

CONCEPTUAL FRAMEWORK FOR SELFREP DOOMSDAY

BY DOKIMI

WHAT IS THIS DOCUMENT ABOUT?

This document is aimed at helping anyone who wants to develop this mod further. You might also find this interesting if you just want to understand how this mod works.

Here I will discuss the core principles and processes that make this mod function. Anyone who wishes to expand this mod will need to understand this conceptual framework.

I will not bore you with line by line code, because, to be honest, you can probably figure out better ways of coding this than I have. Instead these are the key challenges your code must solve.

WHAT IS THE AIM OF THIS MOD?

All the mods in the Self Organizing Systems modpack aim to do two things:

- 1) explore the principles of self organization that underlie so many natural phenomena.
- 2) Do this exploration in a fun way.

If you want to expand this mod then let those be your goal too - be scientific, and have fun. This is my ideal: these mods should be a fun way to spend some time, and be an engaging way to learn about some otherwise very abstract concepts.

Selfrep Doomsday itself is about exponential growth.

People often underestimate just how powerful exponential growth is. These “Doomsday” devices should cure you of any misconceptions. Seeing a single block grow to eat the entire universe can do that.

SOME BACKGROUND ABOUT SELF ORGANIZATION

What is self organization?

Order from chaos. Spontaneous. No guiding hands. No designers. Just interacting components coming together by themselves by following simple rules. All bottom-up. Ant colonies, enzymes, ecosystems, economies, your own brain. All have arisen through self organization. It is why nature works.

People have been using computers to understand self organization for a while.

The inspiration behind all the Self Organizing Systems mods was cellular automaton. These computer programs consist of grids of cells, where each cell can exist in a set number of states. These states change based on what states the other cells are in. The result is highly complex behavior arising out of very simple rules. Some people have even argued that the entire universe is a cellular automaton.

At this point you might be realizing that Minetest also consists of a grid of cells... and that these cells can be programmed to change state based on the state of the cells around them... and therefore...

All of Minetest, it's entire near infinite world, it's huge range of possible blocks, can be turned into a giant cellular automaton. This is an incredible opportunity to explore complex self-organizing behavior in a fun and very hands on way.

That, in essence, is what the Self Organizing Systems mods aim to do.

Where does Doomsday fit? Growth and self replication is a common feature in self organizing systems. And when you have growth, you have the potential for exponential growth. This mod is something of a twin to the Selrep mod (they started as one mod). In Selfrep, growth is so limited it never gets anywhere. In Doomsday growth is so uncontrolled it gets everywhere.

THE KEY CONCEPTS FOR SELFREP DOOMSDAY

Exponential growth is easy to do. Just let a block self replicate... and you have instantly lost control of it.

Replication can be easily programmed by having the block find a position nearby, and changing that position into a copy of itself.

However, this in itself isn't very interesting. You'll probably end up with a giant blob. And you'll get so many active blocks that your computer will soon grind to a halt.

Doomsday solves these issues a few ways:

- Placing conditions on replication.
- Having multiple growth forms.
- Additional powers beyond replication.
- Destroying the replicator blocks.

REPLICATION CONDITIONS

When, where, and how can a block replicate?

How you answer these questions will have a big impact on the resulting structure you end up with. Any condition you can think of you could set, but these are some that I've used:

LOCAL POPULATION LIMITS

This can control the shape of structures. The danger with using this is that you will kill exponential growth. The population limit must be well above this value:

$$p = mr - d$$

Where:

p = the population limit at which replication is stopped

m = how many times larger than the radius the population can grow

r = the radius around each bot which counts as being within its population

d = the dispersal radius, i.e. how far away a new bot can be spawned from its parent.

This seems to be the threshold between static structures and runaway growth (I think... there's a mathematical relationship hiding in there involving m that I don't fully understand, but it seems to work).

As for shape:

- P will control density. For example, with a 1 block radius around the replicator (i.e. a 3 x 3 x 3 cube) the maximum number that can physically fit in there is 27 blocks. If $p = 27$ then that entire cube will get filled in. Below this and it will be forced to leave gaps.

- D will also have an impact on density at the growing points. If it is above 1 then the replicators will be able to "jump". Changing d is useful if you want to control the growth's ability to cross barriers.

Population limits can also be applied to block types other than the replicators themselves. For example you can force them to leave a minimum amount of empty space, or be inhibited by the presence of dead replicators, etc. I used this method to help force the Grey Goo to create an open structure, preventing it from filling in every last gap.

SUBSTRATES

What kind of blocks can the replicator replace with itself? This will decide where the growth will go and how easily it is controlled. For example:

- the Weapon can spread through everything except air (and ice). As a result it eats mountains and digs huge holes. It is practically unstoppable.

- the Blight can only spread through "organic" matter. It comes to a stop as soon as it reaches anything else, like an ocean, or sand. Barriers can be made to keep it in.

ENVIRONMENTAL REQUIREMENTS

Do the blocks need access to something in order to replicate? Air, water, sunlight, human flesh?

This will decide both where it can grow, and it's shape. These needs can be used to make the structure grow appropriately for the conditions. e.g. The Grey Goo's water growth consists of columns which give a nice sense of the whole thing being held up by roots.

ATTACHMENT

If your dispersal is set to 1, i.e. children are spawned right beside their parents, then you must decide how those two blocks will be attached to each other.

Any random side? Only on top? The bottom? Horizontally?

This will determine whether it grows pillars, sheets, or blobs.

You could also set it to only attach to certain other kinds of blocks, so that it only covers over sand, or climbs up stone cliffs etc.

RANDOM GROWTH VERSUS DETERMINISTIC GROWTH

Does a new block always go in the same place? Or is it random?

Deterministic growth will give crystal like structures, whereas random growth looks more organic.

MULTIPLE GROWTH FORMS

One type of growth can get repetitive. In order to give the structure a wider repertoire of abilities, it can be allowed to replicate in many different ways.

e.g. The Grey Goo can form platforms in sunlight, and pillars in the dark. The result is more variable, interesting structure.

Note, differing growth forms will compete with each other for space, especially if population limits are used. The more you allow of one, the less you'll get of the other. This can be useful if you only occasionally want it to use a different growth form.

ADDITIONAL POWERS

Blocks can be used to do more than just replicate itself. They can place other blocks, remove them, whatever you can think of.

e.g. the Terraformer will plant trees all around it, using much the same method as it does for replicating itself. This is essential to its function of leaving behind a rich green landscape.

REPLICATOR DEATH

Performance is an issue with these exponential structures, a somewhat unavoidable side effect - if it wasn't on its way to infinity it wouldn't be exponential. However without clearing out some of the blocks, performance would be impossibly bad.

Here you have to decide what you want left behind once the Doomsday device has swept through. Utter annihilation? A mountain? A forest? Piles of paperclips?

Again we can leave multiple things behind, and we can make this process of block death multi-staged.

e.g. the Weapon decays first to flaming liquid which ignites the TNT it placed earlier and burns anything in the way. This then decays to a corrosive gas which floats around mopping up any surviving blocks the earlier ones missed. Finally the gas vanishes. Each step is used to enhance the weapon's goal of destroying everything in existence.

Death also allows us to create a kill switch. By allowing the user to set the death rates very fast (and/or growth rates slow) the Doomsday device can be destroyed. A kill switch is essential for making this mod user friendly. Does anyone ever really want to burn the *entire* world? Devices that leave behind structures, like with the Grey Goo, are actually most interesting after they are dead.

For exponential growth to still occur, death must not be allowed to balance or overwhelm growth. Replicators must be allowed to spread fast enough so that they get to new areas before the death cycle kicks in.

CONCLUSION

Ultimately Doomsday is very simple.

- 1) allow a block to replicate
- 2) decide how it will replicate
- 3) decide what else it will do while it's alive
- 4) decide how it will die.

These devices can be made as simple or as complicated as you like. Either way their enormous growth is always an amazing sight.