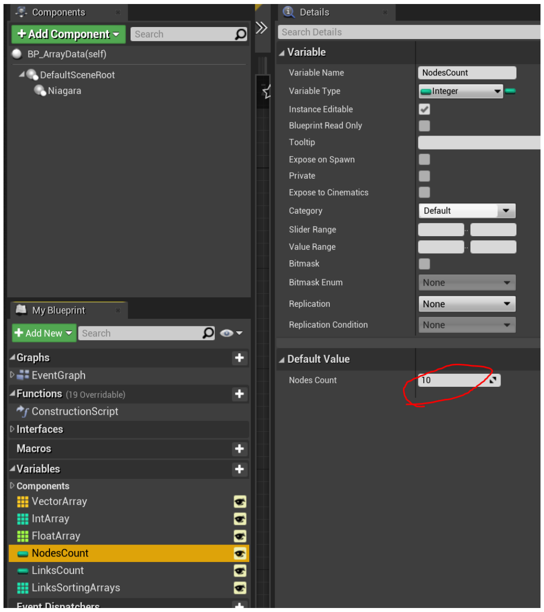
My Goal

Source Particles (Nodes)

Would like to spawn secondary particles (Links) per source particles (Nodes) Index or ID.

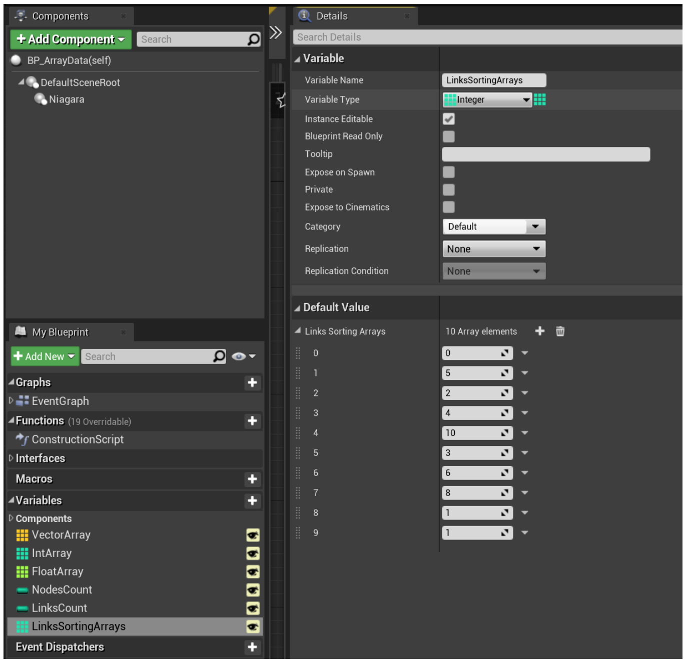


Variable

Variable Name	NodesCount
Variable Type	Integer
Instance Editable	☑
Blueprint Read Only	□
Tooltip	
Expose on Spawn	□
Private	□
Expose to Cinematics	□
Category	Default
Slider Range	
Value Range	
Bitmask	□
Bitmask Enum	None
Replication	None
Replication Condition	None

Default Value

Nodes Count	10



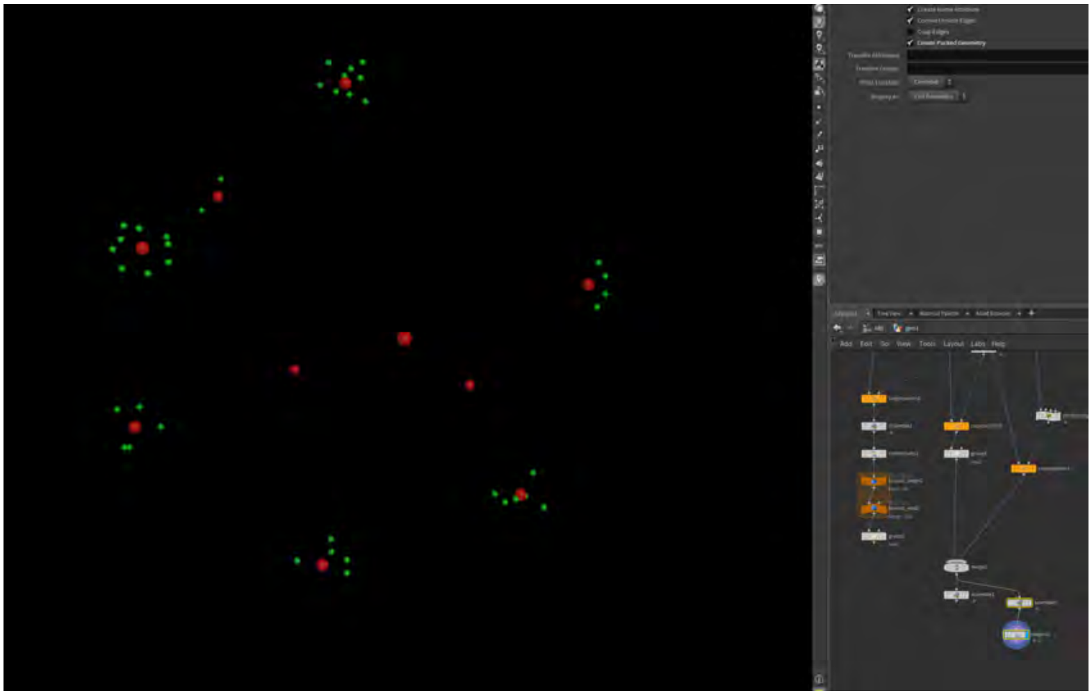
Variable

Variable Name	LinksSortingArrays
Variable Type	Integer
Instance Editable	☑
Blueprint Read Only	□
Tooltip	
Expose on Spawn	□
Private	□
Expose to Cinematics	□
Category	Default
Replication	None
Replication Condition	None

Default Value

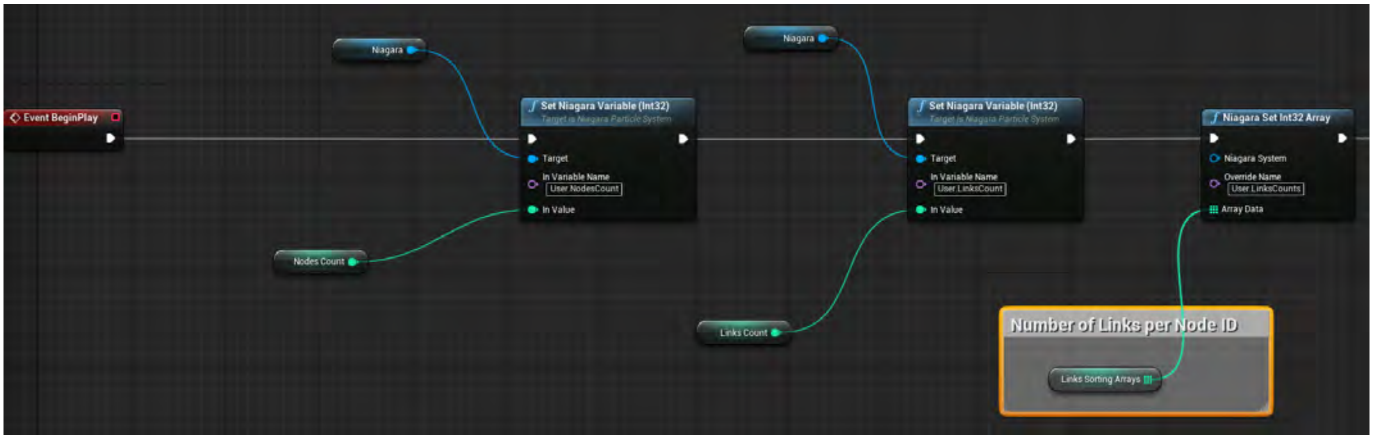
Links Sorting Arrays	10 Array elements
0	0
1	5
2	2
3	4
4	10
5	3
6	6
7	8
8	1
9	1

Example Visual of what I am trying to accomplish, probably with beam arc to link the red and green particles

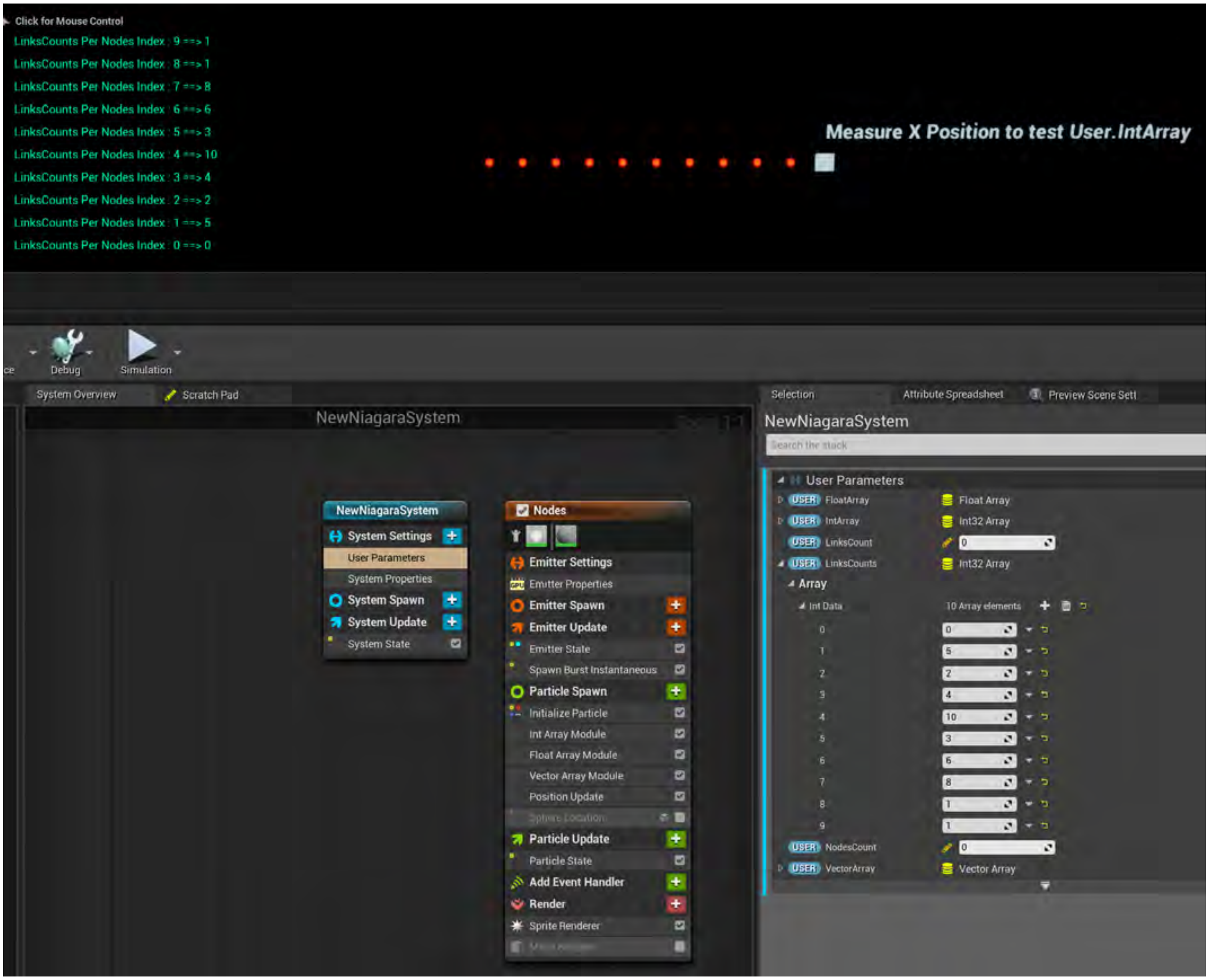


Here is an example I made in Houdini for illustration purposes. The green spheres count is sorted using array table.

Blueprint Setup for custom User Parameters



Tested the custom user.IntArray rm aml dgp k gr umpiq.



```
Click for Mouse Control
LinksCounts Per Nodes Index : 9 ==> 1
LinksCounts Per Nodes Index : 8 ==> 1
LinksCounts Per Nodes Index : 7 ==> 8
LinksCounts Per Nodes Index : 6 ==> 6
LinksCounts Per Nodes Index : 5 ==> 3
LinksCounts Per Nodes Index : 4 ==> 10
LinksCounts Per Nodes Index : 3 ==> 4
LinksCounts Per Nodes Index : 2 ==> 2
LinksCounts Per Nodes Index : 1 ==> 5
LinksCounts Per Nodes Index : 0 ==> 0
```

Measure X Position to test User.IntArray

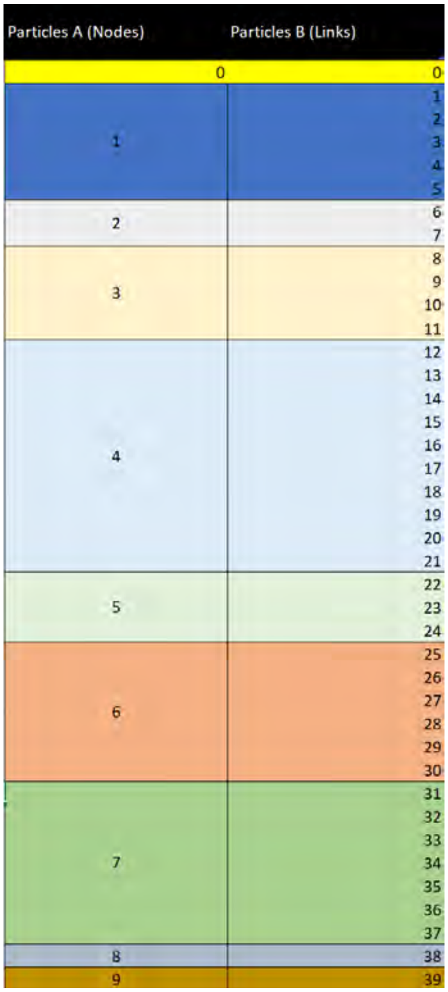
NewNiagaraSystem

User Parameters

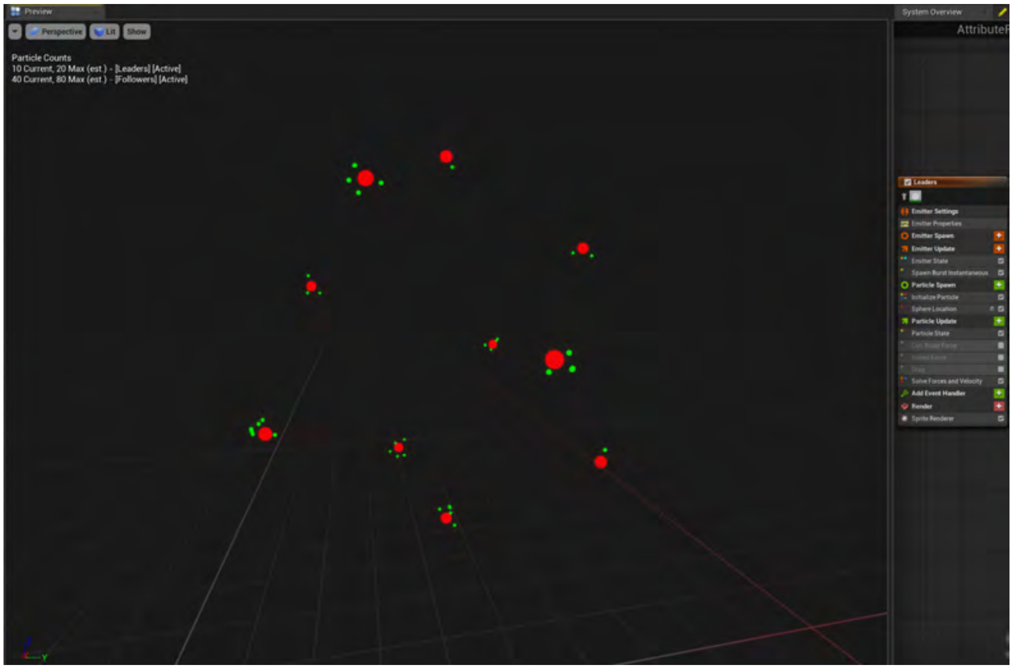
USER FloatArray	Float Array
USER IntArray	Int32 Array
USER LinksCount	0
USER LinksCounts	Int32 Array
Array	
Int Data	10 Array elements
0	0
1	5
2	2
3	4
4	10
5	3
6	6
7	8
8	1
9	1
USER NodesCount	0
USER VectorArray	Vector Array

My goal is as follows:

1. To emit Particles B (Links based on Particles A (Nodes) inheriting the location as well.
2. If that cannot be done, then I am thinking of spawning the total Particles B count and then update their positions to Particles A by ID either by groups or clustering/bucket.



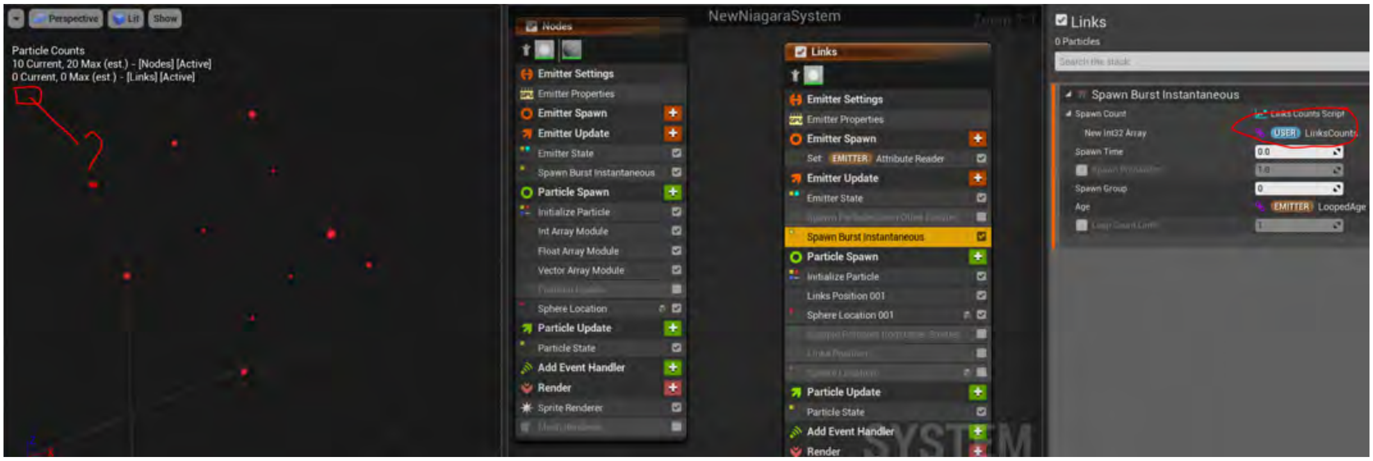
Particles A (Nodes)	Particles B (Links)
0	0
1	1 2 3 4 5
2	6 7
3	8 9 10 11
4	12 13 14 15 16 17 18 19 20 21
5	22 23 24
6	25 26 27 28 29 30
7	31 32 33 34 35 36 37
8	38
9	39



Particle Counts
10 Current, 20 Max (est.) - [Leaders] [Active]
40 Current, 80 Max (est.) - [Followers] [Active]

I feel that I am very close to achieve my target as illustrated above. My only problem using this setup that the secondary particles are distributed randomly.

When I try setting the Spawn Burst Instantaneous Count, I only have the Dynamic Scratch module option.



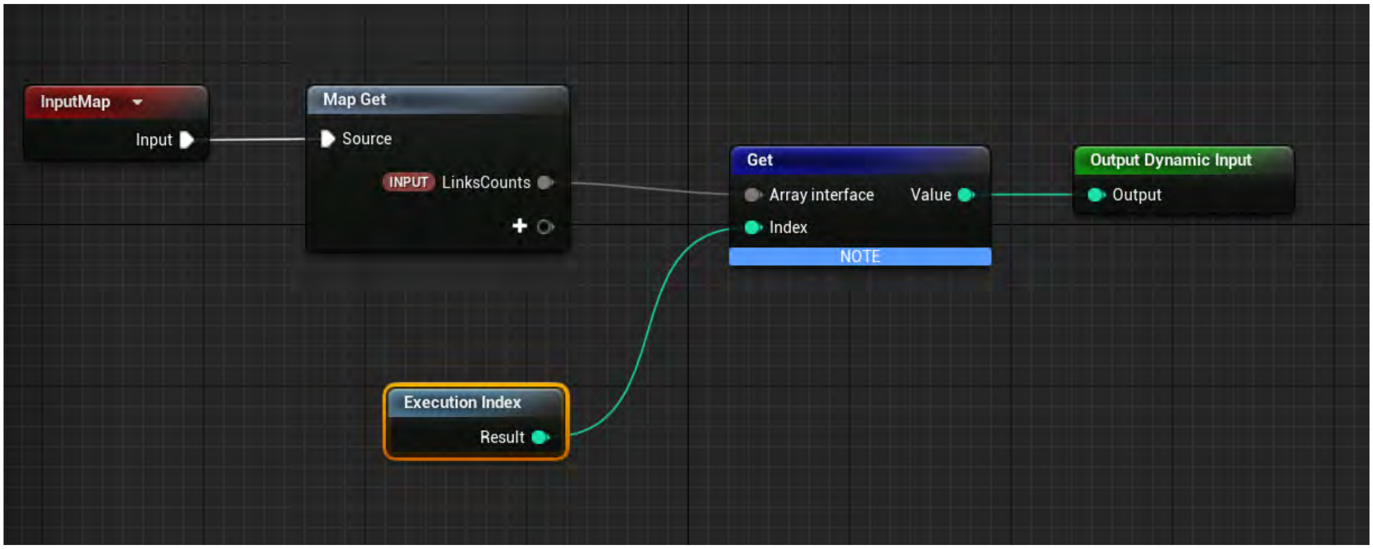
Particle Counts
10 Current, 20 Max (est.) - [Nodes] [Active]
0 Current, 0 Max (est.) - [Links] [Active]

Links
0 Particles

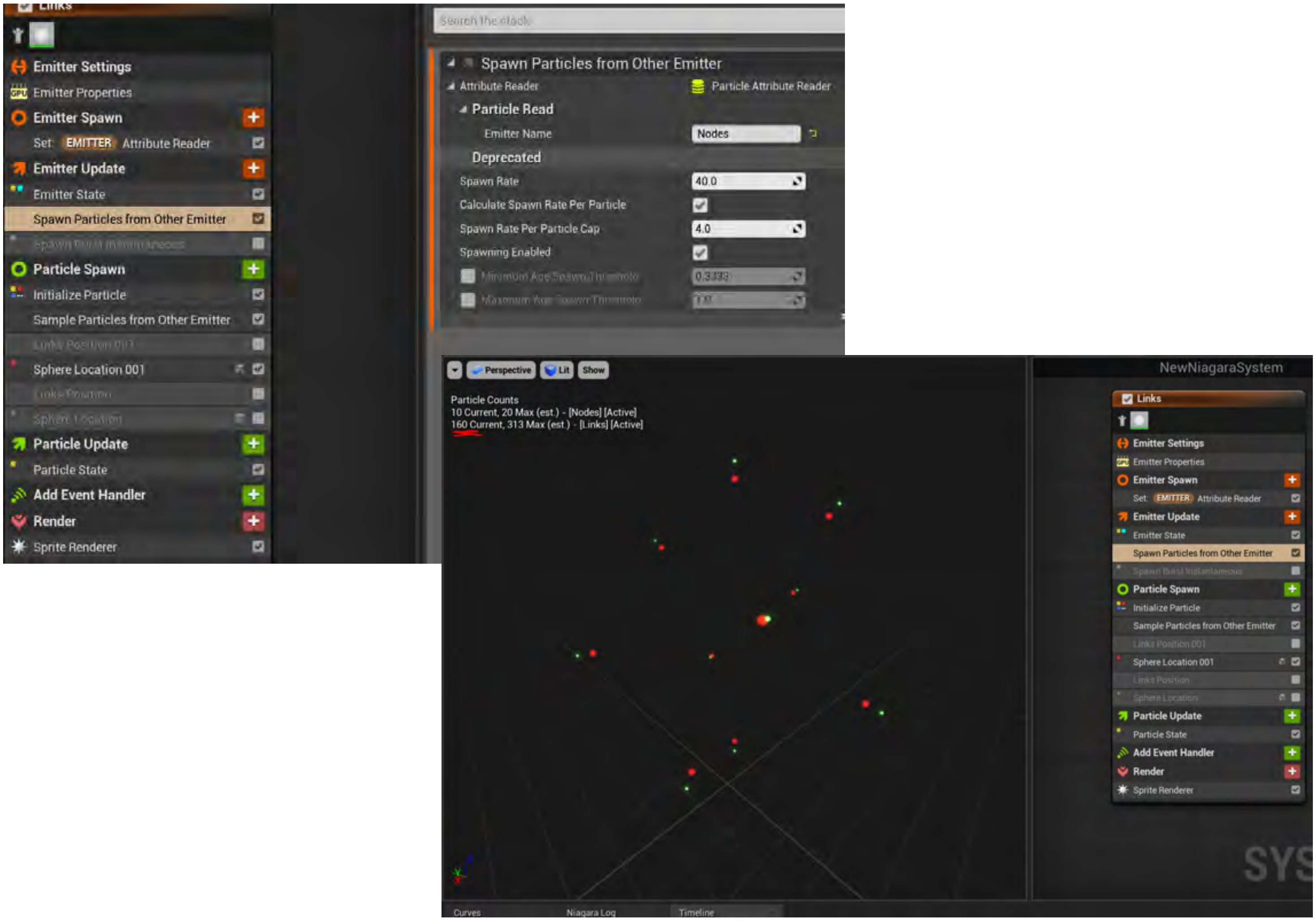
Spawn Burst Instantaneous
Spawn Count: Links Counts Script
New Int32 Array: USER LinksCounts
Spawn Time: 0.0
Spawn Group: 0
Age: EMITTER LoopedAge

Continued next page,

Dynamic Scratch module, the result is Zero spawned secondary particles



I have also tried to use the Spawn Particles from other emitter module! However, it runs on Spawn rate per particle and couldn't figure out how to spawn per particle.



Spawn Particles from Other Emitter

Attribute Reader: Particle Attribute Reader

Particle Read

Emitter Name: Nodes

Deprecated

Spawn Rate: 40.0

Calculate Spawn Rate Per Particle: ☑

Spawn Rate Per Particle Cap: 4.0

Spawning Enabled: ☑

□ Minimum Age Spawn Threshold: 0.3333

□ Maximum Age Spawn Threshold: 1.0

```
Particle Counts
10 Current, 20 Max (est.) - [Nodes] [Active]
160 Current, 313 Max (est.) - [Links] [Active]
```