Hexagon Map System

Source for hexagon math: https://www.redblobgames.com/grids/hexagons/

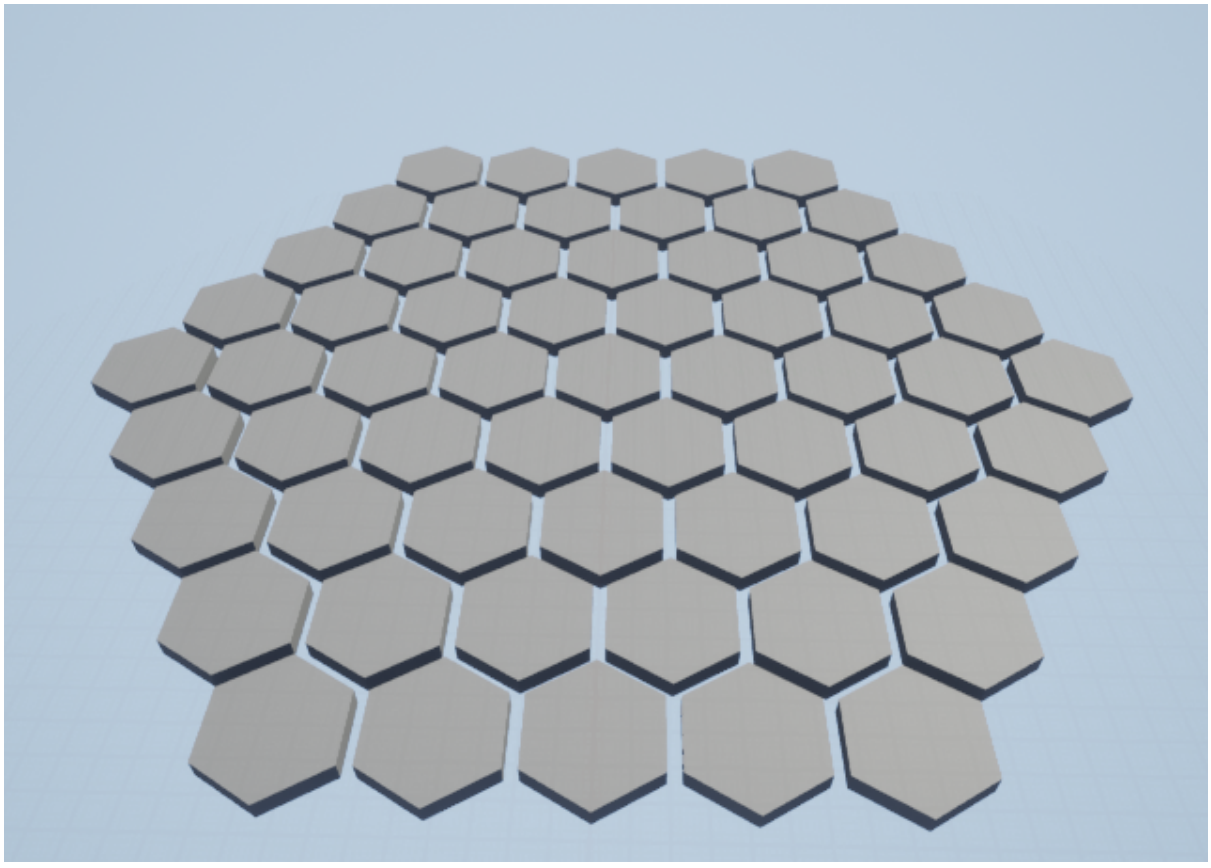
This blueprint seeks to create a hexagon grid game map like in games such as Civilization or Dungeons and Dragons

Inside of a level, the GameManager actor is placed.

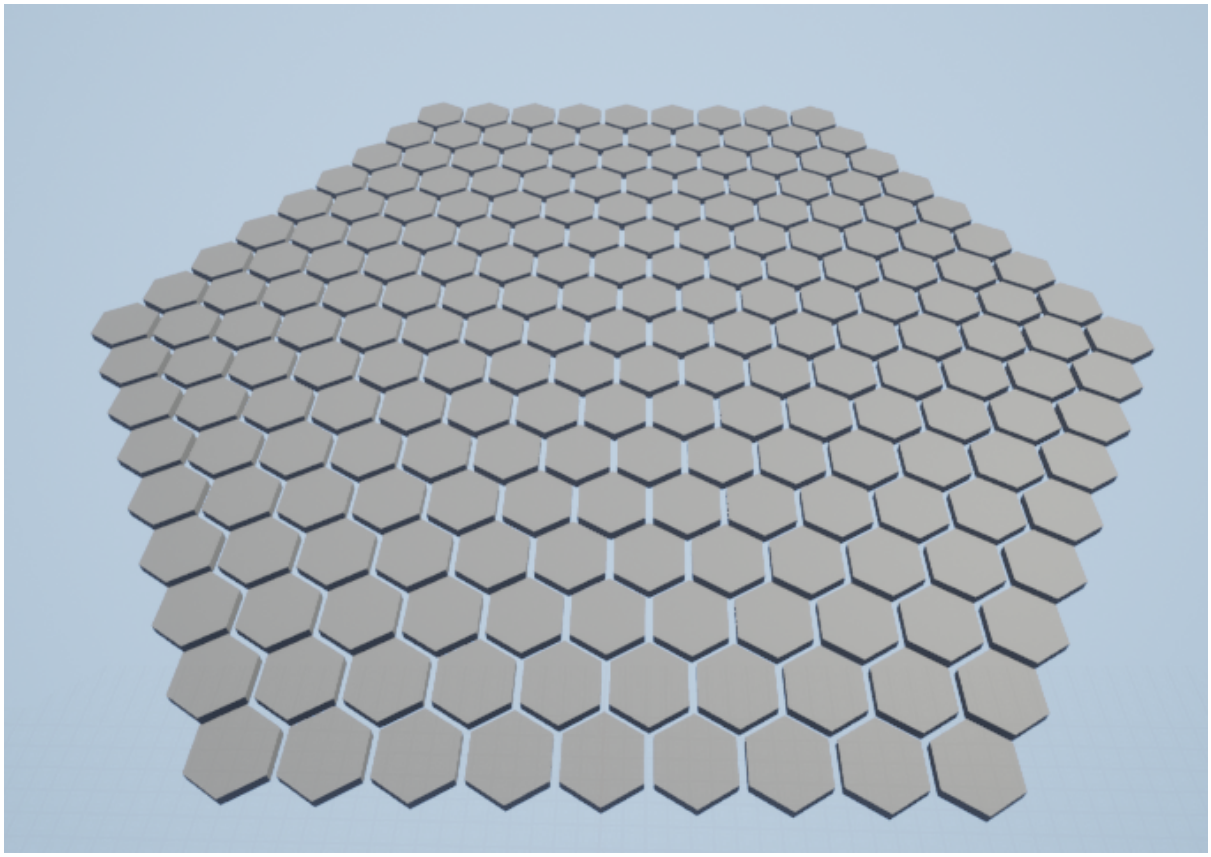
At the start of play, this spawns the Hex Manager actor.

The HexManager spawns a map of radius N.

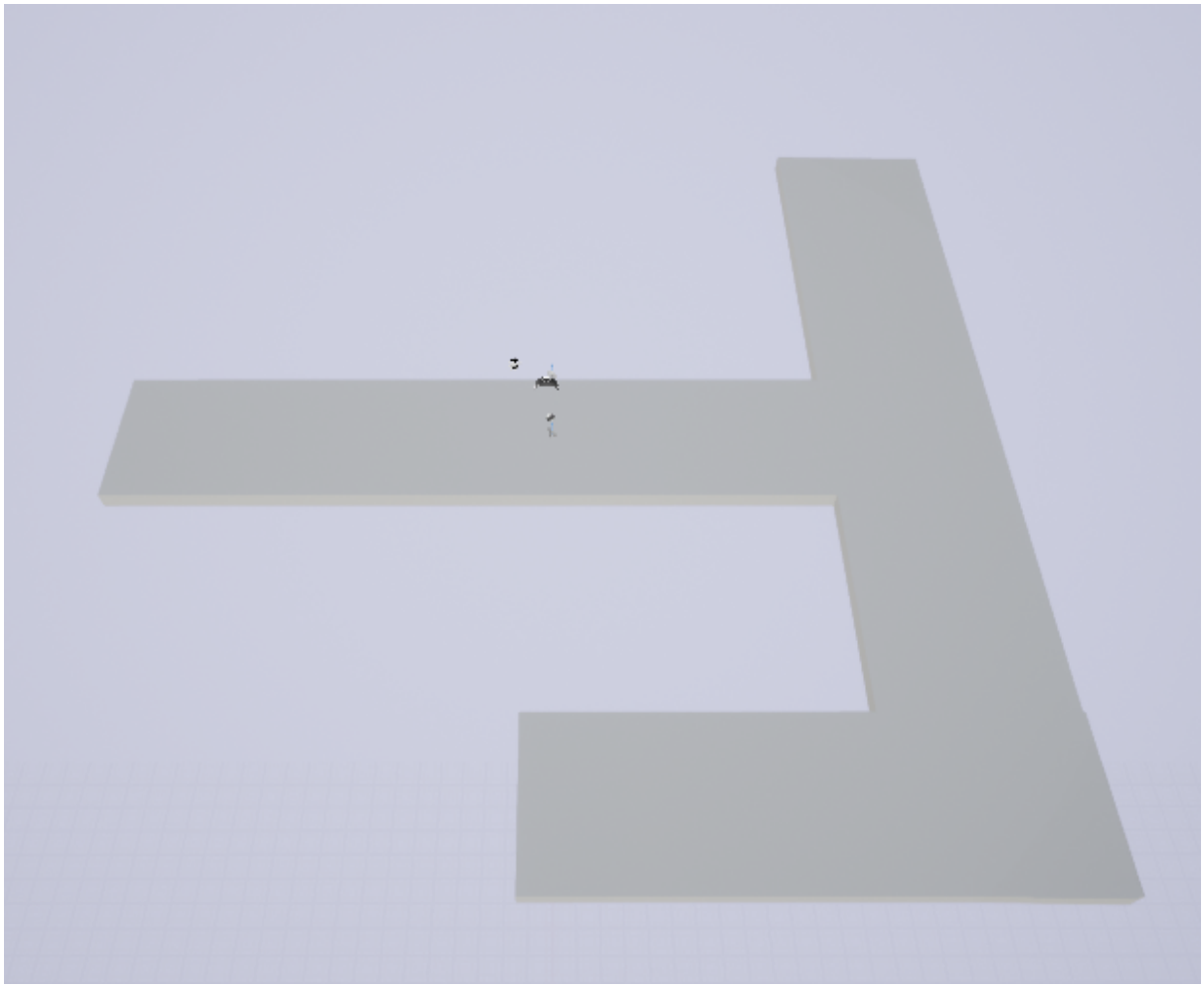
Then, hexes can be set to autodestruct if not enclosed by a map overlapper. This allows map design to function on top of the procedural generation.



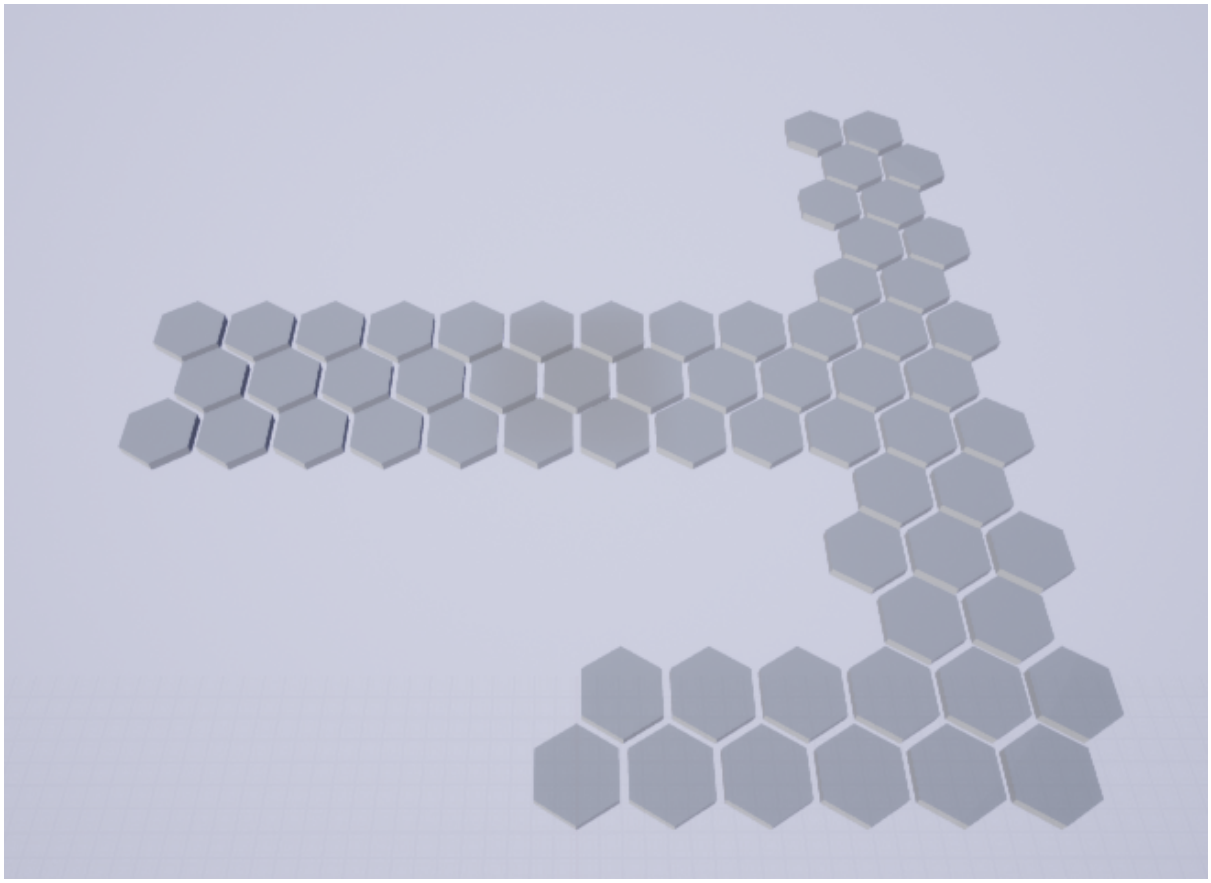
Hex map of radius 4.



Hex map of radius 8



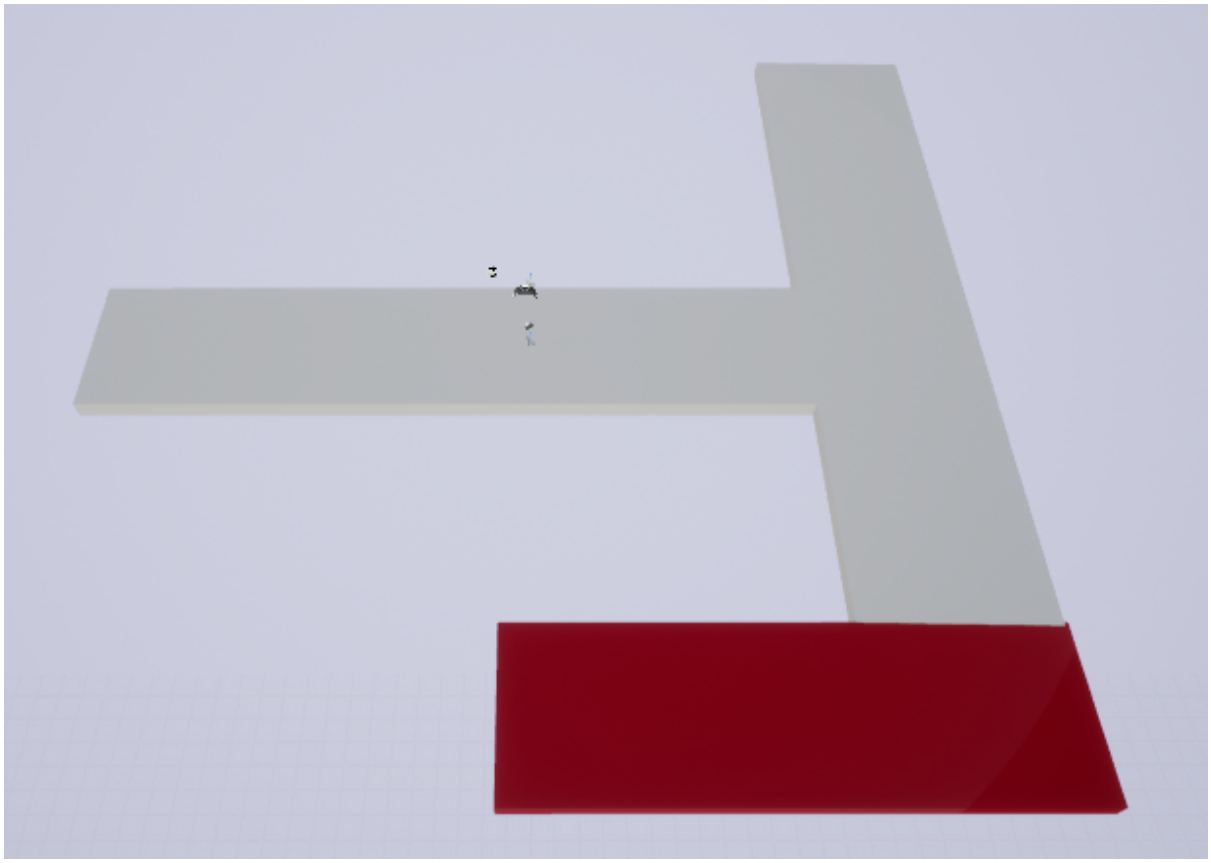
Manually placed level design blocks

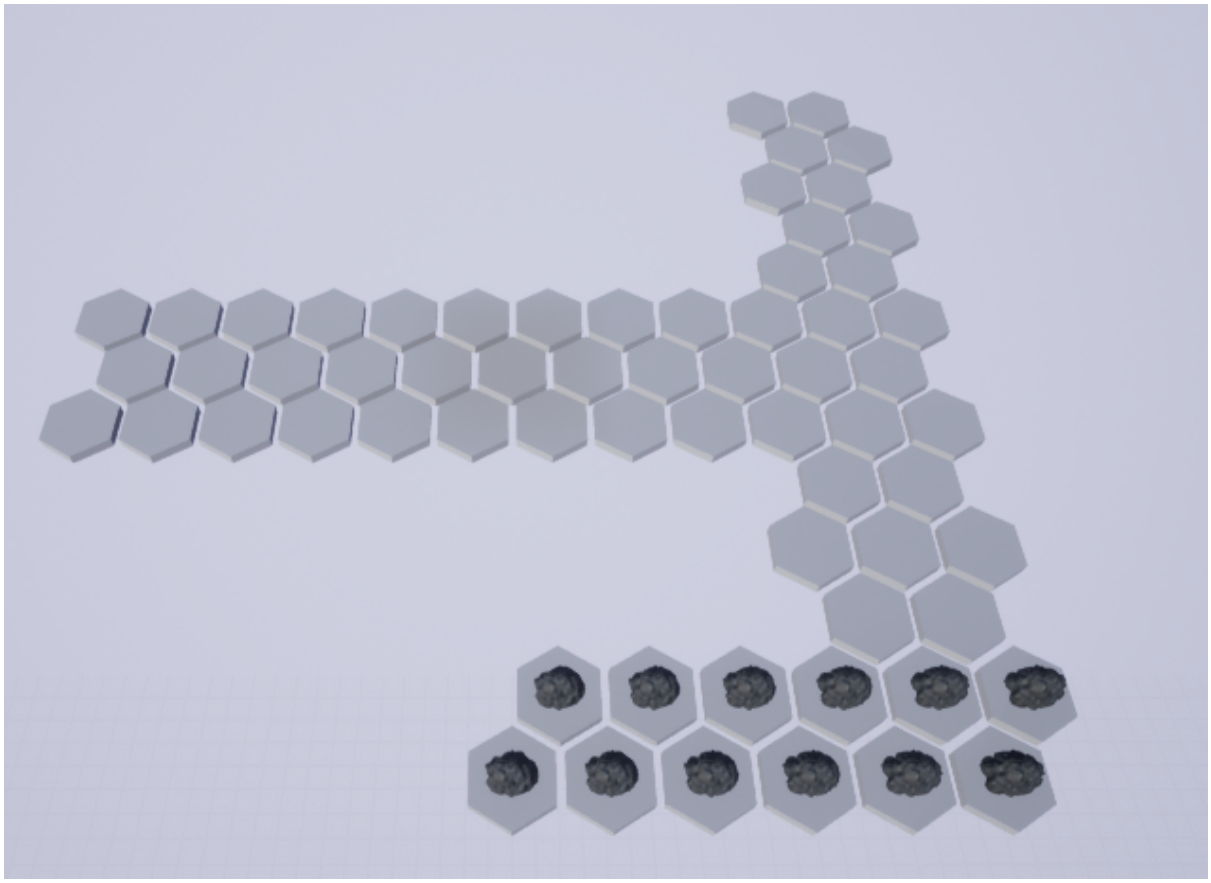


Corresponding procedurally generate Hex map

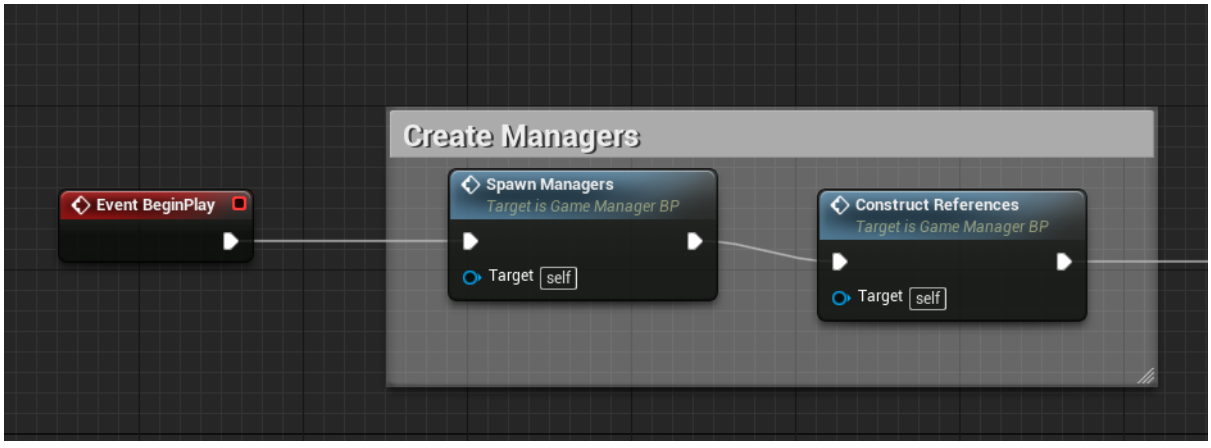
The blueprint also contains a movement cost mechanic for pathfinding, and a hex can be set with a different movement cost with a different type of overlapping square.

This is visualized here as marked with a rock.

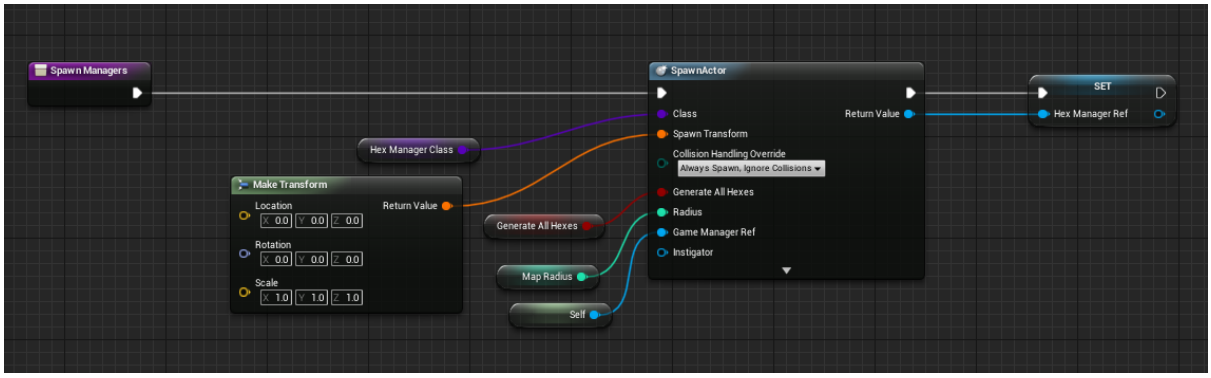




Inside the GameManager event graph

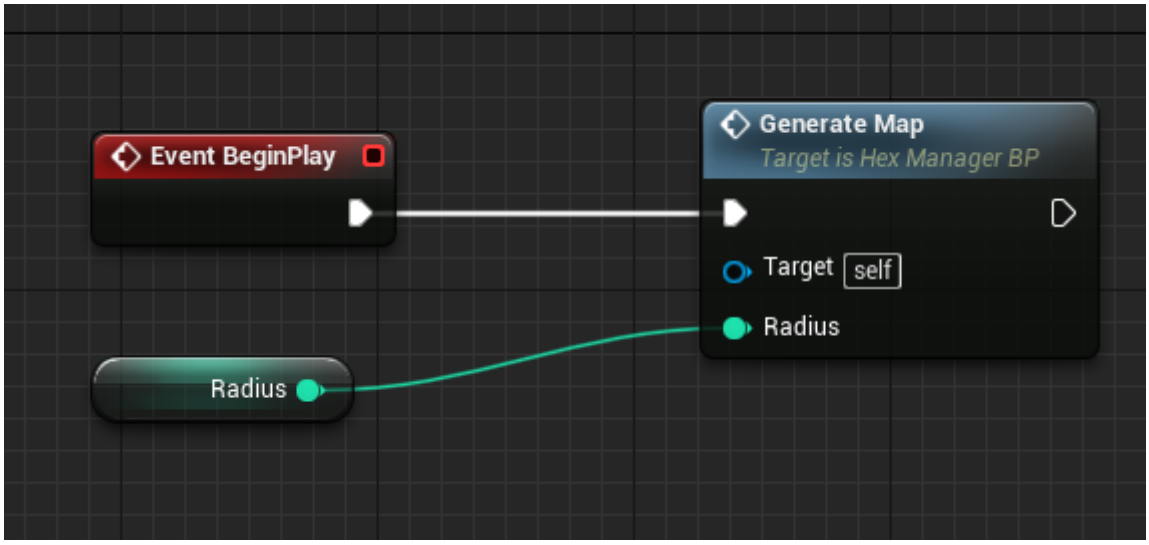


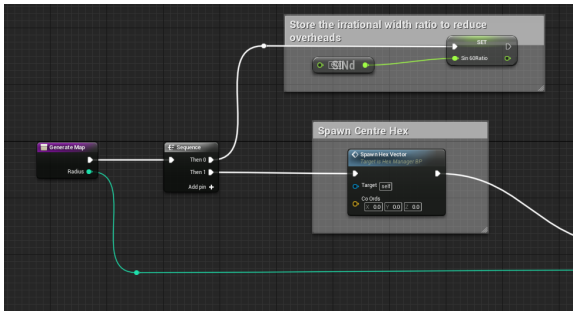
Inside of Spawn Managers (hex manager only)

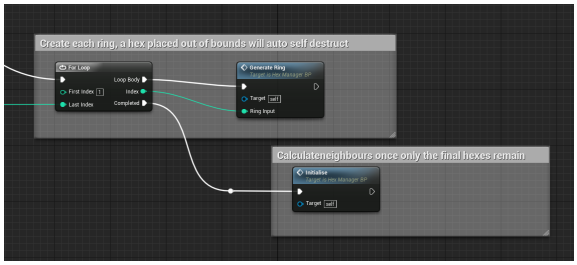


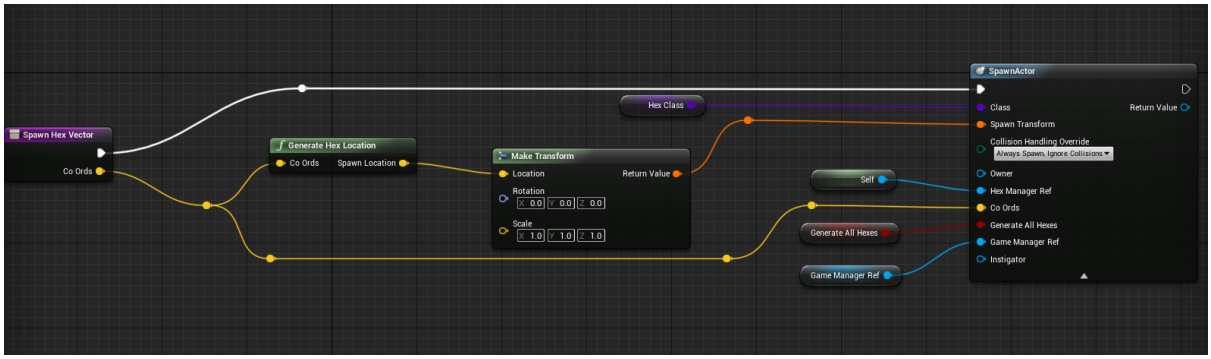
Construct references passes the managers to each other so that they can interact.

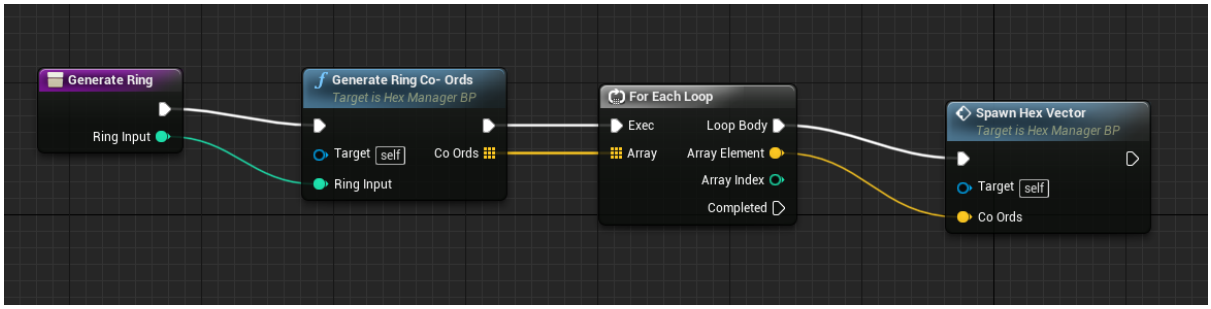
Event graph of HexManager

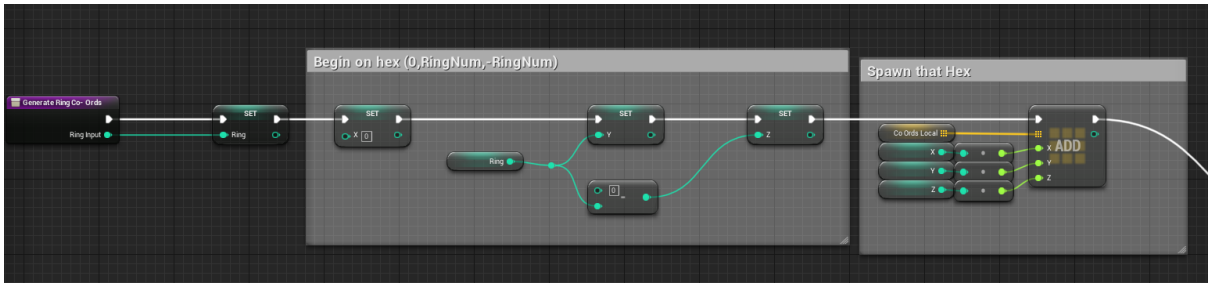


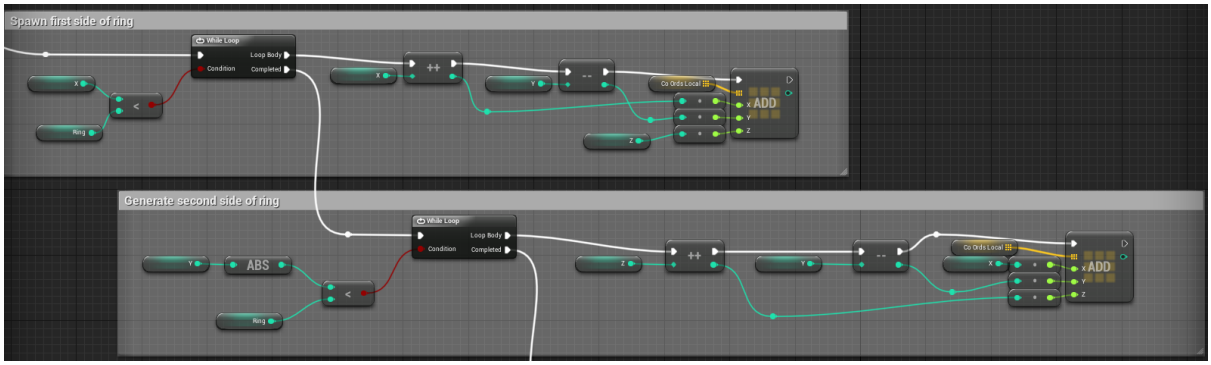


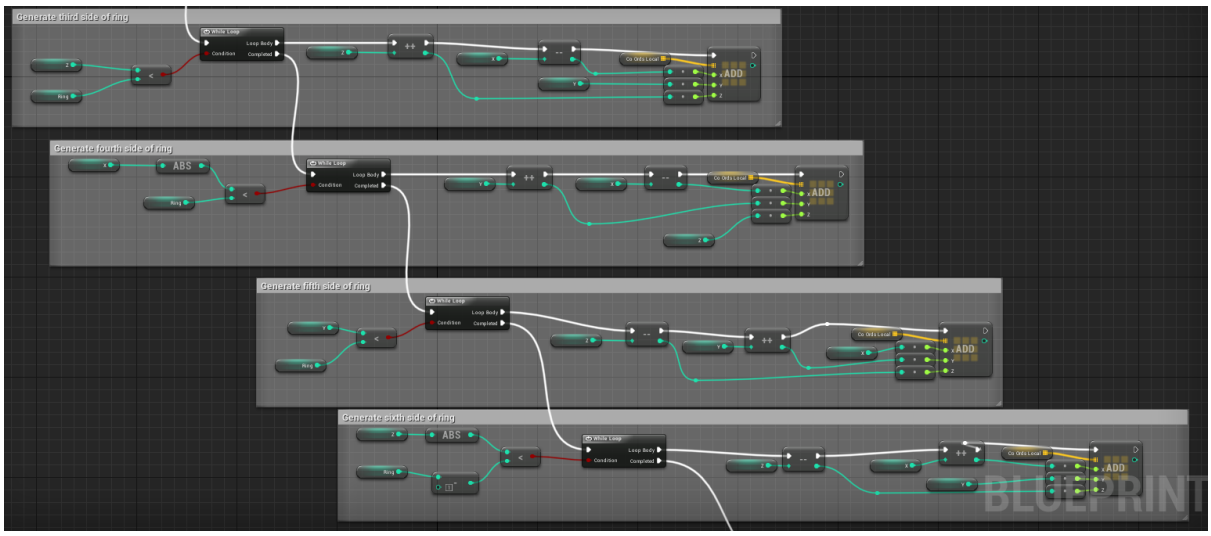


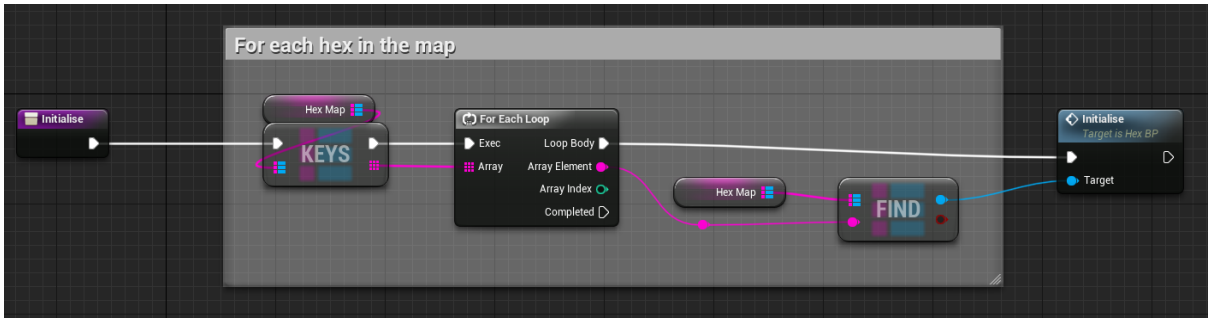

Spawn Hex Vector
Co Ords
Generate Hex Location
Co Ords
Spawn Location
Make Transform
Location
Rotation
Scale
Return Value
Hex Class
Self
Generate All Hexes
Game Manager Ref
SpawnActor
Class
Spawn Transform
Collision Handling Override
Owner
Hex Manager Ref
Co Ords
Generate All Hexes
Game Manager Ref
Instigator
Return Value


Generate Ring
Ring Input
Generate Ring Co- Ords
Target is Hex Manager BP
Target
self
Co Ords
Ring Input
For Each Loop
Exec
Array
Loop Body
Array Element
Array Index
Completed
Spawn Hex Vector
Target is Hex Manager BP
Target
self
Co Ords

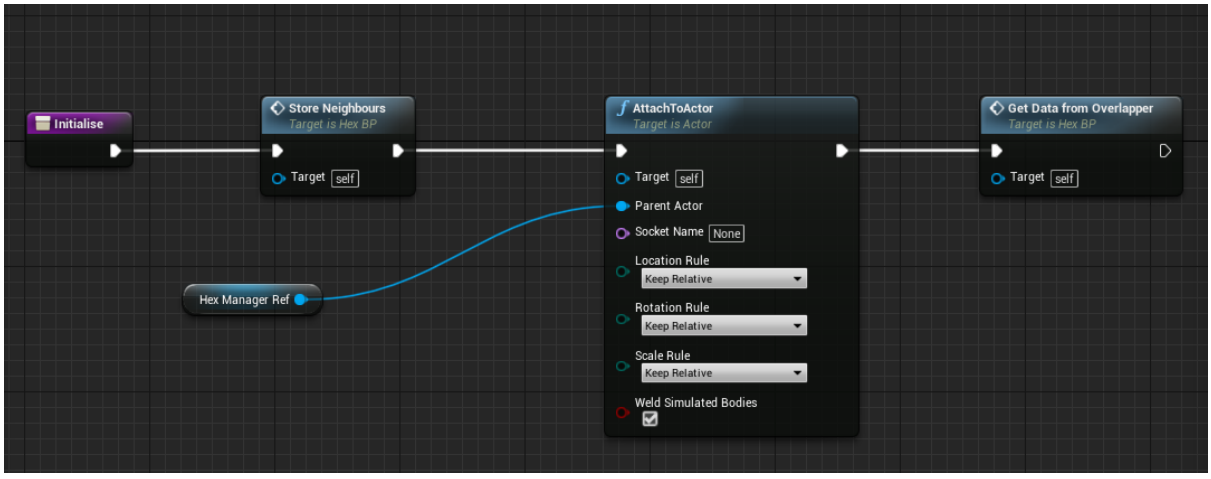

Begin on hex (0,RingNum,-RingNum)
Spawn that Hex
Generate Ring Co- Ords
Ring Input
SET
Ring
ADD

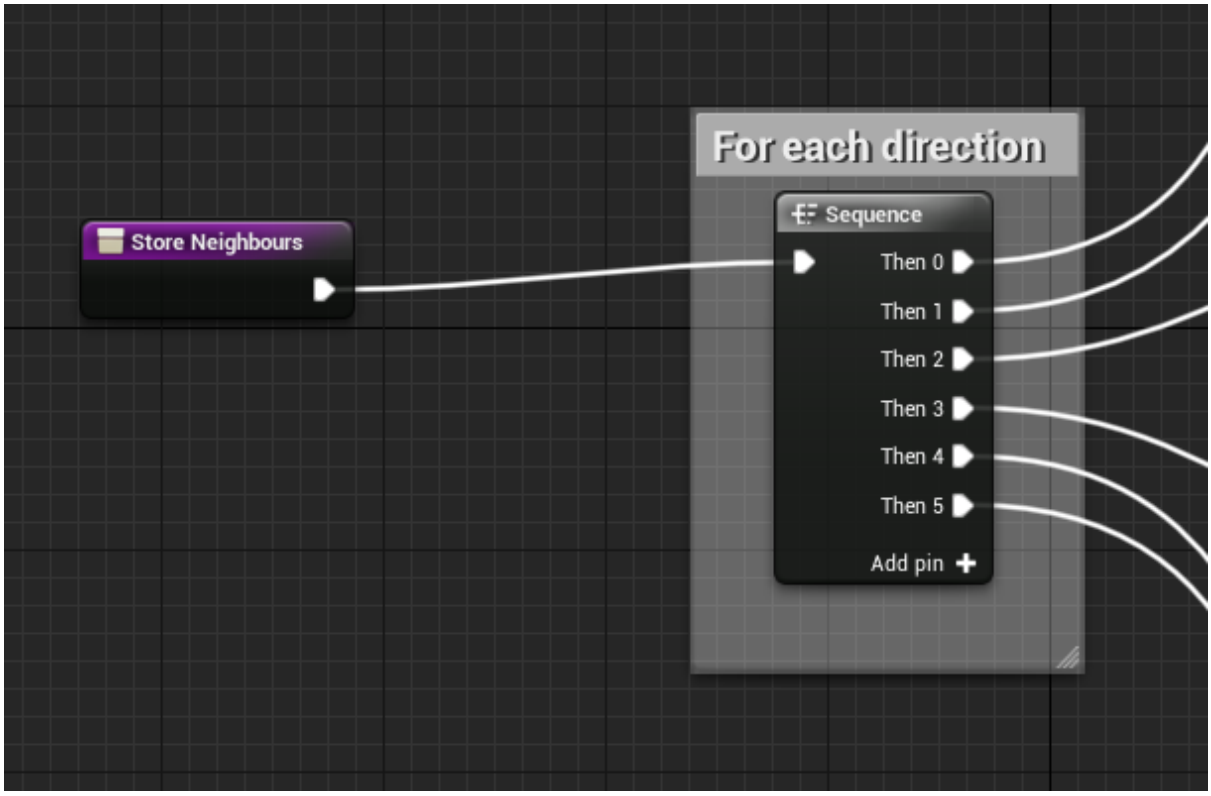

Spawn first side of ring
Generate second side of ring
While Loop
Loop Body
Completed
ABS
ADD

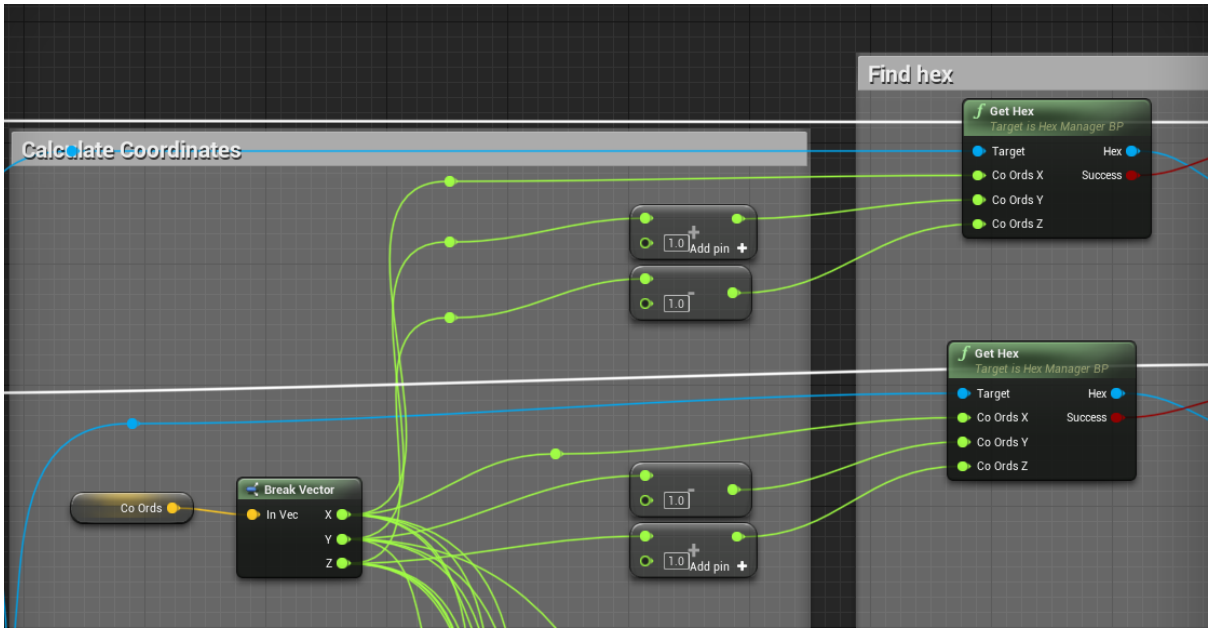


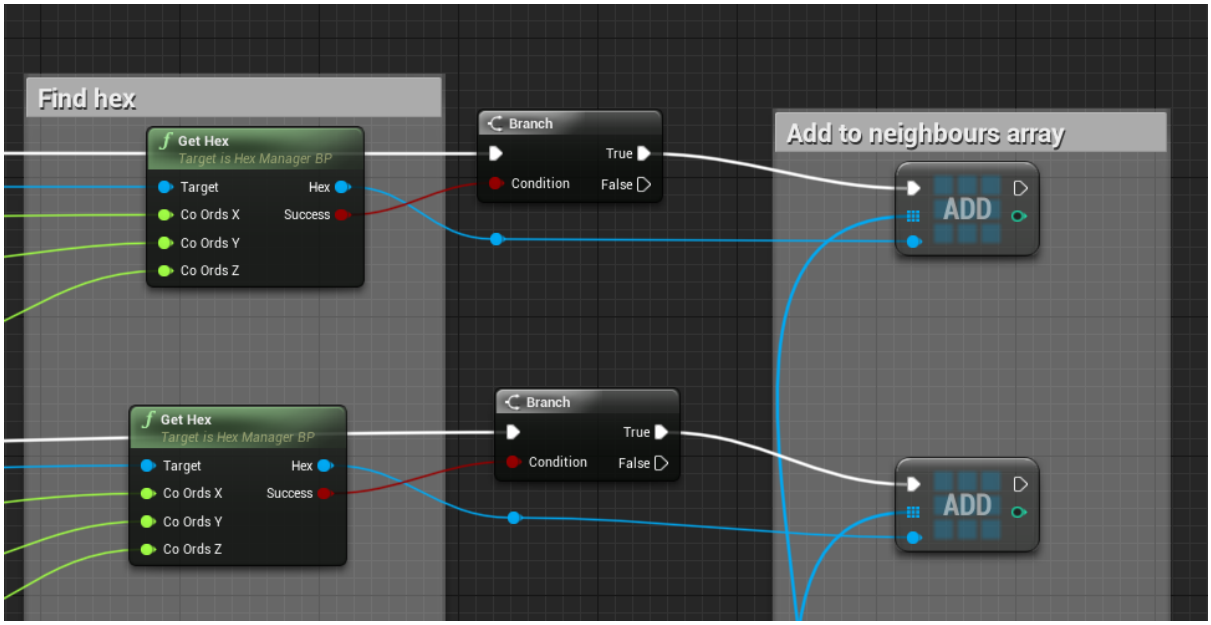


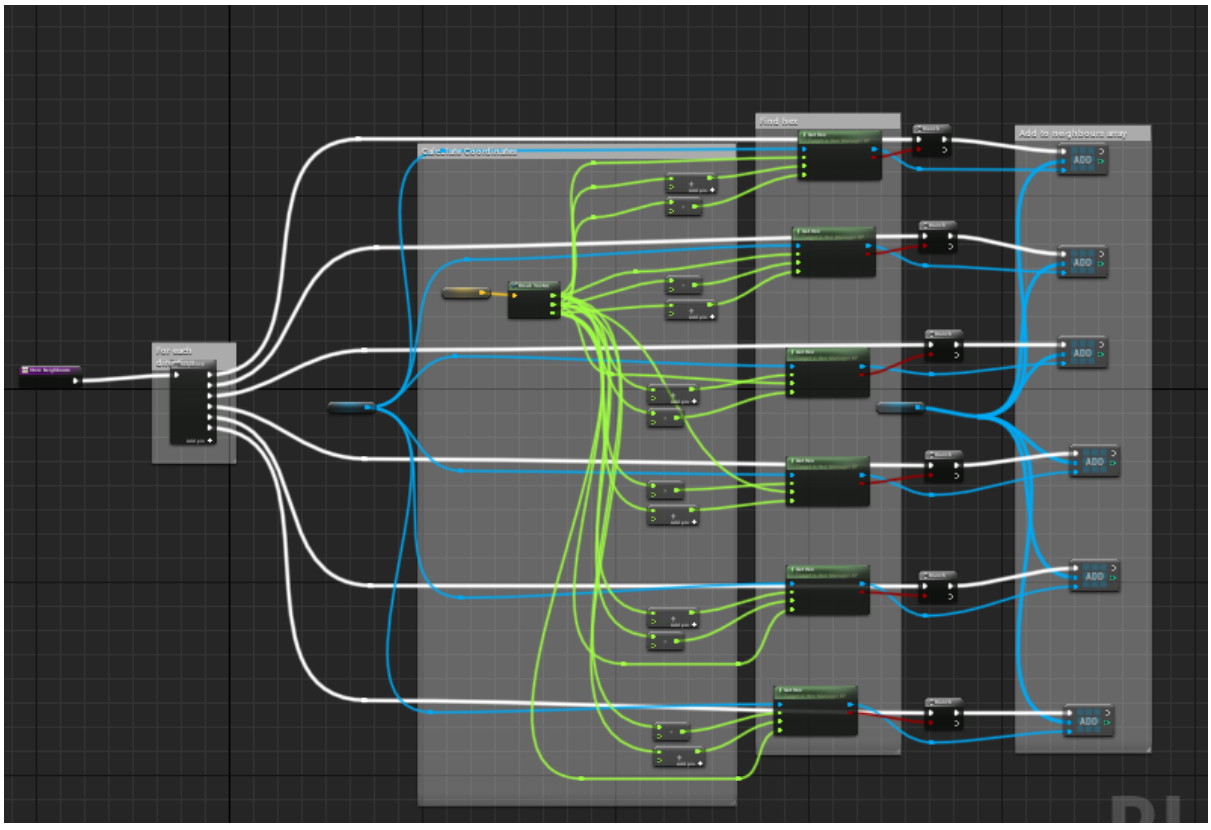
Initialise is a method of an individual hex

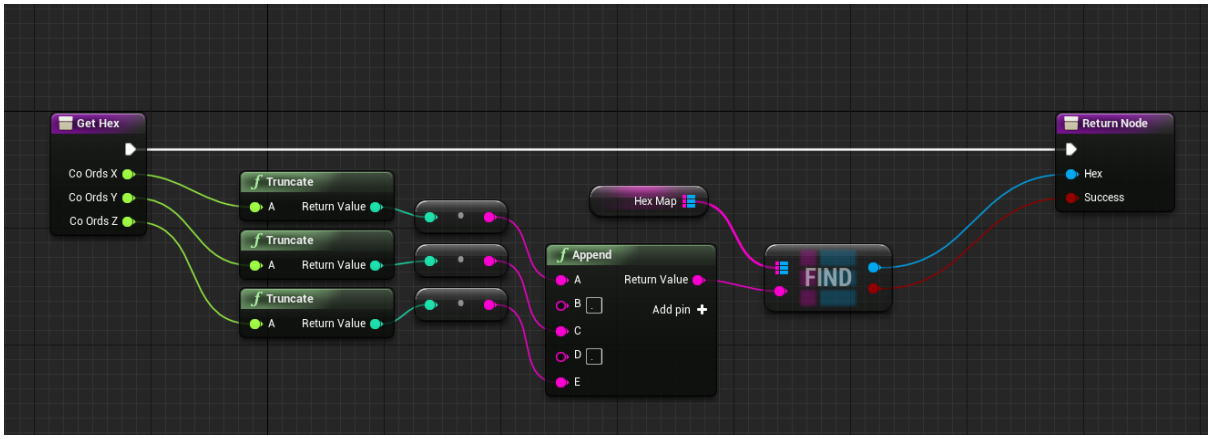


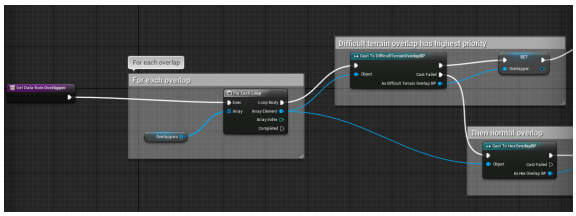

For each direction
Store Neighbours
Sequence
Then 0
Then 1
Then 2
Then 3
Then 4
Then 5
Add pin

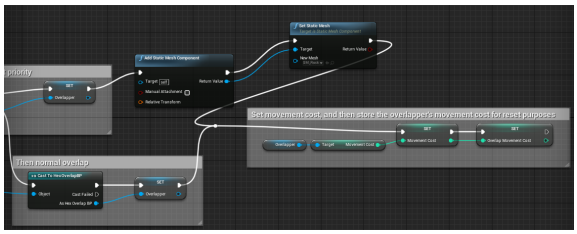

Find hex
Calculate Coordinates
Get Hex
Target is Hex Manager BP
Target
Co Ords X
Co Ords Y
Co Ords Z
Hex
Success
Add pin
Break Vector
In Vec
X
Y
Z
Co Ords


Find hex
Get Hex
Target is Hex Manager BP
Target
Co Ords X
Co Ords Y
Co Ords Z
Hex
Success
Branch
True
Condition
False
Add to neighbours array
ADD
Get Hex
Target is Hex Manager BP
Target
Co Ords X
Co Ords Y
Co Ords Z
Hex
Success
Branch
True
Condition
False
ADD




Get Hex
Co Ords X
Co Ords Y
Co Ords Z
Truncate
A
Return Value
Truncate
A
Return Value
Truncate
A
Return Value
Hex Map
Append
A
B
C
D
E
Return Value
Add pin
FIND
Return Node
Hex
Success


For each overlap
For each overlap
Difficult terrain overlap has highest priority
Then normal overlap


priority
Then normal overlap
Set movement cost, and then store the overlapper's movement cost for reset purposes