

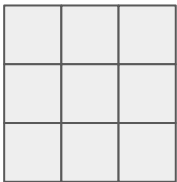
Procedural Generation BombSlinger

1. Division of the map
2. Arrange Enemies
3. Weight (Enemies - Rooms)
4. Step of implementation Weight (Enemies - Rooms)
5. MainPath Progression
6. Pick Up Chances of the enemies variations

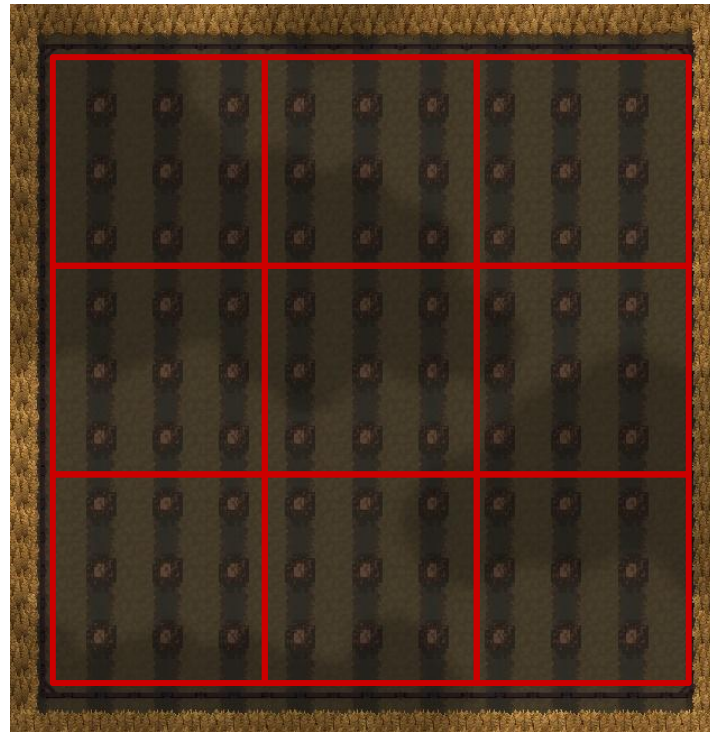
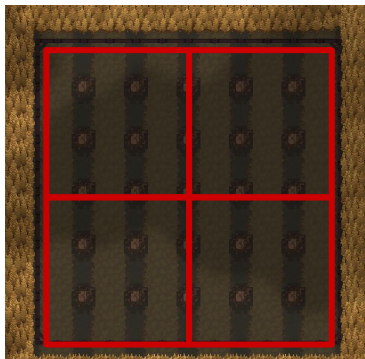
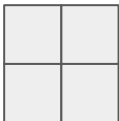
1. Division of the map

-> Get the size of the map

If the map is bigger or equal to 10x10, we divide the grid by 9



Else we divide the grid by 4



2. Arrange Enemies

First enemy :

-> We have now 4 blocks A-B-C-D.

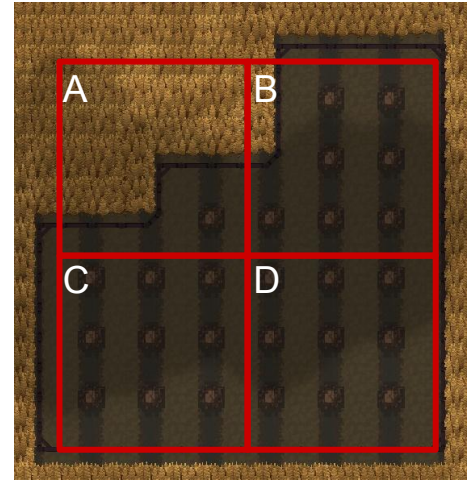
They are sorted in a list based on the amount of available spawn tiles.

-> Then we place an enemy randomly in the first of the list.

Rest of the enemies :

-> we place an enemy in each area first

-> We look at our ratio (enemies / Tiles available). And we put the next enemy in the one that has the lower ratio.



3. Weight (Enemies - Rooms)

ROOM :

The level is generated by the size of the room.

The weight of the room is defined by : the amount of free_Tiles / 5, round to the lowest number, then we add a number which depend on the level we are in.

M = Multiplier, to change but for now should be around 2.5 or 3.

X = additional number if we want to push the difficulty.

$(\text{Free_Tiles} / 5) + \text{Level} * M + X = \text{Weight} \rightarrow$ round to the lowest.

ENEMIES :

Each enemies has an weight_Value defined in a preset file, value can be found here :

https://docs.google.com/spreadsheets/d/1CiG8ud8A1ks5tukNu8w6SHrY-gxgosGB9_S5K7jkLk0/edit#gid=0

The presets for each levels are :

- % of appearance of each enemy
- the amount of each enemy /room.

The image shows a digital formula editor interface. At the top, the equation $\left(\frac{a}{5}\right) + (b \cdot 3)$ is displayed. Below the equation, there are two rows of sliders. The first row is for variable 'a', with a value of 100, a range from -10 to 150, and a play button icon. The second row is for variable 'b', with a value of 4, a range from -10 to 10, and a play button icon. A small box on the right shows the result of the calculation as 32. Each row has a close button (X) in the top right corner.

4 Step of implementation Weight (Enemies - Rooms)

For the room side :

- The formula of each room.
- Check with the safe zone if free_Tiles in safe zone are not taken.
- Code implementation after the destructible generation

For the enemy side :

- Padding in between each enemies.
- Create a preset file were we can put % of appearance of each enemies.

-> 1. The enemies with a different amount

-> 2. The enemies with percent of appearance

- Limit the amount of each enemy in rooms.
- Check if they are enemies in the room before adding more enemies,

-> 1. Check if there are enemies.

-> 2. Get the amount of free tiles he's using (with the padding).

-> 3. Add that amount to the total free tiles.

-> 4. Get the formula for the room_Weight.

-> 5. Subtract to the room_Weight the weight of the enemies.

Preset Type Example

LEVEL 1		
Name	%	<= X / room
Farmer	25	4
Pyjama	20	4
Goat	20	4
Redneck	20	3
Evil Goat	0	2
Bombthrower	15	2
Bandit	0	4
Coyote	0	4
Gattling Gun	0	3
Golddigger	0	2
Trapper	0	4
Wolf	0	3
Shaman	0	2
Boar	0	3
Native	0	3
Gattling Big Guy	0	3
% per level	100	

5. MainPath Progression

The progression of the difficulty is depending on the main Path.

More closer you get to the boss, harder the difficulty should be.

You don't want to have difficult enemies in the first room you encounter.

-> The presets will change with it.

---> you can check in the presets file which enemies can appear in which zone.

Example :

-> Pyjamas can be spawned in zone : 1, 2, 3.

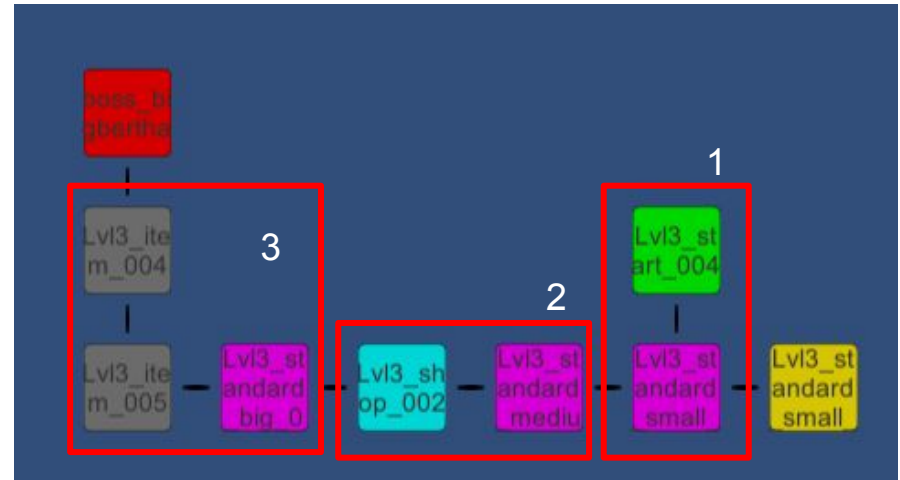
-> Torch Throwers can be spawned in zone : 3.

-> Goats can be spawned in zone : 2, 3.

How it works ->

Get the Main Path -> $\text{amount_of_room_minimum_to_boss}/3$.

For rooms that are not on the main path : check the zone that room that generate the connection.



PYJAMAFARMER	v
Spawn Chance	33
Spawn Max	4



6. Pick Up Chances of the enemies variations

1	Name	Monster difficulty	% at level 1	% at level 2	% at level 3	% at level 4	<= X / room
2	Farmer	2	25	3	5	5	4
3	Pyjama	2	30	0	3	2	4
4	Goat	2	20	0	0	0	4
5	Redneck	3	15	7	2	3	3

In some rooms to avoid the same feeling over and over all the rooms.

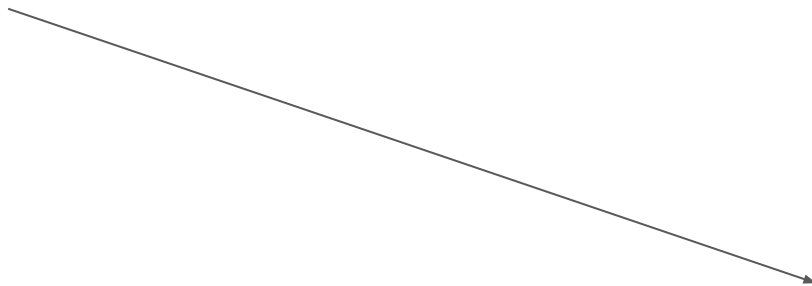
Once you enter a room one of these two systems is picked randomly



1 system - More diversity

$20\% / (3/\text{room}) \rightarrow 15\% / (2 \text{ left to pick}) \rightarrow 5\% / (1 \text{ left to pick})$

Current % = Previous % / (max enemies of that type per room - enemy of the type in the room).



2 system - More focus on one enemy

$20\% / (3/\text{room}) \rightarrow 40\% / (2 \text{ left to pick}) \rightarrow 60\% / (1 \text{ left to pick})$

Current % = Previous % * (max enemies of that type per room - enemy of the type in the room)

References

Combinations Math - <http://villeminegerard.free.fr/Denombre/Cbingene.html>

<http://villeminegerard.free.fr/Denombre/CombBino.htm#relation>