Evolving Asteroid Starship module location assignment schematic.

software:UNITY3d

DRAFT

DATA

Mockdata

	A	B	C	D	E	F	G	H	I	J	K	L
1		tmp	mining module	ore storage	processing module	refined storage	printing robot	manufacturing module	equipment storage	assembly robot	habitation module	life-support module
2	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	1	1	1	1	1	1	1	10	1	1
4	3	3	1	1	1	1	1	1	1	10	1	1
5	4	3	1	1	1	1	1	1	1	10	1	1
6	5	5	1	1	1	7	1	1	1	10	1	1
7	6	6	1	1	1	7	1	1	1	10	1	1
8	7	3	1	1	1	7	1	1	1	10	1	1
9	8	8	1	1	1	7	1	1	1	11	1	1
10	9	9	5	5	1	7	1	1	1	11	1	1
11	10	10	5	5	1	7	1	1	1	11	1	1
12	11	11	5	5	1	7	1	1	1	11	1	1
13	12	12	5	5	1	7	1	1	10	11	1	1
14	13	13	5	5	1	7	1	1	10	11	14	1
15	14	14	5	5	1	7	1	1	10	11	12	1
16	15	15	5	5	9	7	1	1	10	11	12	1
17	16	6	5	5	9	7	1	1	10	11	12	1
18	17	17	5	5	8	7	1	1	10	11	12	1
19	18	18	5	5	8	9	1	1	10	11	12	1

Real data

model load at startup (may become dynamic)

Locations (geometry)	Asteroid Volume	Asteroid Surface
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3	vector3	vector3
vector3		
vector3		
vector3		
vector3		
vector3		

PROCESSING

DATA
Processing one row of mock data during the main processing loop.

CASE(S)
Processing a single module leads to a case containing a set of rules.

RULES
A unique set of rules is applied within a case.
For example, a mining case consists of distancing, asteroid surface and geometry rules.
The list below serves as an example and is not complete.

STORAGE
The location(s) in the relay list are passed and transferred to the Build List after all applicable rules are applied.

BuildList

INSTANTIATION
Instantiation data (Vector3) is retrieved from the Build List. The instantiation script adds modules to xyz space for visual rendering.

mining
oreStorage
processing
refinedStorage
printingBot
manufacturing
equipStorage
assemblyBot
habitation
lifeSupport

x

cycled x times.
x = highest value in sample

mining
oreStorage
processing
refinedStorage
printingBot
manufacturing
equipStorage
assemblyBot
habitation
lifeSupport

Single module per case.

distancing	rule
nearby	rule
priority	rule
availability	rule
random	rule
etcetera	

vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3
vector3

EXAMPLE sequence:

DATA

10: 5,5,1,7,1,1,1,11,1,1

10: 4,5,1,7,1,1,1,11,1,1

1 (mining)

CASE(S)

mining

RULES

mining	rule

STORAGE

BuildList

INSTANTIATION

One row of data is loaded into a single array.In this case step 10 is used as an example.
In this arrangement, each module is represented by an integer, denoting how many modules of this type need to be generated.
A loop within the 'processing' script starts and will process a single unit at the time.
At the start, one type of module and only one unit is sent to the 'cases' script. Here it is possible to apply a certain order of module types or add a random order. One mining module in this example.
Once the 'mining module' is entering the 'cases' script, it will be processed by the 'mining' case.
The 'cases' script contains the set of rules for a single module to cycle through. Naturally, the set of rules is different for each type of module but most modules have a common set of rules like 'geometry', 'place availability' etc. In this case the mining module will have to pass the following rules: geometry, asteroid surface, asteroid volume, *
Between application of rules, the available locations are stored in a relay list. This list is carrying available locations from one rule to the other. In fact, the relaylist is cleared and then fed by a temporary list so it becomes a 'new' updated list for the next rule.
Once the module passes all rules, one of the available locations will be allocated. Here it is possible to select one of the locations randomly.
Once the location is chosen, it will be sent to the build list which contains all actual locations of existing modules.
The instantiation script will finally render the module or multiple modules, it has yet to be decided when this will happen with the option of rendering a single module or an array of modules.

Lists

RelayList is passed from rule to rule.
Temporary list is updating relayList.

RelayList		Temporary List		New RelayList
vector3	&	vector3	=	vector3
vector3		vector3		vector3
vector3		vector3		vector3
vector3				
vector3				
vector3				
vector3				
vector3				
vector3				
vector3				
vector3				
vector3				

July 27
To do:
logic lowchart
verification methods
additional rules
debugging
instantiation of specific types

availability rule

approved locations

vector3
vector3
vector3
vector3
vector3
vector3
vector3

Choose random location.

Location occupied (compared with Build List):

If true: return to last RelayList and choose other random location.

If false: add location to Build List.