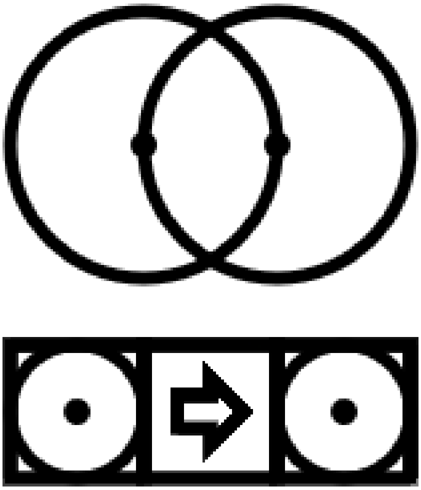
Geometron

Social media for a post-scarcity world

Lafe Spietz



Geometron
Social media for a post-scarcity world

Lafe Spietz

April 9, 2021

Contents

Contents **iii**

List of Figures **iv**

1 Civilizations **1**

2 Scrolls **10**

3 Maps **16**

3.1 Maps 16

List of Figures

Chapter 1

Civilizations

The nature of our present civilization is one of constant consumption. We mine or drill material out of the ground, re-shape it or burn it for fuel, and ultimately turn it into toxic which waste we dump back into the environment. During the process from mine to landfill, we exchange "money" and this is what we call the "economy". The things which can be exchanged for money are called "property".

We seek to build a new civilization which is based not on a stream from mine to landfill, but on closed loops of material in equilibrium with living ecosystems. We look to the technological civilizations of indigenous people as an example of what mature technology should look like. In mature civilizations, technology is all self-replicating.

An example of technology in a mature civilization is traditional wooden boats. If people live in equilibrium with a forest, as old trees die or are harvested new trees grow. People carve logs into boats, and then use them to hunt and sustain their communities. A new generation of humans is born from the old, and they are taught all the skills to copy the construction of the boat, the stewardship and harvesting of the trees, and how to pass along the knowledge of boat building(as well as the hunting and fishing that sustains the whole society) to the next generation after them and so on. Unlike a consumer civilization, this type of self-replicating organic technology can continue in theory indefinitely. If it has the resilience built in to improve and change over time in response to changes in the ecosystem it can truly sustain itself without limits. Mature civilizations like this have existed for thousands of years all over the Earth.

Three things to note about a technology such as the one described above: everything replicates, everything can be modified, and everything dies. Every wood boat will rot, and eventually become soil again in the forest which will produce more trees. The technology is soft enough that people can modify it, carve and repair it over time, change it as needed, and evolve it as needed in response to changing conditions in the environment or new innovations in the technology. And of course, every boat can always be copied freely. In this context the idea of property doesn't make sense. Nothing is permanent. Everything is constantly going through a cycle from soil to tree to boat to soil again, growing, evolving dying, and being reborn.

When we build an economic system based on self-replicating geometric constructions from trash and organic material, money becomes a completely unusable system. To illustrate this, we consider a very simple thought experiment. Suppose I take a pile of trash and transform that into a robot which builds more robots out of trash, and which has media encoded in it which instructs others in how to copy the whole thing. If I and 100 people go from trash pile to trash pile and run the replication, converting all trash piles into trash-fed robot factories, we have created an ever expanding amount of value, with no property or mined materials going in. A vast amount of economic activity is being created, but without a central bank we have no way to denote all this added value. And if we somehow created a mechanism with some innovative new banking system to create enough currency to represent the added value, the instant we went and replicated the system again we would find that again, the numerical currency system breaks down, as it fails to keep up with the replication of value.

This contradiction is precisely the situation we currently find ourselves in. A small fraction of the population, namely the software industry, the media industry and the finance industry, can use replication to create value from nothing. With billions of people having already bought smart phones or other networked devices, there is zero marginal cost to add some useful new feature. So when an app goes from running on just a single developer's computer to a whole software team to a beta test group to early adopters and ultimately to a billion smart phone users, this is replication of information, not actual production in the sense of a factory. Meanwhile, everyone else, who are compensated for labor or some type of physical material have to directly produce value to get money. As long as some people create value which freely replicates and others do not, with a finite money supply, over time a larger and larger fraction of money will always flow to the replicators rather than the people doing labor.

There is no way out of this trap. Not a new banking system, not a better government policy. The trap is the numerical aspect of money itself. No matter how fast a hypothetical banking system prints money and pushes it out into the economy, a fully networked replication-based society can create value faster, and continue to amplify the inequality until all value flows to the replicators. We have seen this during the 2020-2021 COVID-19 pandemic in the United States. The government poured trillions of dollars into "the economy" and at least half of it appears to have gone directly into the personal fortunes of people in the replication-based segments of the system. While these people also engage in massive exploitation of labor, this is beside the point. Labor can fight back and win some concessions from the capitalists, but as time passes, the power will always slip away as the replication rate of pure information-based value increases.

Some people have proposed universal basic income as the solution. This is especially popular with the group of people who have benefited the most from the current accelerating inequality: software engineers and their collaborators in the professional classes. In this system, the software people are allowed to accumulate unlimited and ever-accelerating wealth, and simply paying out money directly to everyone else will help the rest of humanity survive. But if a software person is making 1 million dollars a year, do you really want a 2k/month UBI? How about 5k/month? or 10k/month. . . but now the software people make $1 million/month? Is that really a working economy? If this sounds insane, look at what is happening to rents, look at who all the new housing is built for. This is happening now, but without any UBI to soften the blow, and it will continue to happen as long as we all cling to this failing economic model.

So what is the alternative? The alternative is a switch from an arithmetic to a geometric economy. This means that rather than an economy of exchange, where numbers are used to represent both property and labor, we have an economy of replication of geometric constructions. In this situation, the replication of information *is* the economy. It does not turn into numbers at any point. One can create a thing via a geometric construction from trash which contains information for its own replication. This then replicates out into the human network in a replication-based future descendant of the current Internet, and we all share the increased value. When we create value we do not "trade" it for something, because the amount of value will *increase* as it replicates, making any trade that truly represents the added value impossible. In a post-scarcity trash-sourced geometric economy, everyone benefits from the creation of creators because the best things naturally replicate to all of humanity, including the back to the creators themselves, but now amplified by the added benefit of billions of people potentially improving it.

The Geometron/Trash Robot system is an attempt to build an information technology system which serves this purpose: to build a purely geometric economy based on replication. In this system, everything replicates, everything evolves, and everything can be deleted at any time. In the beginning we source some parts from trash and some parts from common off the shelf consumer items that are easy to buy without any centralized company. In order to replicate successfully, however, this information networking system has to provide immediate value to the user, both within the existing money-based economy, and also outside that system.

In order for all this to work, we have to give up the three main elements of our existing system: mining, money, and property. As long as money exists, replication will break the economy. As long as mining exists, we will continue on a path to total destruction of the world and eventually scarcity and extinction. And as long as property exists, replication which could end all scarcity for all of humanity will be hindered by the people who control everything. The extension of property into the domain of pure information is another major reason that people in the replication industries have been able to so brutally exploit the rest of humanity. They claim to "own" most of the freely replicating information we rely on in the new economy, and they're effectively landlords who can create new land as many times as they want for free, but then can charge rent on all of it. Only by creating a new economic model from scratch which simultaneously abandons money, mining and property, can we build a just and sustainable future.

It is not immediately obvious that geometry can completely replace numbers-based thinking. But in most cases, our technology is already completely driven by geometry, we just choose not to look at it that way. What is a microchip? It is a geometric design imprinted in silicon. That design is made by a geometric program interacted with by a user thinking geometrically. Even the code itself, encoding into the physical circuits, is a geometric pattern of either magnetic or electric perturbations of physical objects. And "computers" are not used primarily to compute things, but to display graphical information, both text and pictures and graphics. The repeated actions of automation machinery represent geometric operations(e.e. move over, move up, move down, move over, etc. . .)

Saying "geometric economy" sounds abstract and to say it self-replicates sounds far fetched, but none of this is a new idea. Many of the oldest and most sustainable civilizations in the world have very very old technologies involving textiles made from natural materials woven into patterns. This is self-replicating geometry. It satisfies all the properties listed here: it dies of natural decay, is taught as a skill from the old to the young and is constantly replaced, and it can be modified and improved over time by the constantly-replicating group of experts in the technology within the community. So the transition to a geometric economy is not to some futuristic new idea but a return to ideas which predate the rise of industrial societies and nation-states.

All that said, we live in times different from any which came before. The Internet is the baseline information system now for all of humanity. Even the few people not connected to the Internet now have their lives completely molded by it, as all power networks exist on it now. And consumer society has injected every single element or type of finished product our civilization uses all over the planet. So while in times before now one group of people might discover the use of bronze and another of steel, now every single person on the planet can get aluminum sheet metal, titanium reinforced steel, rare earth metals, ultra high purity silicon, etc. All these incredible materials, already shaped and processed into the most useful form, are simply sitting in piles of trash in every corner of every nation on the planet now, all directly adjacent to nodes on an information network which connects to every other part of the planet.

This combination of universal networking and universal access to identical, standardized, trash elements creates a formula for building a new information system from scratch from which a whole stream of new civilizations can rise. It is our task to build a seed from which such civilizations can be built. Geometron is an attempt to build this seed.

Chapter 2

Scrolls

Scrolls are the text documents of Trash Robot. Think of this like the Microsoft Word of the Trash Robot ecosystem(with some drastic differences). Scrolls, along with maps, feeds, and symbols, form the basis of the user-facing system of documents which are shared on Geometron. They are used to document the system, to share ideas, post articles, create ads or lists of ads, or really any type of document one can imagine.

The format of scrolls can take some getting used to for people used to, as it is not WYSIWYG(What You See Is What You Get), but rather uses the Markdown language to create formatting with code. One of the first tasks of some enterprising new Trash Robot participant will be to create the fully WYSIWYG version of the scroll editor, but for now this is what we have. Part of the goal here is to have no documents ever be in a format other than human readable. Even if Markdown is a little awkward to read, a real live human can always read the text and someone with very very basic understanding of code can immediately turn it into fully a formatted document.

Symbol for scroll:

Create a scroll

To create a scroll, go to the scroll editor at scrolleditor.html, and enter the name of the new scroll in the input marked “new scroll name”. When you do this a blank document should appear with black background and green text(this is easy to change if you find it annoying). Just type out your document if it’s just text, hitting enter twice between paragraphs.

For further information on using the Markdown language, see

- the wikipedia page
- the official web page
- The Markdown Guide

The most annoying thing about markdown is putting images in, which you do as follows:

![](image url here)

Edit a scroll

To edit an existing scroll in the scroll editor, click on it or enter its name in the new scroll name input(no need for the “scrolls” prefix).

copy a scroll from another place online

All things in Trash Robot self-replicate and scrolls are no exception. To create a replicator we use the program copy.php which is on every TR server. To this scroll is called “scrolls” and to replicate it we make a link to “copy.php?from=[some web address of a trash robot server]/scrolls/scrolls&to=scrolls/scrolls”. If you run this on a server it will fetch this scroll and place it locally in the scrolls directory. Of course as with any replication this will overwrite the scroll on the local server so be careful, as any new edits on the new server will be lost.

To copy all the scrolls, as well as maps and other data, from another TR server, we use copydata.php. On whatever server you are on, make a link to “copydata.php?from=[the url of the domain from which you’re copying]. Note that for both copy.php and copydata.php the source domain can be the IP address of a TR server on your local network.

Also, the simplest way to copy a scroll is to open a new scroll on a new server in the scroll editor, then open the scroll to be copied in the scroll editor on the old server, and just select all, copy, and paste to the new one. This functionality is part of why having everything be in a human readable format like Markdown is important.

link to a scroll from a scroll

Links from a scroll to a scroll or from a map to a scroll can be realized in Trash Robot by having the target link be either "scrolls/scrollname" or "maps/mapname", and the code in the TR user page will convert those to local links. E.g. link to terminal scroll. Scrolls can also be linked to globally by using "user.php" with a scroll specified. For example to link to the Terminal scroll on trashrobot.org we link to `https://www.trashrobot.org/user.php?scroll=scrolls/terminal`.

mathuser.php

Delete Scrolls

Everything on every instance of Trash Robot can be deleted quickly and easily and with no backups. When something is deleted it's really gone. Rather than backing things up or saving to "the cloud", in Trash Robot we replicate what we want to keep and whatever doesn't get replicated will probably eventually be deleted. To get this functionality, we have a delete scroll page called `scrolldelete.html`. To delete, click on a red X. But this is for keeps! Deletion really is deletion. If you see some bad stuff on a server, just delete it. If you want to post stuff and not have it deleted, replicate it to a quiet place where no one will see it where it can get replicated back to a live page later if it's deleted.

Some technical details and use of Math

The basis of the scroll software is the JavaScript library Showdown.js, which is great, and it converts from markdown to html. So scrolls are all in raw markdown but display as html. Use of HTML tags still work as well. By default it's commented out but by editing the code using `editor.php` it is possible to turn math on using the MathJax JavaScript library, making it the same LaTeX-like markdown that is used in markdown elements in Jupyter notebooks. This allows for rapid free self replicating math papers to be created and shared on the Network.

Code structure

Showdown.js, scrolls/*, filesaver.php, fileloader.php, MathJax.js, dir.php, deletefile.php,

LaTeX workflow

address stability issues for large documents, alternative editors

mathuser.php

To convert a scroll to a tex document, copy the scroll into a new directory at the *nix command line. Then create a header and footer text file as follows:

header.txt =

and

footer.txt =

in the new project directory using your favorite text editor. Now be sure you have Pandoc installed, as well as pdflatex, and any stuff that needs to be installed for latex to work.

Now convert the scroll to a .tex file using pandoc as follows:

Then concatenate with header and footer using

And finally compile from text to pdf using

and if there are no errors in the tex code you will have a printable pdf document.

add full work flow to create a book with multiple chapters, this book. Articles. Links to more information. More details on installation and use of latex, workflow with latex editors to finish the project.

Chapter 3

Maps

3.1 Maps

Maps are a format in Trash Robot/Geometron which are a generalized meme. They represent an ordered list of objects, each of which has a position in a rectangular area on the screen. Each element in the ordered array has an x and y position and width all normalized to the size of the square area, as well as an angle in degrees. The other properties each element has are a url for an image if they're an image, HTML text for both if they are not an image and for alt text if they are, and a link destination which can be either a url or a map or scroll link inside the geometron system. Maps can link to scrolls as well as other maps. Also, each element has a Boolean variable “maplinkmode” which is false if it is just a normal HTML link and true if it is a map or scroll link. Maps are all stored in the “`maps/`” sub-directory of each Trash Robot/Geometron instance. They are in JSON format.

Scrolls are all stored in the `scrolls/` directory. Links inside the Geometron system are identified as to whether they are scolls or maps by the full name of the file. For instance one would link to this scroll from anywhere in the system using the name “scrolls/maps” as the destination of either a link in a map element which has maplinkmode set to “true” or in a hyperlink in the markdown format of the Scroll.

Maps are defined with the JavaScript library “mapfactory.js” which is in the “jscode” directory at `jscode/mapfactory.js`.

Maps are created in Javascript by for example in a DIV element called “mainmap” with following code:

Maps are edited using the program `mapeditor.html`. Click on all the things at random to figure out how to use that program. Save often. Copy/paste JSON code from the text area to share maps across the Internet or privately with other users. You can email JSON code, store it, copy it etc, and anyone can import it with a paste into their Geometron instance and save it locally on their server. This generalized meme format replaces both meme making software and PowerPoint as well as a large number of HTML frameworks and formats. It allows for a generalized system for encoding information on an image, which can be critical to documenting self-replicating physical technology. The three pillars of all Geometron/Trash Robot software are the Map, the Scroll, and the Symbols which are created with the Geometron language. This "symbol" is generalized to include those made in all physical media, so that includes things like lab-on-chip fluidic circuits, hybrid upcycled electronic circuits, laser cut shapes etc. Once Geometron is used to encode all human language and all symbols and also all technology, it can drive the hardware which displays maps and scrolls. When all of this lives on fully upcycled hardware, the system if fully metabolized and we can build self-replicating technology that does not have any mining, money, or property, the ultimate goal of Trash Magic.

Deletion

Maps are deleted with `mapdelete.html`. Just click "delete" to delete. Be careful, there is no backup. Also on public servers this might break, as do all file creation and editing functions from time to time. It will work instantly on a Raspbery Pi Terminal.

Replication

When you create a new map, run `dnagenerator.php`, and the next time the whole tree is replicated that map will come along for the ride. To replicate a specific map, find the URL of that map and use copy.php. The syntax is

The "from" url can be anywhere on the Open Web or anywhere visible on the local network. For example, pastebin.com or a raw code link on Github

Map editor Icon Meanings

Chapter 4

Feeds

home

4.1 Feeds

A Feed is a sequence of elements. The elements don't have geometric structure like a Map. They can be text, links, symbols, or any other kind of media. They are generally stored in the "data" directory as JSON format files which end with ".txt" so that they can be read by humans in a browser.

The Feed is a general framework for building formats, but in the basic Trash Robot server we implement a few versions.

Global Image Feed

This is an array of image urls. This is a key component of how Icon Tokens are made. We often start by doing an image search on the Web for some symbol, logo, image, or icon. We then right click the image and "copy image location" to the clipboard. Then we drop the url in the input in the global image feed to add it to the feed. Click the red "x" to delete the image. Image feeds can be exported from the text area, copied, and pasted into the same window of any other Trash Robot, imported and used anywhere on the Network. Since this data is just text it can be sent via text message or email so that feeds can be privately shared. The local image feed is stored at `data/imagefeed.txt`

We can make global image links in this Feed by uploading images to www.imgur.com, then right clicking the image to get the url and putting that url in the image feed. This method is used to document much of the Trash Robot system or for general rapid information sharing.

Link Feed

This is a feed of "links" in a general sense which can be images, links, or just text. They are edited using the "operator screen", which should be in the link feed itself, and can be found at `linkfeededitor.html`. Each element has three fields: “href”, “src”, and “text”, which are the url the link points to, the image if there is one, and the text. The data are stored on each Trash Robot at `data/linkfeed.txt`. As with the image feed, the whole feed can be copied, pasted, imported and exported using a text area, but in this case it is on the editor screen not the feed display. The input is used to put in urls of other link feed files. These can be anywhere on the Web. This can be used to make anonymous pastebin links which are link feeds which can display on any local Trash Robot without ever posting to a global server, for private exchange of link feeds. f ## Text Feed

The Text Feed is used for a number of Trash Robot applications. In spite of its name, it is not just a feed of text, but consists of three feeds(arrays): “text”, “src” and “href”. These really are what they sound like, three feeds in one. Users can add links, add images, add text, or delete any of them, and can copy and paste and share and import feeds. Text feed has a number of functions in the Trash Robot/Geometron system. It is used for the Map Editor as a source of links, images, and text which do not need to be entered in a keyboard. It is also used in the Poetry Engine and Duality. These are documented with the poetry engine scroll and duality scroll.

Chaos Feed

Chaos Feed is a user friendly text feed. Type in the input to post. Hit red “x” to delete. Nuke the feed with the explode emoji. Reload with the arrow loop emoji. HTML works, so you can manually enter html for links and images, allowing a link out to be added. Chaos Feed can be set to be the top level of a Trash Robot Server for text feed sharing mayhem and fun. Chaos feeds are stored at `data/chaosfeed.txt`.

Icon Feed

This is a critical feed for the overall system work flow, as it is how we share the Token Icons which are printed into clay. See the workflow map for links to the elements of the process by which these are made. Here again is where the copying, pasting, importing and exporting of feeds is very important. Users can create a whole feed of icons locally on a private server, then send that via private message to other users anywhere in the world, who can then edit on their own private servers, without any data ever leaking to the public Internet, while still having no users and no databases on each individual server.

Symbol Feed

This is not really a feed in the strict sense above, but it behaves like a feed in the user interface. Every time a symbol is saved using `symbol.html` an SVG and PNG file are both created, and these are saved in a directory called `symbolfeed/`. These can be saved locally and then used for anything. The pairs of files are also used when programming the Dremel laser cutter to directly create laser cut acrylic geometry shapes. The SVG files alone, with different layers as different colors are used for the cut and etch layers when making laser cut shapes ordered from Ponoko.com. Clicking on an SVG file also loads it up into `symbol.html`, including the structural JSON information which sets styles and positions of the symbol.

Wall

The Wall is a feed of one element. It is just a text document, stored at `data/wall.txt`, which is edited and read by users. Type to edit. Delete to delete. There are no users, no databases and no logins. Just information freely shared.