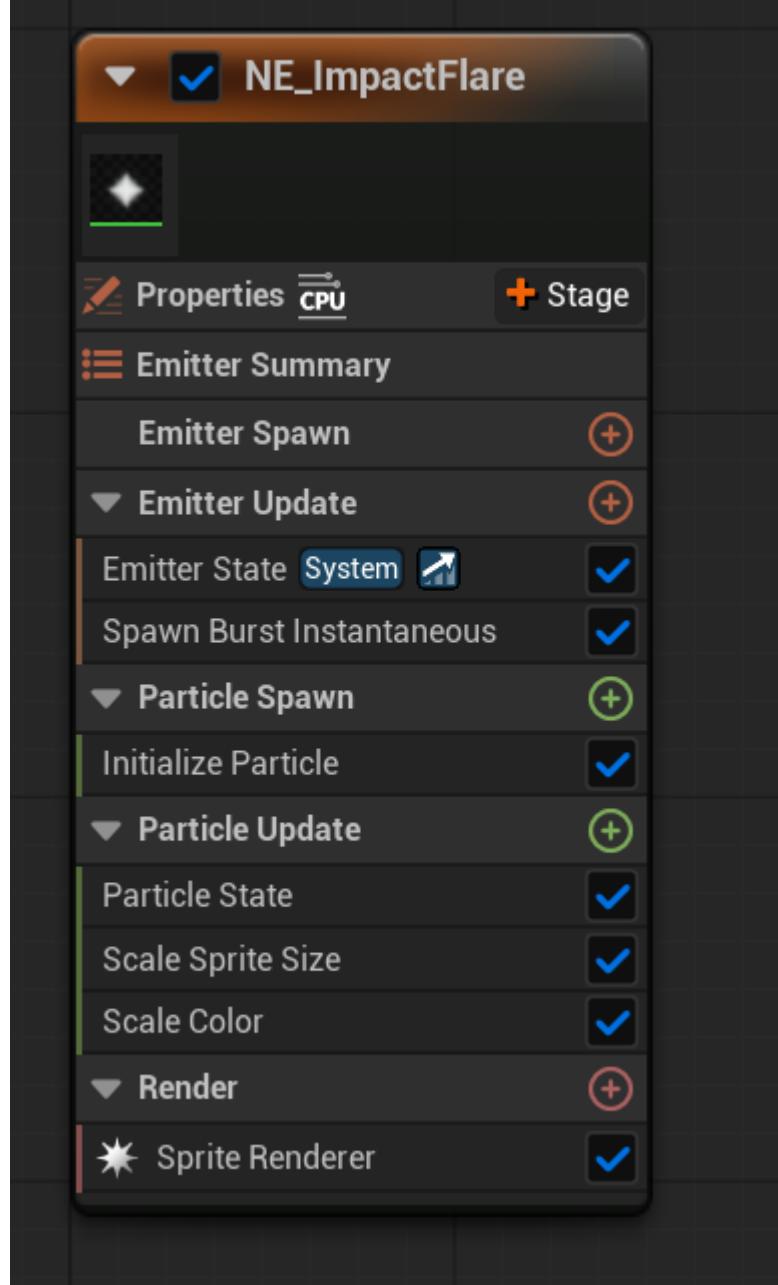
module also sets the colour to be a bright blue, similar to the glow effect from the lightning.



NE Impact Flare

The impact flare uses the flare material to make a glint-like effect at the end of the swing to show that something was hit. The niagara system spawns one of the sprites and then, in the initialize particle node, sets the lifetime to 0.15 so that it doesn't last long, sets the colour to a glowing orange colour, and then sets the

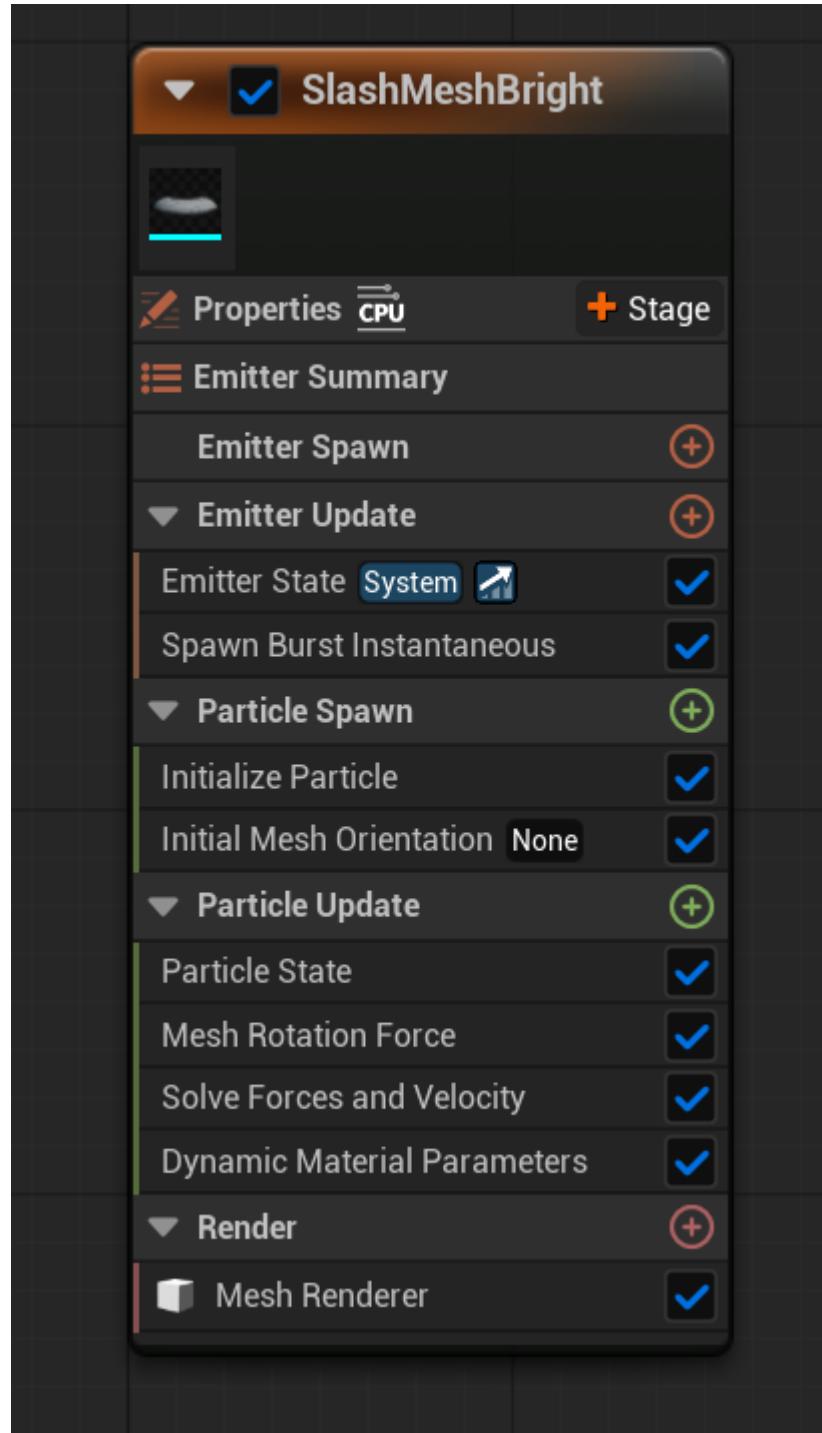
effect to have an offset so that it spawns at the top left of the effect, where something would be hit.



NE Slash Mesh Bright 1

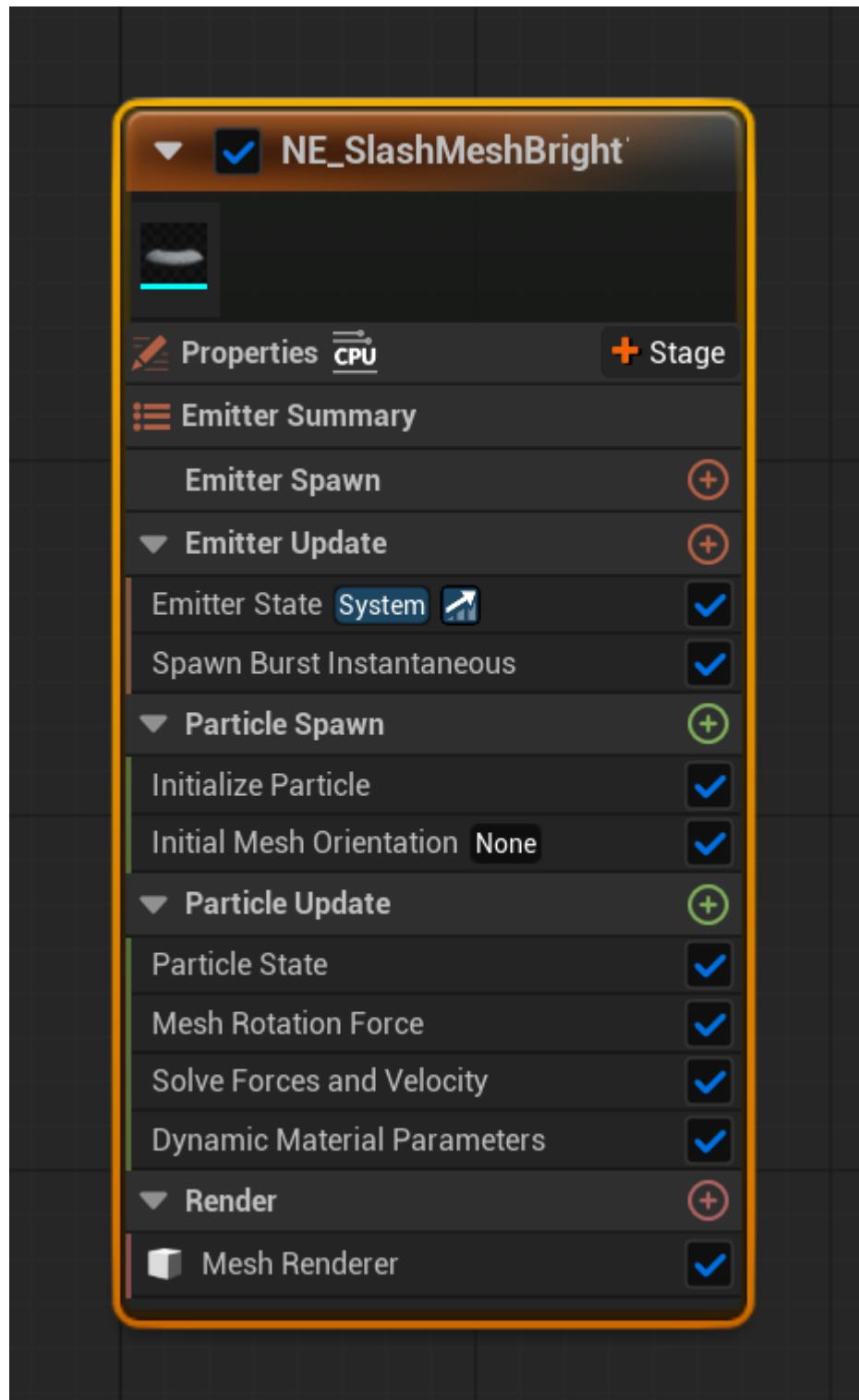
The first bright slash mesh uses a mesh renderer with a ring mesh that I created in Blender. This mesh is used to create the ring effect that simulates a sword swing. Mesh rotation force and dynamic material parameter modules are used to create the effect that the colour is moving around the torus mesh. Initial mesh orientation is also used to rotate the effect around the torus in a specific direction. In this effect's initialize particle module, the lifetime is directly set to 0.35 so that it doesn't last long, the mesh scale is set so there is no variation in the size and also the position offset is directly set so that it spawns at a specific place within

the effect. The colour of this slash is set to a bright blue within the previous module as well.



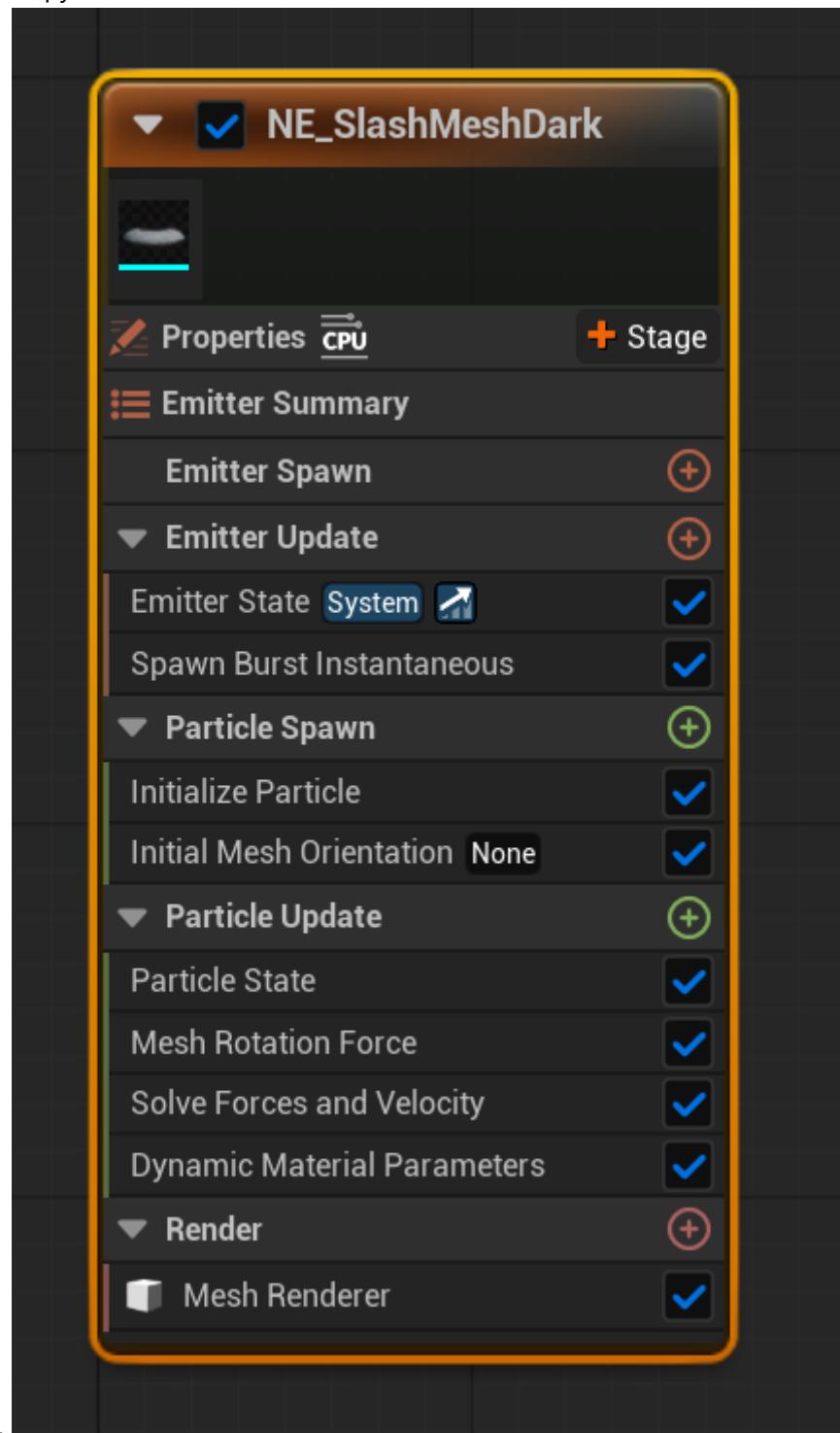
NE Slash Mesh Bright 2

The second slash mesh bright effect is a copy of the first one, but with a different offset to reduce overlapping and with a bright purple colour instead of bright blue.



NE Slash Dark

The dark material is also a copy of the first one with a different offset, but this time the colour has been

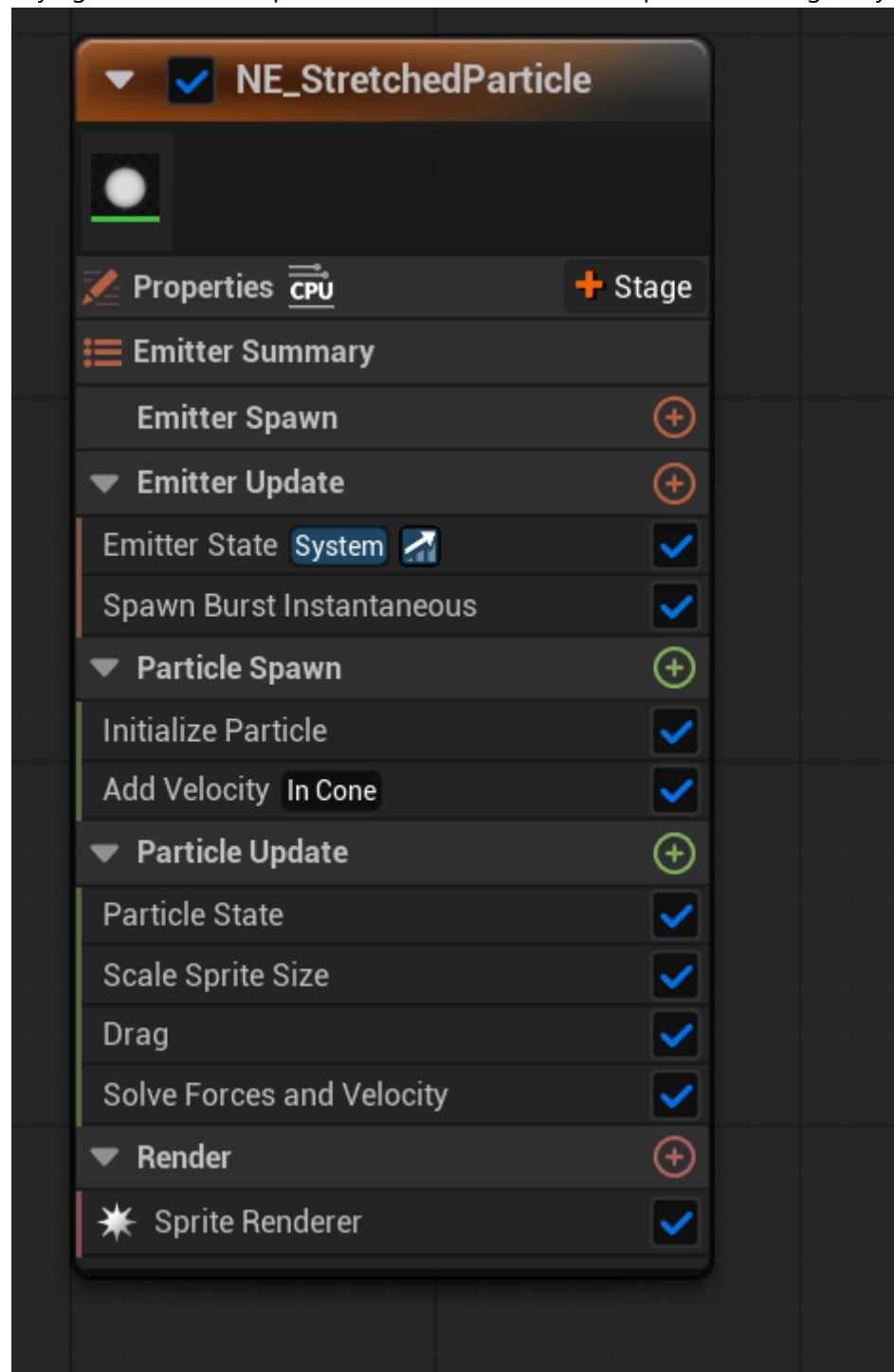


changed to a grey colour.

NE Stretched Particles

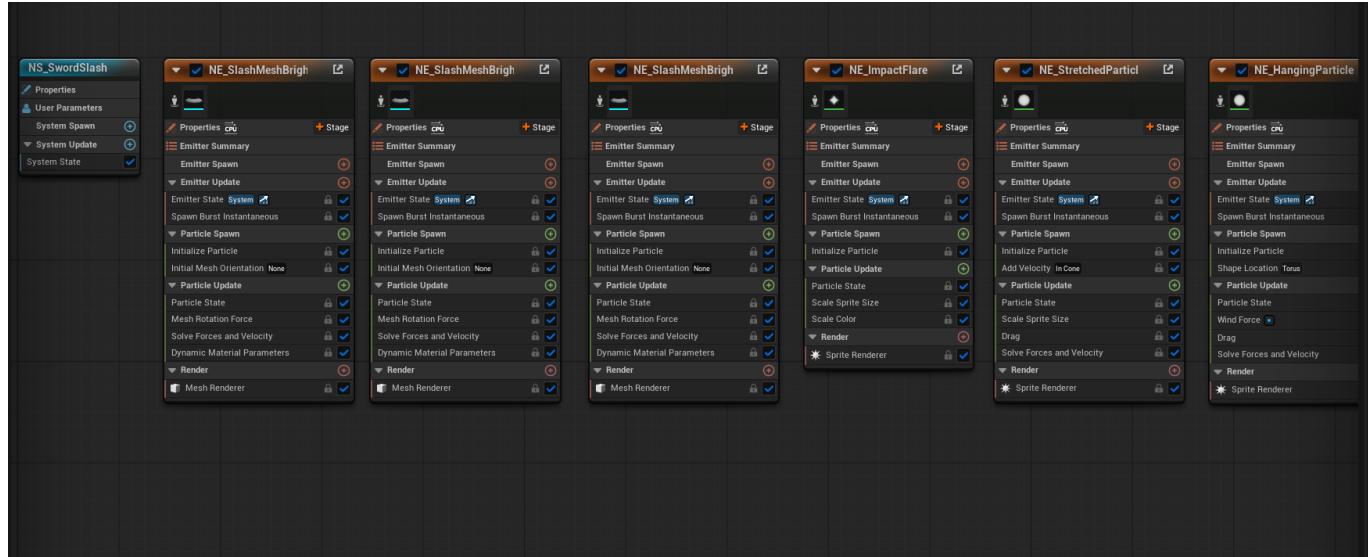
The stretched particles are used at the collision location to look like sparks flying off the sword as it hits something. This effect is made by using the spawn burst instantaneous, add velocity, drag, and scale sprite size modules. The spawn burst instantaneous module causes the particles to only spawn once, and the spawn count is set to twenty. The add velocity module uses the cone velocity mode to spawn the sparks at a point and then have them flying off forwards in a cone shape. The drag module is just used to slow down the

particles as they are flying, and the scale sprite size module simulates the particles fading away as their size is reduced over time.



reduced over time.

NS Sword Slash

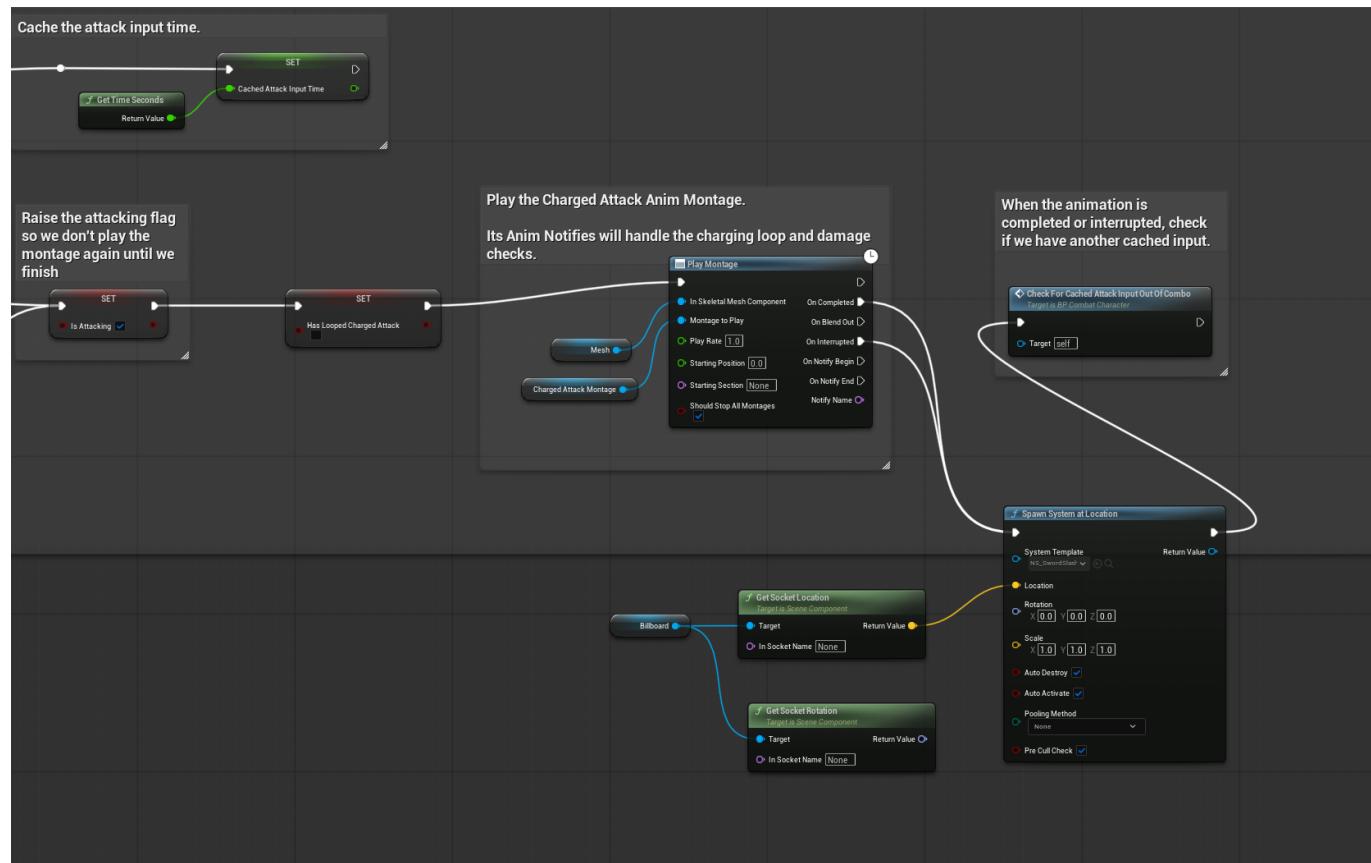


Niagara System Parameters

NS Sword Slash

Sword Slash Implementation

What does it do? Using the script provided by the combat variant in Unreal Engine 5.6, this script is added in to spawn the sword system at a billboard that is in front of the player after the attack animation is played.



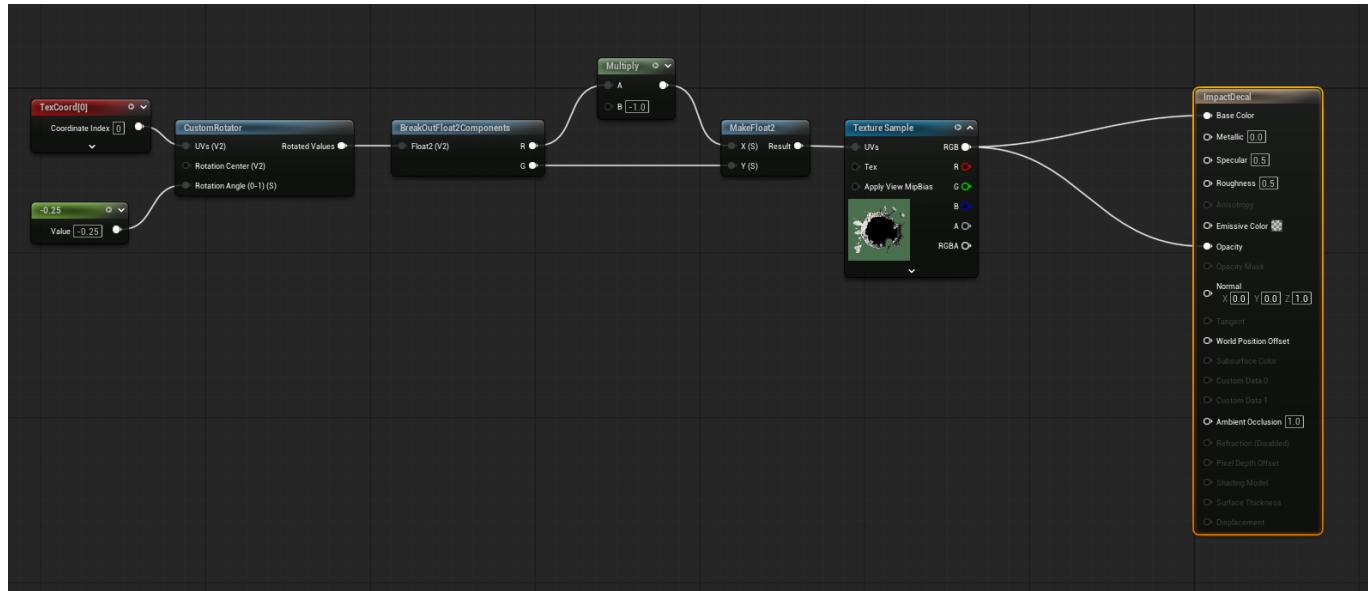
Final Result

Sword Slash Showcase

5. Decal Impact (triggered by Animation Notify)

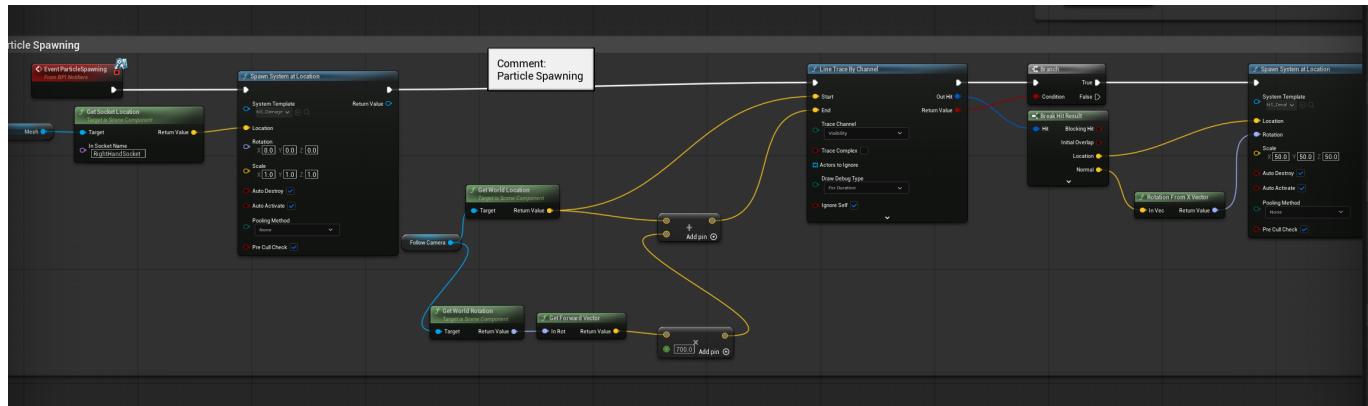
Development Process

Decal Material



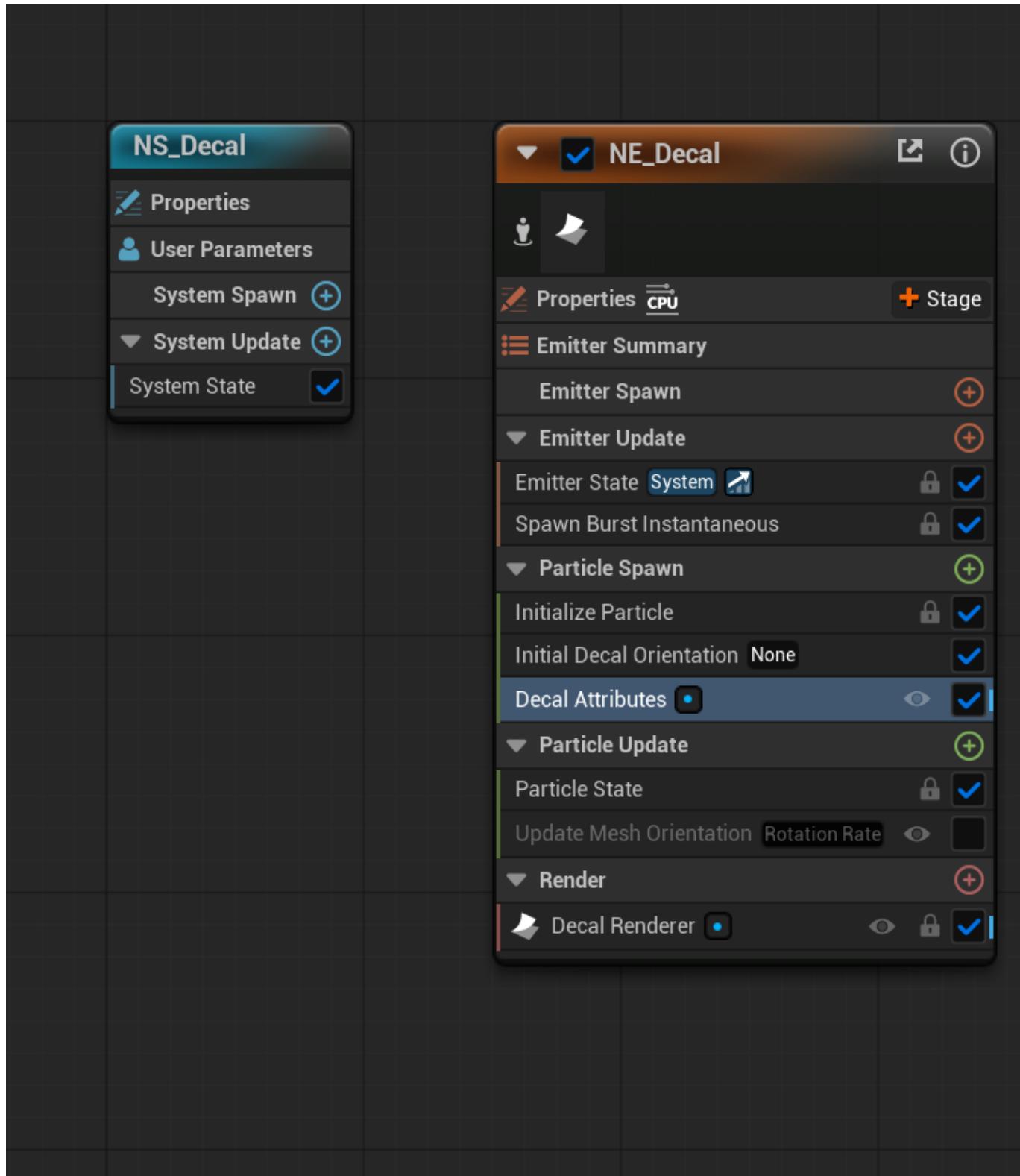
Decal Spawning

What does it do? When the decal spawning event is called after an attack animation plays, a damage particle system is spawned at the player's right hand. Then, a line trace is created to check if the player has hit a surface. If a surface is hit by the line trace, a decal is spawned where the line trace hit.



NS Decal

The decal niagara system just uses a spawn burst instantaneous module to spawn one decal, an initialize particle module to set the decal's lifetime and a decal renderer to render the decal.



Niagara System Parameters

[NS Decal Parameters](#)

Final Result

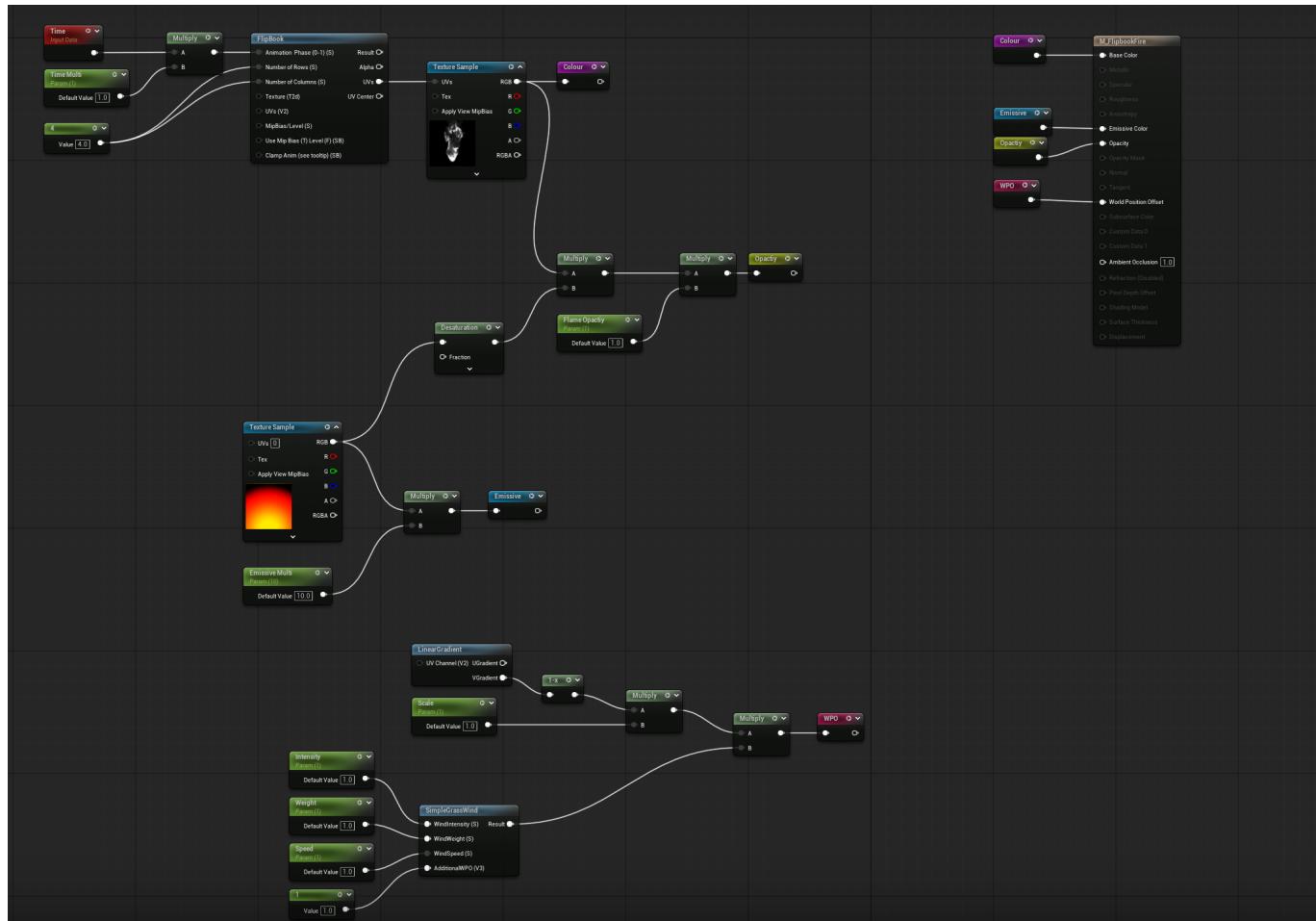
[Decal Showcase](#)

6. SubUV Flipbook Implementation (Fire)

The fire flipbook uses a sheet of images of variations of the same fire to simulate an effect or video of a fire. This is done by flicking through the images fast, like going through a flipbook.

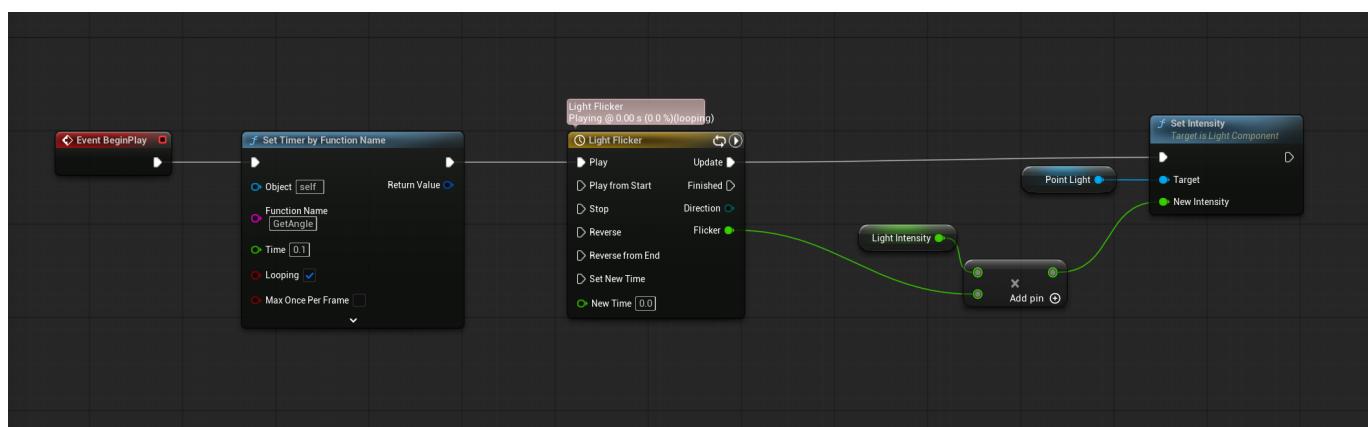
Development Process

Fire Material

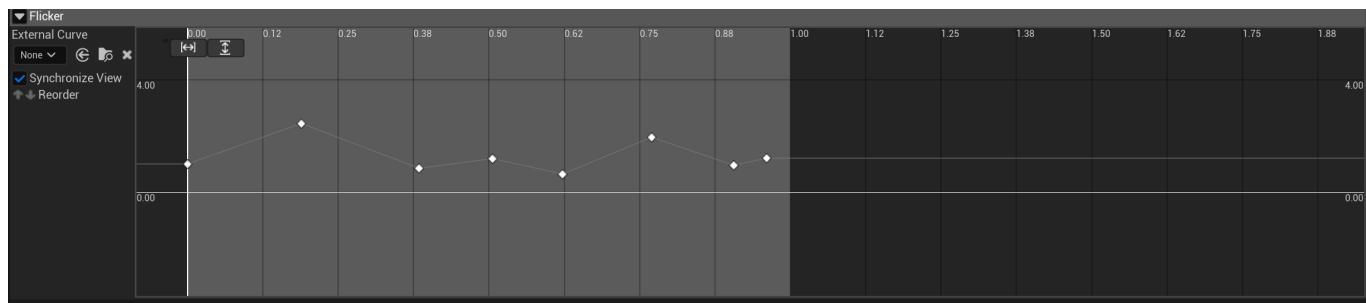


Fire Blueprint

What does it do? This blueprint rotates the fire so that it is always facing the player and also causes the light intensity to constantly change based on a timeline to simulate the fire flickering.

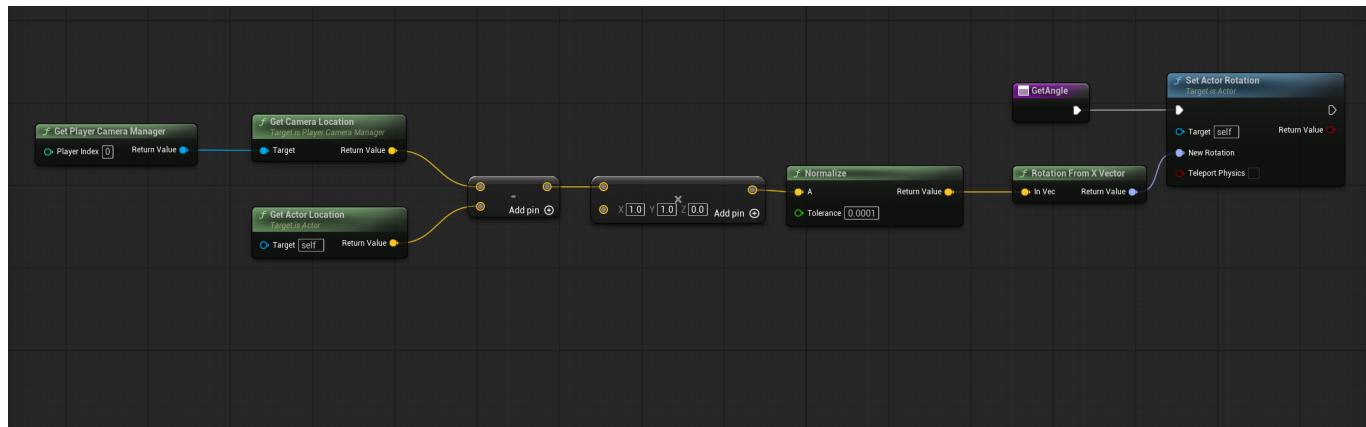


Flicker Timeline

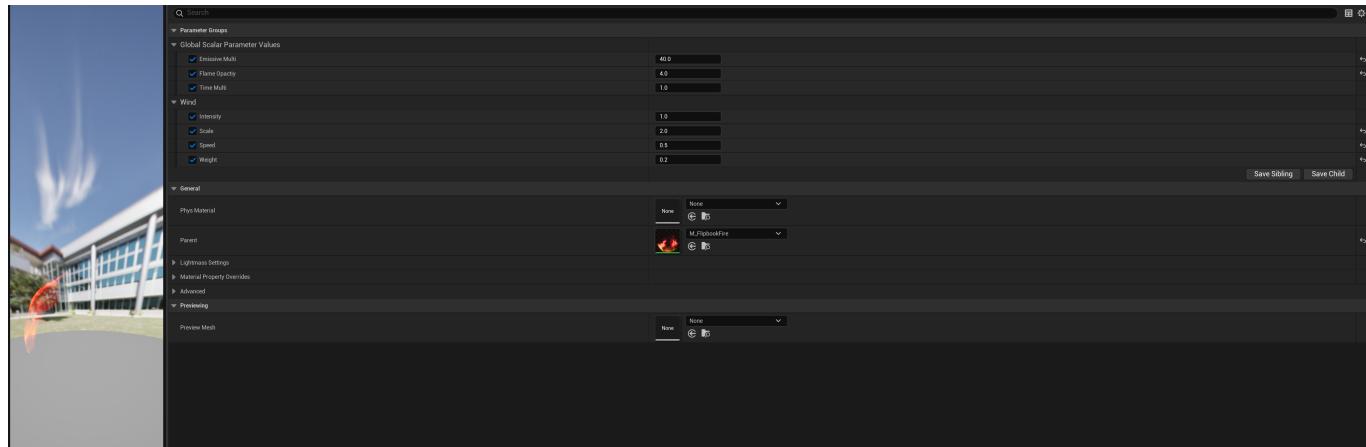


Get Player Angle Function

What does it do? This blueprint sets the rotation of the fire so that it is facing towards the player camera.



Material Instance Parameters



Final Result

Flipbook Fire Showcase

Bibliography

- Unreal Engine 5 Tutorial - Stylized Lightning Strike in Niagara (2024) Directed by UnleashFX. At: <https://www.youtube.com/watch?v=vV7OqTAiHzg> (Accessed 28/10/2025).
- Unreal Engine 5 - Sword Slash VFX - Niagara Tutorial (2024) Directed by Gabriel Aguiar Prod. At: <https://www.youtube.com/watch?v=djInnPvFR0Q> (Accessed 28/10/2025).

- UE5 | Spawning Decal on Your Level | 5-Minute Material Tutorial | Unreal Engine 5 (2023) Directed by Coreb Games. At: <https://www.youtube.com/watch?v=6MXGzvot8tA> (Accessed 01/11/2025).
- UE4 Tutorial: Fire Effect (2018) Directed by underscore. At: <https://www.youtube.com/watch?v=B8Gya38KkJY> (Accessed 02/11/2025).