

GD399.01
OPEN SOURCERY

GD399.01
OPEN SOURCERY

GD399.01
OPEN SOURCERY

GD

OPEN

SOURCE

RY

GD



| | |
|---|----|
| Front Matter | |
| F/LOS Regularly / Kristian Bjørnard | 3 |
| The Open-Source Illustrator / Sasha Landar | 13 |
| Open Model / Cole McDonnell | 19 |
| An Open Community / Katie Hurley | 29 |
| Savate Serif / John Vetter | 33 |
| A Blender Handbook / Qingwen Zheng | 41 |
| Generative Moving / Gefang Zhang | 55 |
| Steps Mono Bold / Chris Corey | 65 |
| Agora: Making Salad / Devin Halladay & Calvin Hutcheon | 71 |
| Nodeboxing / Pragun Agarwal | 91 |
| A Running List / Mazzy Bell | 99 |

F/LOS Regularly

GD399.01 Special Topics in Graphic Design: Open Source ran from January to May 2018 at the Maryland Institute College of Art (MICA). We explored new tools and ideologies for making design using the lens of "Free/Libre/Open Source (F/LOS)." Each student was asked to use F/LOS tools in the making of their design projects all semester, and to experiment with how the ideas and technologies from this realm might serve their needs. While using some of the F/LOS software for final outcomes wasn't always possible (we worked on templates for our school library for example, they still needed things in InDesign), the conversations we had and ideas that surfaced from readings and classroom visitors made their way into student work both inside and outside the classroom.

Open Source Design

Why have developers latched onto the open source movement, but designers have not?

— GARTH BRAITHWAITE

A web search for "Open Source Design" returns several resources to start understanding how F/LOS has a graphic design impact. Near the top of the list is *The Open Source Design Manifesto* by Garth Braithwaite, a designer at Adobe. The manifesto is as follows:

I will:

- find opportunities to design in the open
- share my design experiences; both the good and the bad
- find time for meaningful projects
- openly participate in design discussions
- work with other designers by choice
- improve my toolbox

With *the Open Design Manifesto*, and a 2013 talk called *Designers Can Open Source*, Braithwaite explores simple actions/behavioral changes designers can take to become more "open." This revolves around an increase in sharing: sharing processes, especially the failures; and post as you are working, show how things evolve. This (hopefully) creates a new kind of ecosystem where designers are friendlier and more collaborative with their neighbors — more un-self-concious.

In response to Braithwaite's ideas, we decided to post all of our work on github and to record all of our discussions. This felt like a good place to begin the term. We had decided to be open, to share more, to start to utilize at least one tool of open source developers (Github), but where are the roots of this way of working?

Linus's law

The way we think about and talk about open source today owes a lot to Eric Raymond's *The Cathedral and the Bazaar*. In the essay, Raymond analyzes how Linus Torvalds (and the distributed hacker crew Torvalds helped lead) developed the Linux Kernel. Raymond outlines and explains nineteen points key to Linux's and GD399.01: Open Sourcery

Linus's successes. There are two in particular that merge well with Braithwaite's desires.

As soon as a new version of Linux was working, it would be posted for others. This "release early, release often" maxim flew in the face of existing software development practice: one typically did not want to release buggy code, it was feared that users would abandon your project, so one waited until every error was found before releasing. Raymond found that this was not the case with Linux. [I]t was clear that something different and much healthier was going on [with Linux]. The success was in how Torvalds treats users — they are treated as colleagues and co-developers. Release early and often is a predicate for Raymond's next takeaway from his analysis: "Linus's Law." Linus's law states that "Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone," or "Given enough eyeballs, all bugs are shallow." Get as many users as you can to tinker with your design, and make sure they tell you about everything right and wrong that they find — especially the wrong.

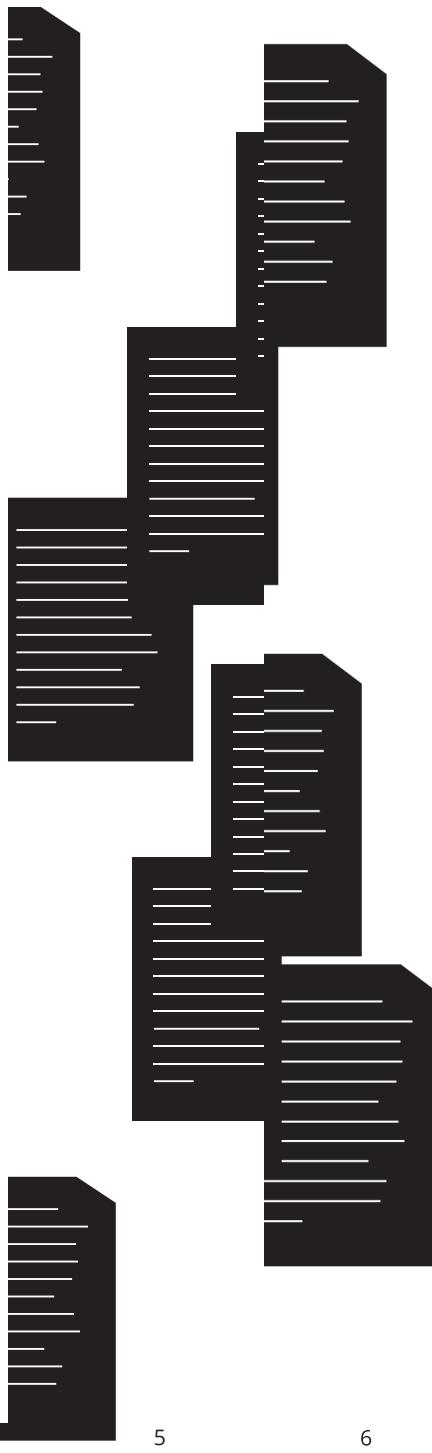
This is what Braithwaite is trying to get designers to understand that developers seem to have latched onto: many eyes make light work. In graphic design we are trying to find the best visual solutions — but how often do we throw out partially formed ideas and see if anyone can poke holes in our thinking early on in the process? This can be done within a classroom or studio between colleagues — as things come together make sure you're passing things around between desks/desktops; or, it can take place out in the world at large (via something like Dribbble? Github? Behance?).

More Than A Theory

It is absurdly arrogant to begin the design process with an empty piece of paper

— HELLA JONGERIUS & LOUISE SCHOUWENBERG

What Raymond is describing in *The Cathedral and the Bazaar* is not anything novel. There are other structures enabling cultural creation via distributed means and



including bug fixes from the community: vernacular design. What is described by Raymond is not so different from the way that human creative endeavors have come into existence for most of our collective past. In fact "the 'open source' way is closer to how human creativity has always worked" — it's the proprietary model of production that is new.

In *How Buildings Learn*, Stewart Brand dissects vernacular building and describes a process that predates the Linux community while sounding uncannily similar. In Brand's view, the vernacular process is a systematic framework for evolving concepts. A culture steadily culminates frameworks over time. Vernacular designers reuse invented common forms and methods for common, everyday tasks. Vernacular in this context means common designs by common people. Common features survive the passage of time when they are collectively understood as "good." Over the years then, a vernacular incorporates more and more "good" features while eradicating "bad" ones. Contemporary design typically looks for new or unique solution to a given problem — this is counter-productive to creating traditions and building tested solutions over time, and it cannot easily take advantage of this methodology.

To illustrate this in *How Buildings Learn*, Brand references the Cape Cod house. His included diagram is named "The Evolution of the Siasconset Whale House." This chart visualizes what Raymond is trying to explain to us in *The Cathedral and the Bazaar* (Braithwaite also wants contemporary designers to understand this), that simple things released into the wild of a user community will grow, evolve, and change — and will evolve improvements. The improvements merge back into the computer code or architectural language. Repeat. Torvalds is a genius to Raymond not because he uncovered something new, but in that he managed to take an existing form of cultural production and accelerate and decentralize it so as to maximize merging fixes and improvements and features!

If new software can be designed and constructed this way, and vernacular cultural objects have always existed this way, what's so hard in contemporary visual design?

Free as in Freedom

I consider that the Golden Rule requires that if I like a program I must share it with other people who like it.

— STALLMAN IN GNU MANIFESTO

Richard Stallman first published the GNU Manifesto in 1985. In it, he described “Free Software,” which was in opposition to proprietary software. Stallman’s motivations for this were both pragmatic and utopian. He hoped to maintain the kind of environment expected from academic computer programmers: a vernacular-like model where everyone builds upon everyone else’s work. Stallman also was hoping to fight the tide of proprietary, softwares and tools he felt rising. Free software is predicated on the following freedoms:

A program is free software if the program's users have the four essential freedoms:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

A program is free software if it gives users adequately all of these freedoms. Otherwise, it is nonfree. While we can distinguish various nonfree distribution schemes in terms of how far they fall short of being free, we consider them all equally unethical.

Free software, as defined by Stallman, is ethical. It has a point of view. Its clear about where it stands. Braithwaite and Raymond do not deal with this in their discussion or analysis of “Open Design” and “Open Source” respectively. Students that were attempting to get a grip on their

position as a designer intentionally using design as a political or critical act found something of use here. The software one chooses to use *can* reflect ethics.

In Opposition

The way Stallman positioned GNU was intentionally oppositional to “main stream” software. As GNU/Linux and other F/LOS tools have grown, they’ve maintained an outside, counter-cultural status. Stallman was never against developers making money — the GNU project and Free Software Foundation have always been clear that Free means Freedom, not free meaning price — but he is against developers or companies making money by sacrificing or minimizing the freedoms of their users.

Software/OS as a radical act is a fresh take for design perhaps? Anthony Dunne and Fiona Raby in *Designer as Author* propose that designers develop a parallel design activity that questions and challenges industrial agendas. F/LOS does this.

Walking into the average design classroom or studio presents one is greeted by an aluminum wall punctuated by glowing fruit behind which everyone has the same “creative cloud” raining down similar outputs from (mostly) the same sets of aesthetic inputs. By replacing one’s computer and software with F/LOSS alternatives a designer immediately engages in Stallman’s and Dunne & Raby’s critical activities. Designer’s can better access the transparency Braithwaite talks about and better actualize the open source/vernacular un-self-conscious iterating, improving, and forking; as well as provide a critique of our current situation by embodying alternative social and technical values.

Free Does Not Mean Easy

We have on the one hand a pragmatic approach based in the history of designing — how we as humans created cultural and societal artifacts; and on the other, a critical design approach that intentionally positions itself as oppositional from mainstream moderno-capitalism. By choosing openness, transparency, and sharing designers

are working in a way that is at odds w/ what the market wants. If you see design as serving business, then making your designs available to everyone might not seem like the right approach (Stallman has a rebuttal to this kind of thinking, but its a barrier).

While one might adopt more open practices for themselves and be more open to collaboration, one is less able to collaborate as the tools have changed — these are not the tools of most peers. In our class we collaborated with MICA's library on some new materials. The library thinks of itself as an open entity within the school (the ideals of Stallman's and F/LOS at large are mirrored by the Library's director and staff — we even did a wikipedia editing demo with a library staff member). However, there were limitations to what we could do; we still had to create everything for them with Adobe Creative Cloud. Our content and ideals were inspired by F/LOS — but our tools were not able to be. We were able to collect and share all manner of our designs, but we could not include the typefaces used in our repositories as they aren't open (the Library adheres to MICA's brand guidelines for its materials, so choices like Theinhardt for type were pre-determined).

That F/LOS tools are hard to install and/or hard to use is cited as a barrier to use. Every student came in frustrated at least once with not being able to do something in Inkscape or Blender, or was unable to get a feature of Scribus to work, etc. Is it actually harder or more complicated or "worse" though really? or have we just given so much of our time to other things already that it just feels like "bad" software because of the inconvenience?

Forks in the Road

Or, Random final thoughts/questions...

We need utopian thinking because without it, we are constrained by the tyranny of the possible.

— STEPHEN DUNCOMBE

What do you see out there when you look for open source desi

gn? Fonts, Icons, Templates, Frameworks. Is this what F/LOS design means? Is that all that's possible? We should start to see chances for design as a social critique; design that can return to its vernacular roots and not just serve clients or business. F/LOS ideals create an oppositional model useful for critiquing our mainstream culture. By choosing to use this surreal set of softwares (Linux, Scribus, etc.) a designer is specifically saying "I am not using the same computer or software as you." and "my computer software has different ethics than yours" This has use as a critique of the status quo.

As an educator my role is to help create new knowledge for my discipline. F/LOS allows for this in pragmatic and utopian ways. Its pragmatic in that there are F/LOS tools that do not exist in main stream operating systems nor from Adobe or Autodesk (the Spiro tool in InkScape for example, or NodeBox or Processing). It is utopian in that the tools themselves can reflect non-traditional values and propose alternate fictions or expand what is understood as possible or "standard."

Another pedagogical advantage to F/LOS is that the file types are open standards. This means files can easily be tracked and shared with tools that make a workflow of collaborating, sharing, etc. easy. It also relinquishes one from the tyranny of tools. One can focus on designing over application learning.

As a course, Special Topics in Graphic Design: Open Source, what was most useful was being reminded that Open Source software's successes are not unique to contemporary startup culture. The objects we actually need as humans have been being made in these ways forever. This means that we as visual designers are not precluded from accessing the techniques and ideas.

The end?

Kristian Bjørnard
2018-05-04, Baltimore, MD

Colophon

All type used in this section (well, and the whole book!) is F/LOS. The main text blocks use Noto Serif. The titles are Savate Serif by John Vetter (forked originally from Velvetyne's Savate). Inline quotations are in Noto Mono. Block quotes following headlines are in Cooper Hewitt. Imagery was created in NodeBox. All the SVGs are in the repo containing this book, feel free to use them — they're all GPLv3 licensed — Do what you want with them, but whatever you use them in you must also let everyone else do what they want with that. Except the image on this page, it is from the Internet Archive's Flickr ... It is part of Flickr Commons, my favorite place to look for photos and illustrations that have no known copyright restrictions, or are actually just flat out in the public domain. For the physical books, we collected paper from our department recycling bins. Everything is french folded. If you're adventurous, cut open a page and see what you find inside.

On The Cover

The cover is a risograph print Mazzy Bell kindly printed for us. The illustrations were drawn in InkScape collaboratively on a Raspberry Pi.



The Open-Source Illustrator

With the popularity of adobe tools and ever present .ai file, outside the Open-source community, we value the program that is a paid one. There is also a massive stigma around free programs, they are condemned as unprofessional or at least, less professional than if one was to claim to be an expert in the Adobe Suite. But why does an open source tool that can match the functions of an expensive tool, fall short for the attention of Designers?

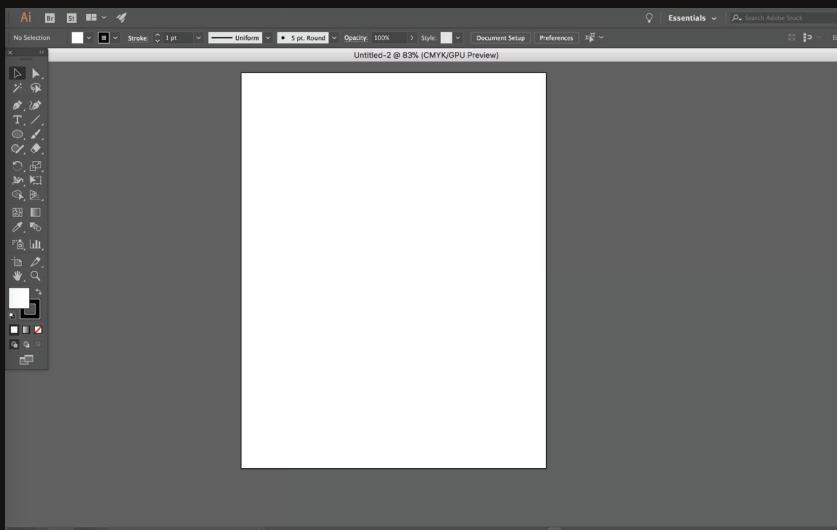
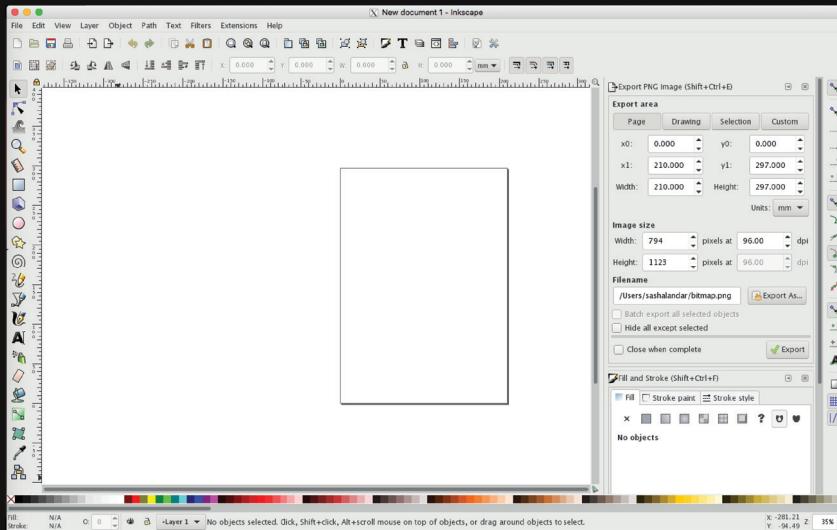
Personally, I dont understand why people training or working as Professional graphic designers do not learn to use some non-Adobe products. While many of us have familiarized ourselves with Open-Source softwares, most workplaces and schools do not utilize the free tools, or let alone teach them. Students and professionals alike would benefit if they become fluent in Inkscape, Gimp, Notepad, etc.

Not all professional work requires the use of Illustrator. For example, someone working freelance and is not required to share editable files could be just fine using Inkscape or many of the other SVG applications, as the result is the same.

While I do understand why agencies use a certain, single program to simplify and expedite the workflow among their in-house personnel. Illustrator being more feature rich would help in that decision as well. Also, there is the importance of using Illustrator when the client requests the source files from their projects. Although this just provides them with the ability to sidestep you when they decide to make changes to your work.

I guess I am just hostile to the suggestion that we have to use the same programs because the professionals say we should, especially when the final product will be delivered in .png,.svg,.pdf,.jpeg, etc. format anyhow. In reality, the request for source files is the only reason for why I have ever needed to work with Adobe products above other versions of raster or vector softwares.

There are obviously differences in keyboard shortcuts as well. Do not expect to command-z in Inkscape. Using the Control key in Inkscape vs the shift key in Illustrator to manipulate your shapes is one of many differences, but is not a disadvantage of any sort. It's all in the matter of how willing you are to learn new short cuts, or locate them on the tool dash, as Inkscape puts in the effort to lay all of the necessities out at hand.

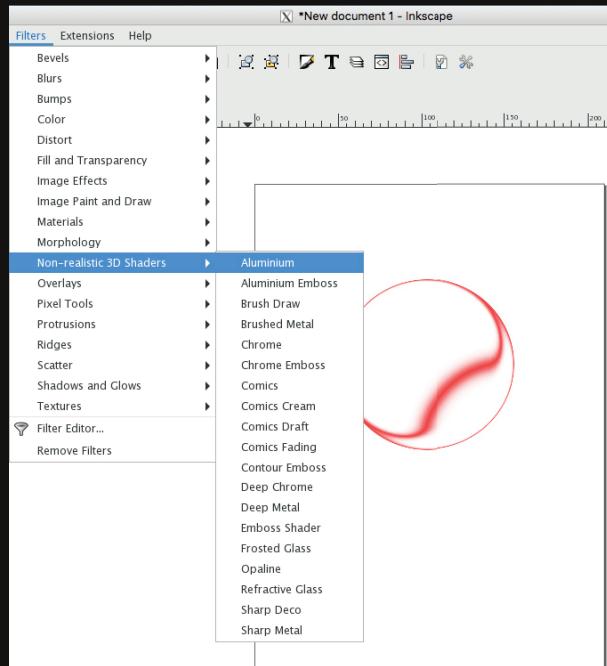


The interfaces are very similar, and while Inkscape may not look as polished as Illustrator, both interfaces are very user-friendly for both beginners and professionals. When using Illustrator, we expect a simplified vector based designing tool, but with Inkscape, there's a bit more discovering to do.

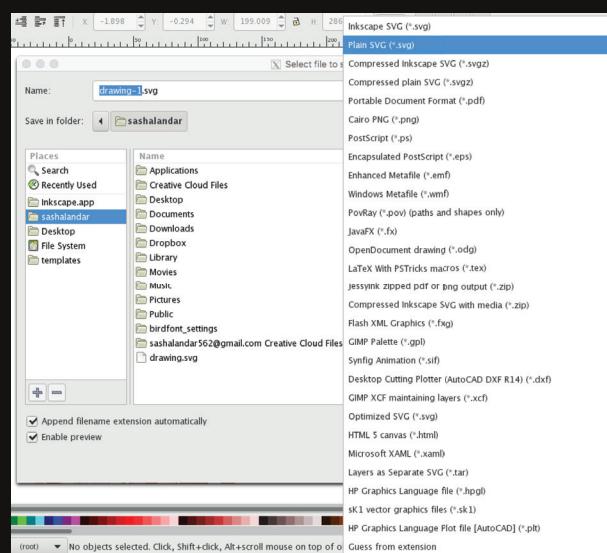
I think it is time we step away from the comforts of the Ai lifestyle and enjoy a more liberating software. Because Inkscape is Open-source, you can alter or improve the programming if there is something about the software that is not to your liking. It is always evolving, and things are always being added. Inkscape can be expected to surpass Illustrator, if it has not already.

While there certainly are hiccups in Inkscapes software, like not being able to export your files as .jpeg, again, this feature can be added with extensions.

Another pro of Inkscape is the massive library of textures and filters that make room for expression and experimentation. This wide range of effects certainly beats Illustrators.



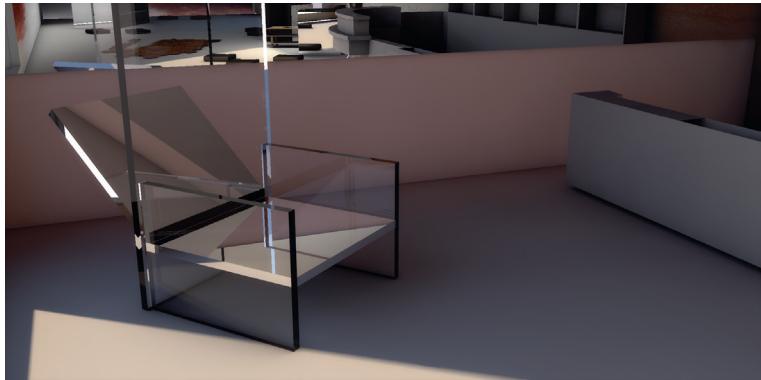
Maybe it is time to throw off the shackles of the evil capitalist oppressors and use an Open-Source software.



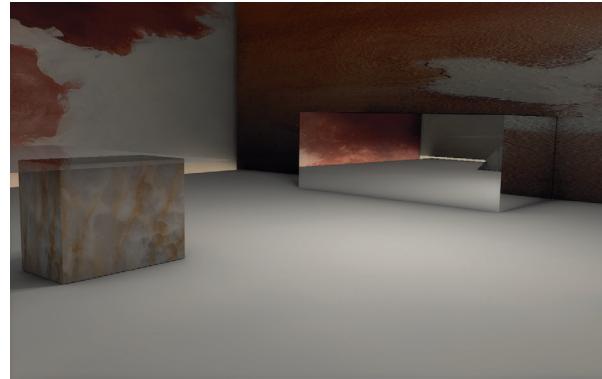
Open Model

Open Model served as an experiment with C4D, assuming the accessibility issues that come with the software.



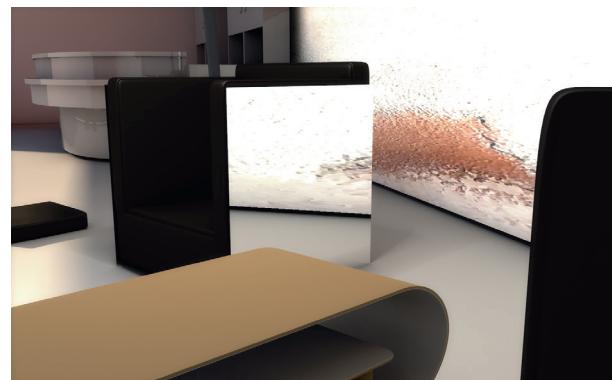


I designed multiple pieces of furniture inspired by mood imagery I've been gathering, and exported as .obj files so they can be used freely.



Because the .obj files don't retain the textures and imagery that I'm working with, it allows the user to only work with the initial form.





Open

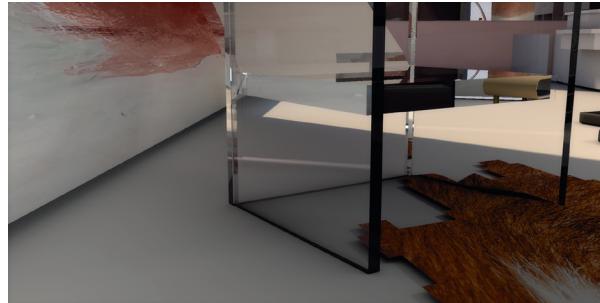
Model

25

26

GD399.01 : Open

Sorcery



The direct accessibility issue affiliated with Cinema 4D was directly apparent. Because a Maxon license is so pricy, only a select few of my peers are able to entertain the idea of opening the raw C4D file.

While I was asking other 3D artists already familiar with open source softwares like Blender, there was still a disconnect with the .obj files, as they retain the form, but not the fully rendered textures.



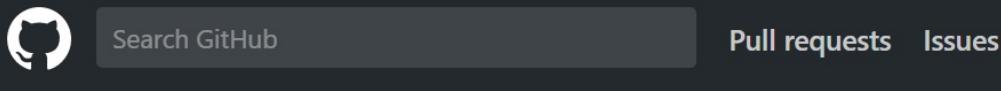
Open

Model 27

28 GD399.01 : Open

Sorcery

An Open Community



KHurleyDesign ▾

Browse activity Discover repositories

mikima pushed to [updated-libraries](#) in [densitydesign/raw](#) 23 hours ago
c5f7b34 Added C

mikima pushed to [updated-libraries](#) in [densitydesign/raw](#) a day ago
db52e31 Horizontal chart: update to d3v5

mikima pushed to [updated-libraries](#) in [densitydesign/raw](#) 3 days ago
ecbdf1f Bar Chart: clean up

mikima pushed to [updated-libraries](#) in [densitydesign/raw](#) 3 days ago
e6c11 Fixed paddi

mikima pushed to [updated-libraries](#) in [densitydesign/raw](#) 3 days ago
5b451fb Updated

andy5995 pushed to [master](#) in [mbwkb/mbwkb.github.io](#) 5 days ago
7393d2c Delete CNAME

andy5995 pushed to [master](#) in [mbwkb/mbwkb.github.io](#) 5 days ago

How does a GitHub community work? What does it mean to contribute to an open source project? How do things get accomplished? These are all questions that I explored when viewing and trying to engage in an open source community.

GitHub is a web-based version control and collaborative platform. Based on git, this platform functions as an open source code management system. It fosters connections between the creators of the projects and the user. Allowing for collaboration and advancement of projects. But what does it really mean to be apart of an open source community.

Most open source projects develop out of someone's specific need or goal. For example, you want a program that generate random quotes from your favorite book. You create this program and then publish it for others to use. But, users don't always have the same intentions when it comes to your project. So they find bugs in your project created from unexpected use. They can either fork the project and fix the issue themselves or they can submit the issue for the creator to fix.

Community are created around projects where the creator and the user are invested on up keeping and investing in the project. I had the opportunity to participate in one of these communities. I found out that within these communities there are generally only a select few people that actively participate in these projects along with the creator. But these people form a connections. And often they end up talking about more then just the project. For example, they will share music they like with each other.

Filters ▾ Labels Milestones New issue

① 8 Open ✓ 52 Closed Author Labels Projects Milestones Assignee Sort

| | |
|--|----|
| ⓘ [discussion] Language switcher pri: 3 (normal) question | 4 |
| ⓘ It may not be clear that clicking on a language is required design help wanted pri: 3 (normal) question | 10 |
| ⓘ tags have no language associated with them help wanted jekyll plugin pri: 3 (normal) | 10 |
| ⓘ Have browser default to the proper language functionality help wanted pri:5 (low) | 1 |
| ⓘ post will appear on tag page twice if a tag is listed twice jekyll plugin pri:5 (low) reference | 2 |
| ⓘ Implement weighted tags (a cloud) on a separate page css enhancement help wanted html jekyll pri: 3 (normal) | 12 |
| ⓘ sorting posts alphabetically enhancement help wanted jekyll pri:5 (low) | 4 |

adc jek #11

KHurleyDesign commented on Jan 26 Member + ↗ ×

What if the top navigation was a sticky header and followed the user down the page? That way its always accessible or there was some sort of prompt next to the language selection like "select one".

andy5995 commented on Jan 27 Owner + ↗ ×

That sounds like a good solution to me. I can't think of any downsides to it. @erichoog @escottalexander what do you think?

Thanks for the suggestion @KHurleyDesign

escottalexander commented on Jan 27 Member + ↗ ×

Are we saying that the language selection will follow a user down the page until they select it and then it will leave them alone? I like that idea because it will help a new user realize that there are other languages available.

KHurleyDesign commented on Jan 27 Member + ↗ ×

I was just suggesting it followed them period, but I think that your idea would be good as well. I am just not sure code wise how that would work.

erichoog commented on Jan 28 Member + ↗ ×

Is this what you mean @KHurleyDesign?

Github Issues are an essential part of the open source community. This is how the creator of the project engages with people using the project. It also allows the creator to discover bugs that he might not know about, as well as discuss and accept fixes from users. This allows the project to progress at a faster rate than if the creator was the only one using it.



You don't have to have extensive knowledge in code to get involved. You can contribute graphics, user input/feedback, edited text blurbs and much much more. Contributing in one source communities is a way to help projects progress and to find interesting and often amazing software outside of the norm of proprietary software's like Adobe.

Savate Serif

A type design experiment with John Vetter

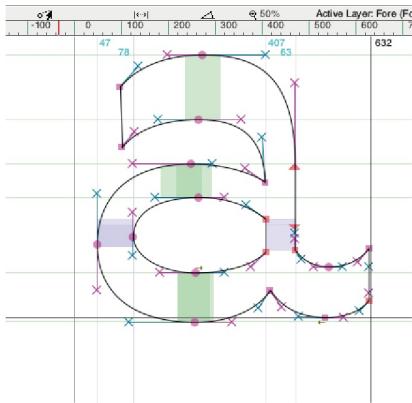
ABCDEFGHI
JKLMNOPQR
STUVWXYZ?



Before this class, I had no experience with the open-source community. I generally of it's tools, but had never used them before. The domain I was the most familiar with was the type scene. The open source community, especially in France, seemed to be the most experimental and really pushing boundaries in type. This evolved mostly after my discovery of The Velvetyne Foundry. Savate was one of the fonts on the site that immediately struck my interest, being well made, having a fairly full character set, and two styles. I chose to make a serif version, both to add to the community and to learn the tools (FontForge and Github). On the next page are some documentation of some things I learned.

Learning Fontforge

Most of the learning I did with FontForge was through opening font files in the program and seeing how I could manipulate them. It isn't the most intuitive since I learned with Robofont, but it was not nearly as clunky as I was expecting. The most helpful resource I found was www.designwithfontforge.com. A few of the main problems I encountered in FontForge were drawing counterforms, and re-naming font files.

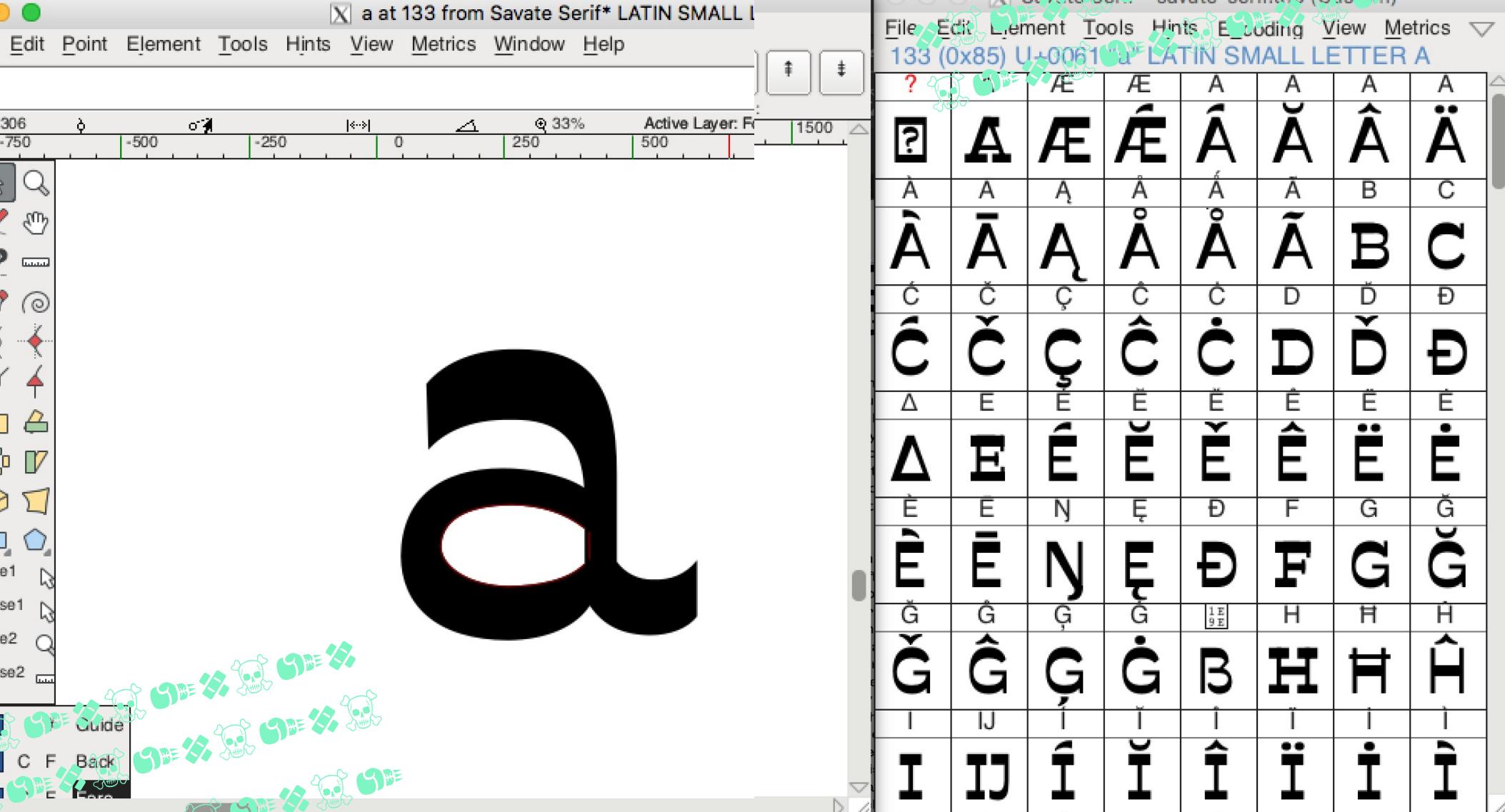


An early issue where the 'a' character looked fine as outlines but previewed as a filled shape.



I encountered my counterform issue drawing the lowercase 'a' character. I figured out how to remove, join, and add points easily. I created the serif for the character, then closed the counterspace again. However, when I closed the forms, they all remained positive space. I could not figure out how to change the interior bowl of the 'a' from the positive space to negative. My initial solution was to revert to the original character, and then re-draw it. This time, I added the amount of extra points I needed to the stem of the 'a' and pulled the serif out. This way, without breaking the counterform, I was able to create the serif. This was much clunkier in the drawing of the character, but it worked. I made most of the typeface with this method in place. Not proud of my sloppy design practice, but it happened.





Ac

File Edit Element View Metrics Window Help

PAROLES DE SC

New Lookup Subtable...

| name | F | A | R | O | E | S | space | C | O | U | T | exclam | | |
|--------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|-----|--------|-----|--|
| dt.. | 680 | 791 | 703 | 694 | 535 | 626 | 657 | 230 | 660 | 694 | 829 | 610 | 330 | |
| aring: | 50 | 30 | 50 | 50 | 50 | 50 | 50 | 0 | 50 | 50 | 40 | 30 | 82 | |
| aring: | 30 | 20 | 10 | 50 | 30 | 35 | 48.99 | 230 | 20 | 50 | 40 | 30 | 82 | |

Problems in The Source File

Inside the Savate source file, I found that the capital X was 5 points or more taller than the rest of the capitals. In the original Sans version, this issue wasn't so easy to spot. However, with blocky and tightly spaced characters, this problem was immediately apparent. This turned my experiment not only into an attempt to add to the Savate family, but an attempt to fix it.

A B C D E F G H I
J K L M N O P Q R
S T U V W X Y Z ?

Caps with the original 'x' height



Text by Lorain Furter

FREE AS IN
BUY ONE GET ONE

by John for Kristian

[**] is the perfect font for body text and **headlines on a website**. It's **modern style**, suited with past characteristics of great typefaces, make it **highly readable in any context**. The full-circle curves on many characters make [**] **a great font to blend seamlessly WITH OTHER FONTS WHILE STILL MAINTAINING IT'S UNIQUENESS**. Whether to be used for body text or **headlines on a web page**, [**] is the right font for any project.

Fonts used include: Sporting Grotesque, Young Serif, Trickster, Cormorant Garamond, Bagnar, Poppins, Work Sans, Cooper Hewitt, Savate, & Heebo

Here's a poster I made of some of my favorite open source fonts using text by Loraine Furter. Download it on github or something, I don't know. _('')/_/

Bringing The Humor to Life

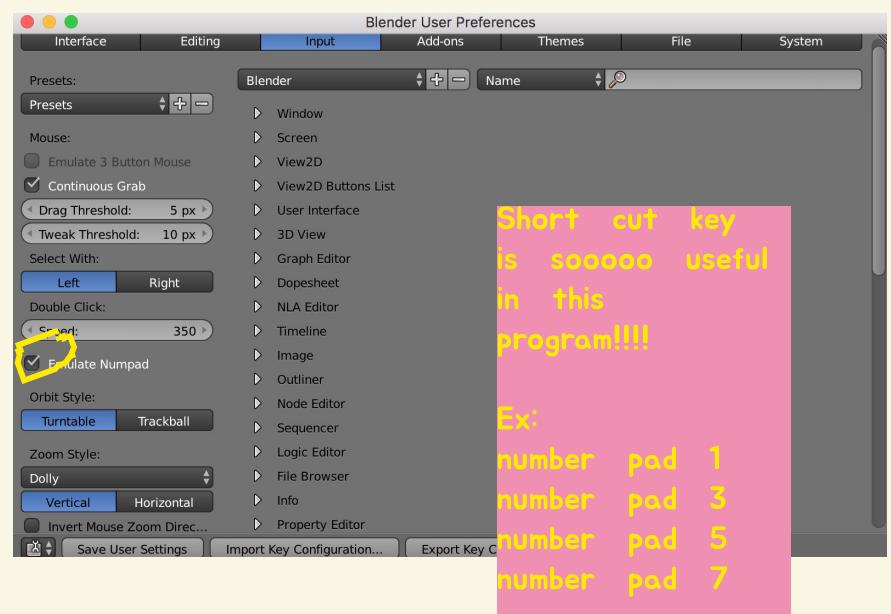
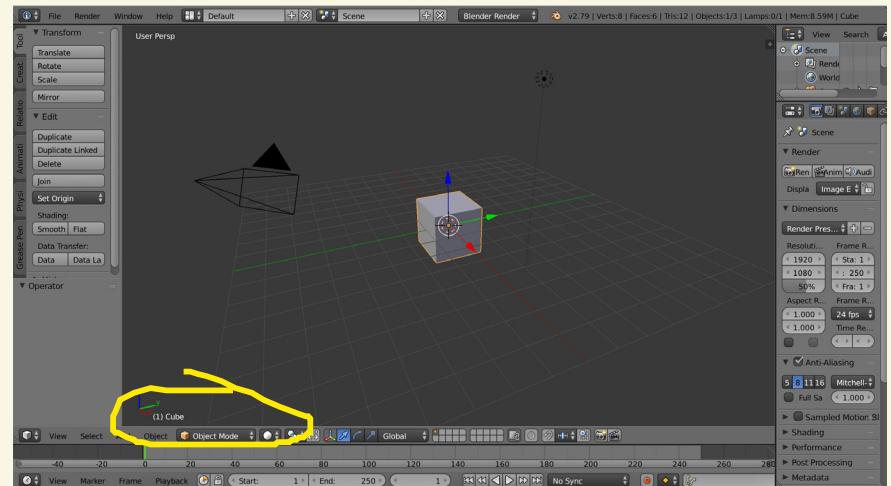
A BLENDER

HANDBOOK

1

Interface

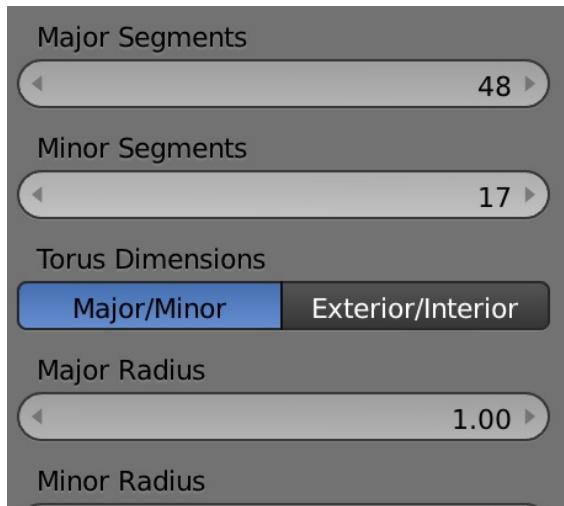
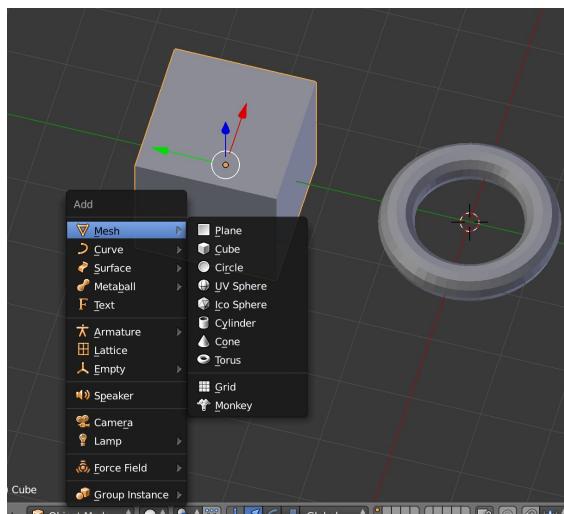
Clicking, setting preferences,
working panel, viewing object



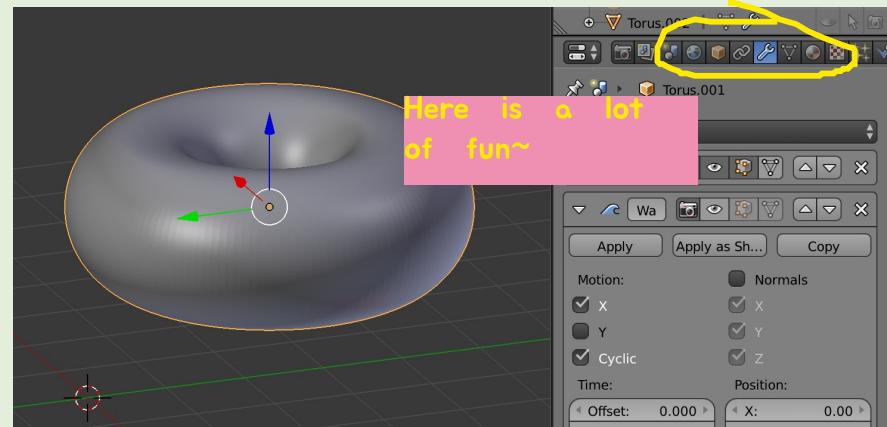
2

Objects

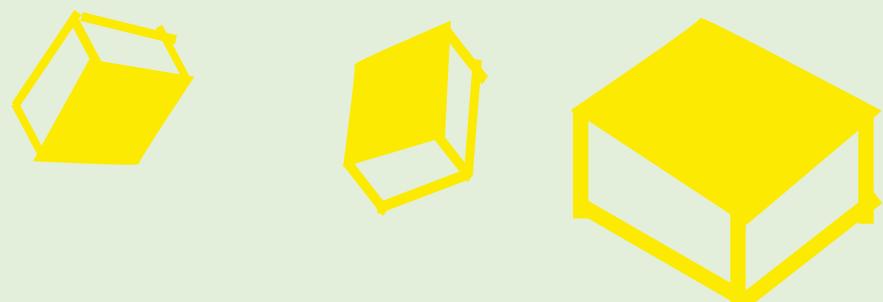
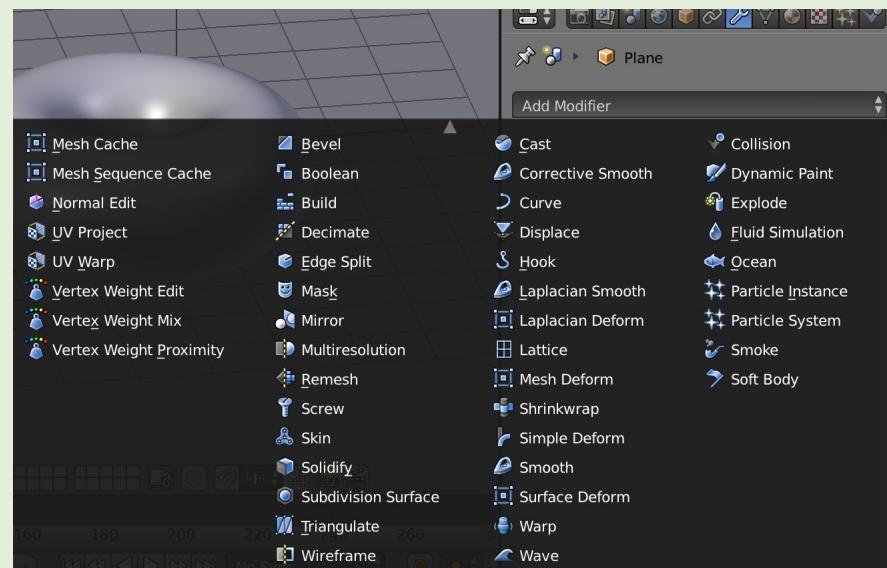
Placing object, moving, scaling, viewing using the cursors and selection cursors



It's annoying that you can only change your segment before you make any click.



As well as these:



3

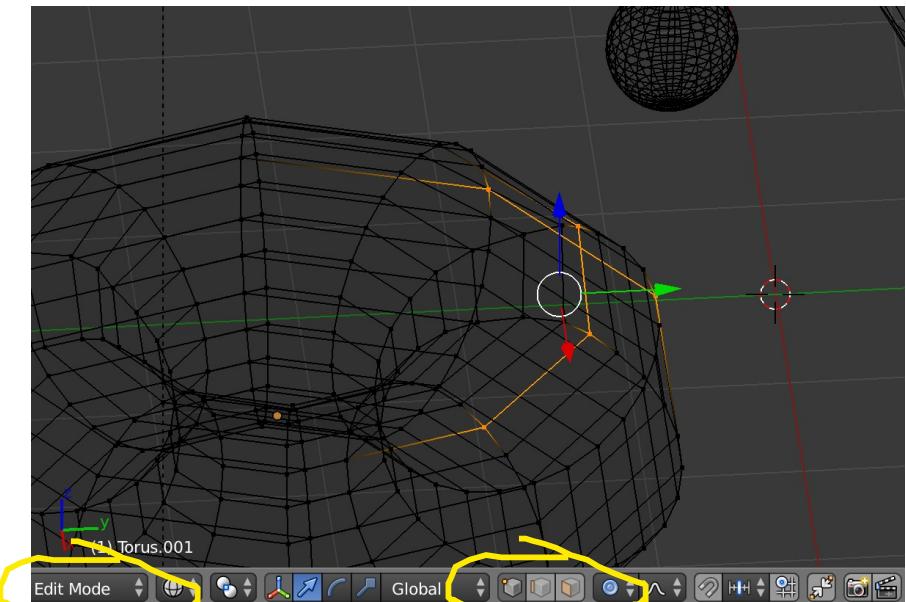
Editing Mode

segment

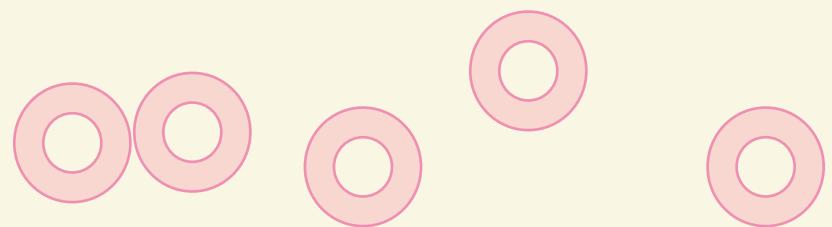
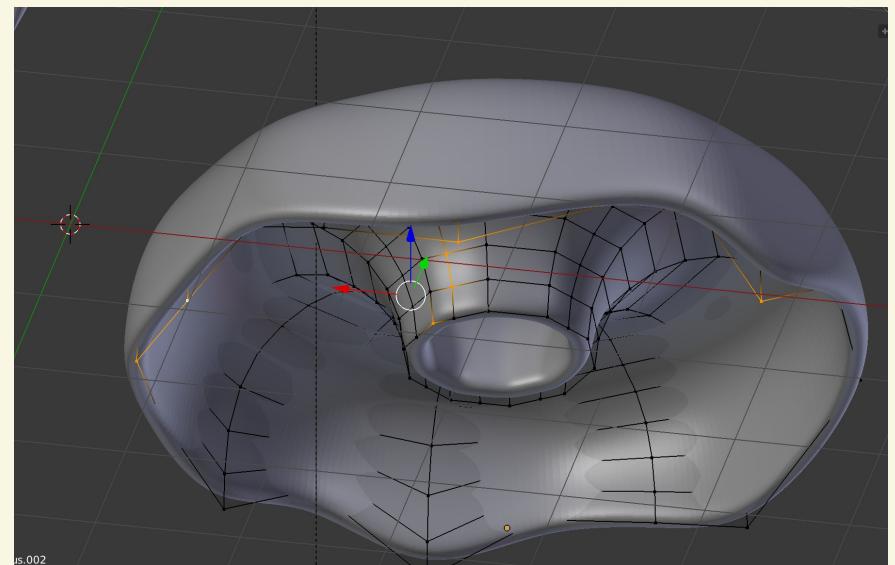
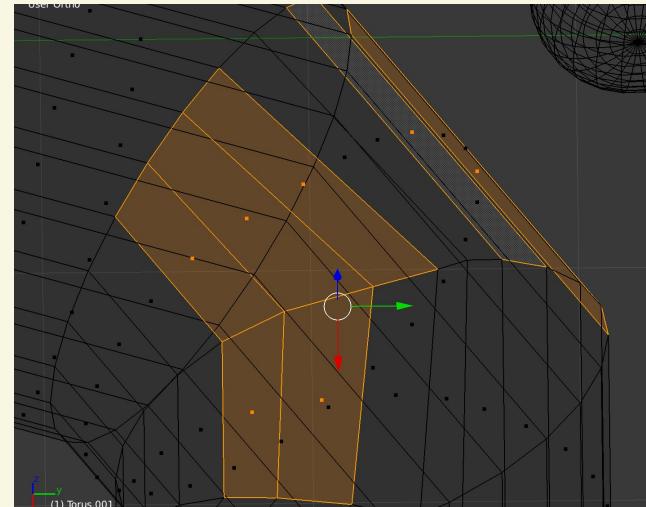
Point

Line

plane



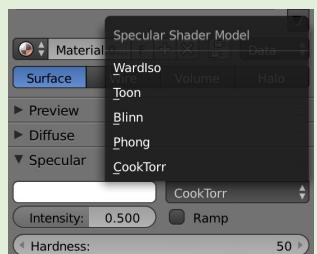
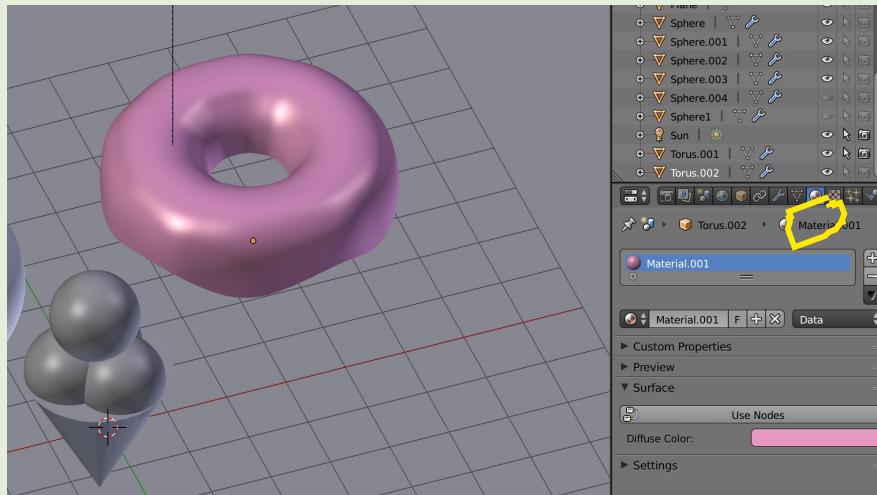
Scaling, moving,
reflecting can
also apply to
edit mode.



4

Material

Color, texture, bumps.

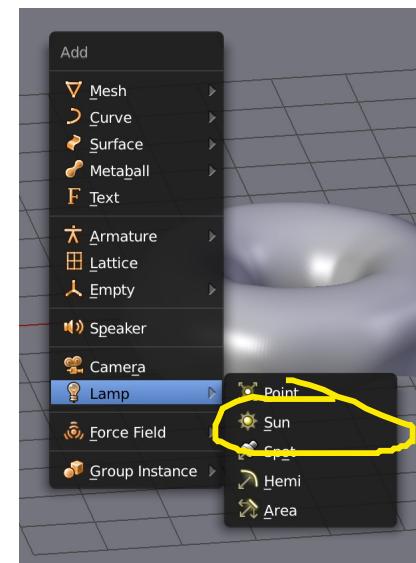
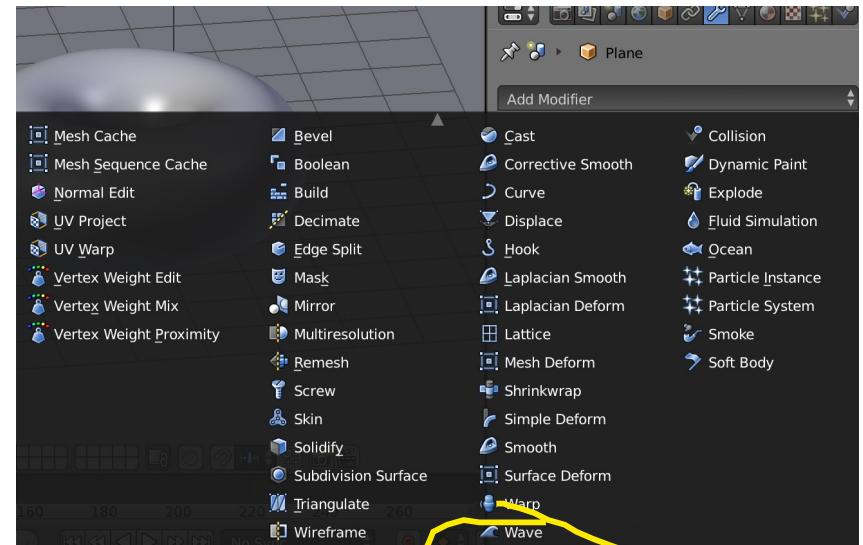


Specular is so
caooooool.

5

Animation

Waaaaaaaaave~~~~~
& Suuuuuuum lighting~~~~~

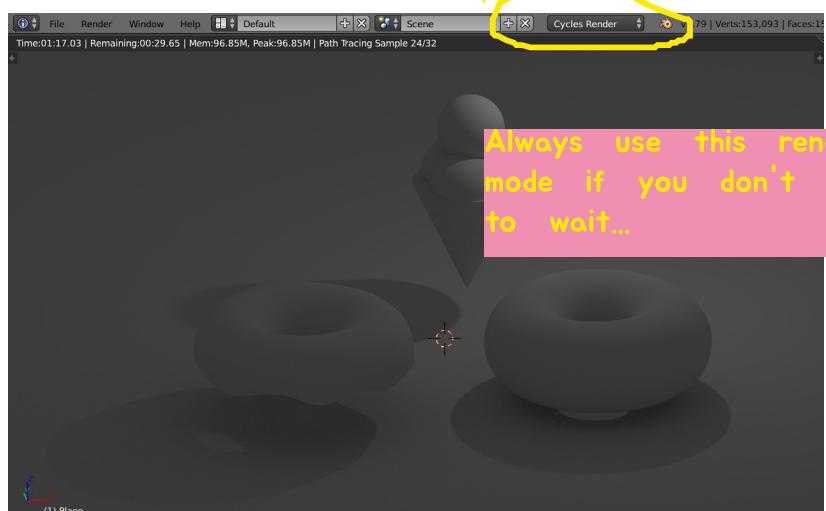
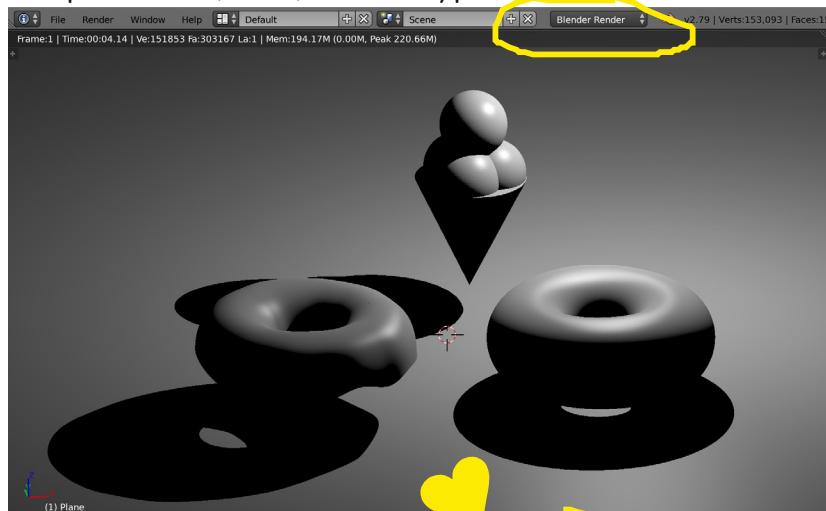


Yes, I only used these
two features to form
my beautiful final, haha~

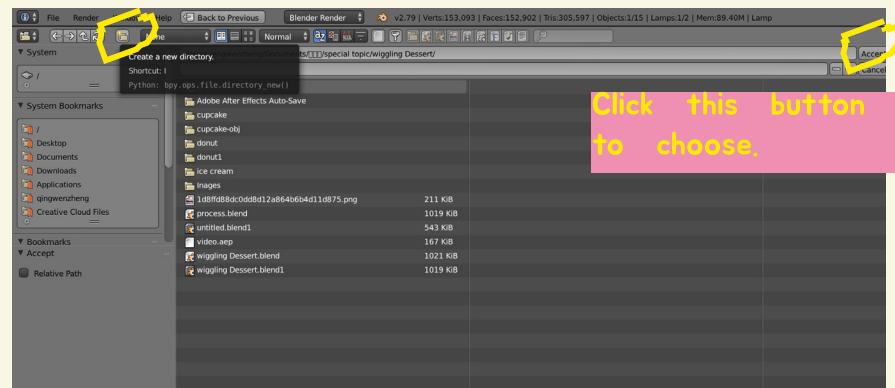
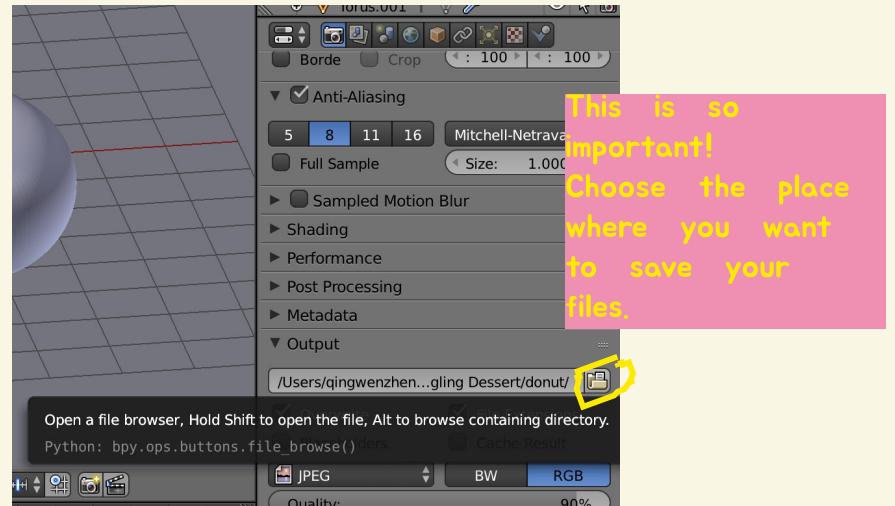
6

Render

Output folder, size, render type

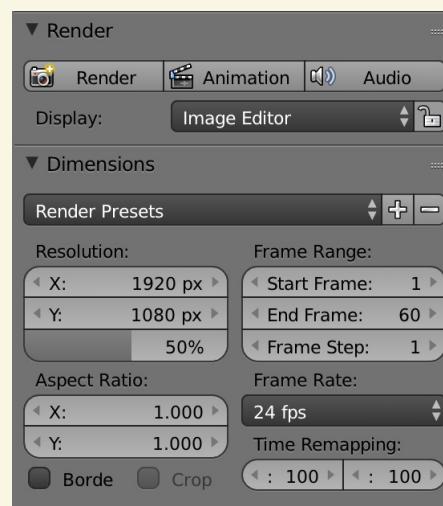


Always use this render mode if you don't mind to wait...



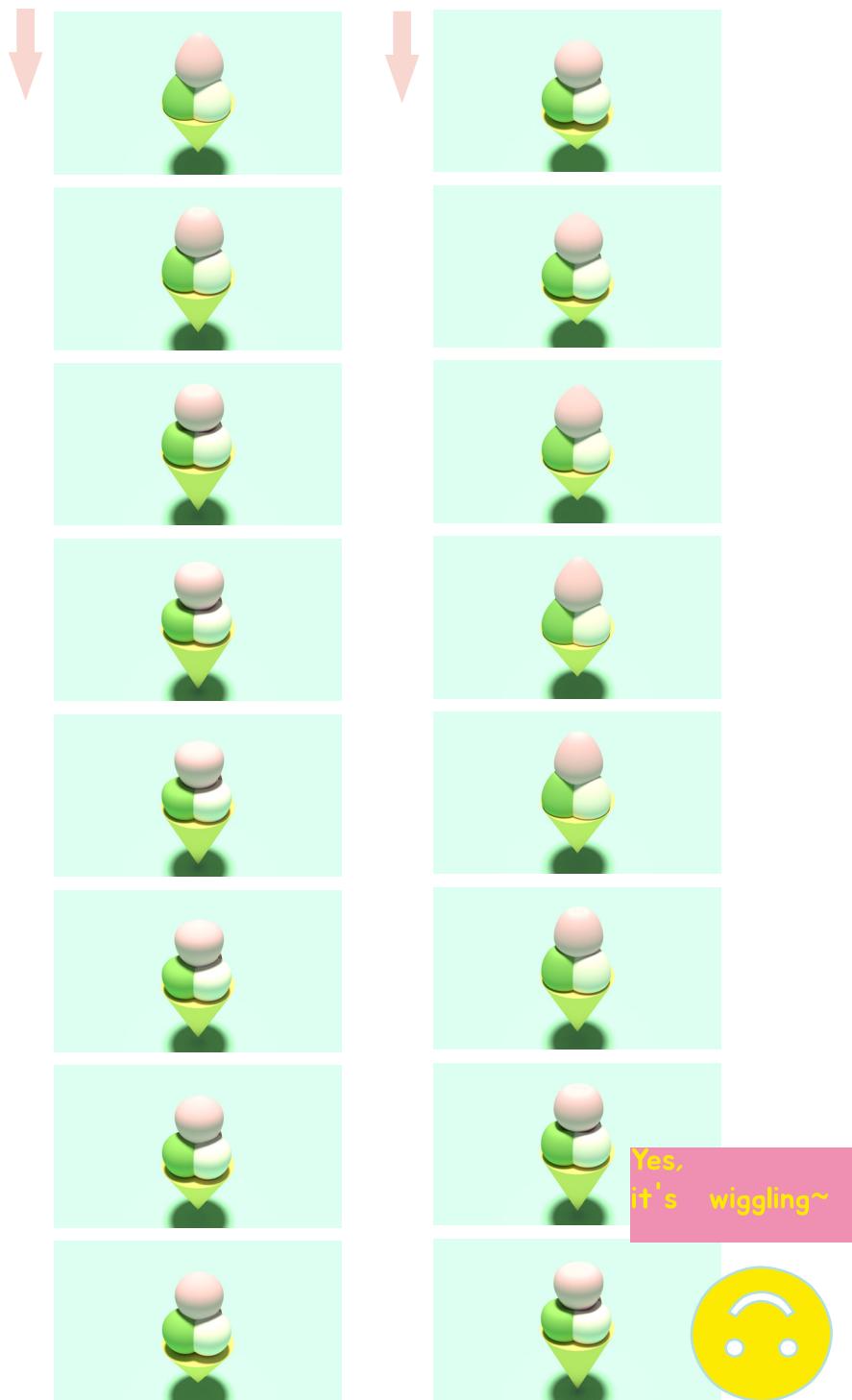
This is so important!
Choose the place where you want to save your files.

Click this button to choose.



Carefully select those items, they might let you render forever...

FINAL



Ching's Thought About Using Blender as Open Source Programs

The first 3D graphic software I started using is Cinema 4D, which is considered one of the most user-friendly 3D application. However, as a beginner with both Cinema and Blender, I actually felt that they each have different advantages according to their interfaces.

Learning Blender is not much different from learning any other kind of graphic editing tool. Except that there are even more features you could explore than in other things I have tried. Starting with watching Youtube tutorials, I got familiar with its way of viewing objects. Little by little, I started to be able to achieve my own creativity. I found it's not hard to make funny and awesome images even as a beginner. It's also very hard to believe such a powerful tool is free, while other 3D programs could cost thousands of dollar. I really appreciate this since people could express their imagination without worrying about whether they could afford the approach. Once a program is open to everyone, there is more inspiration between users.

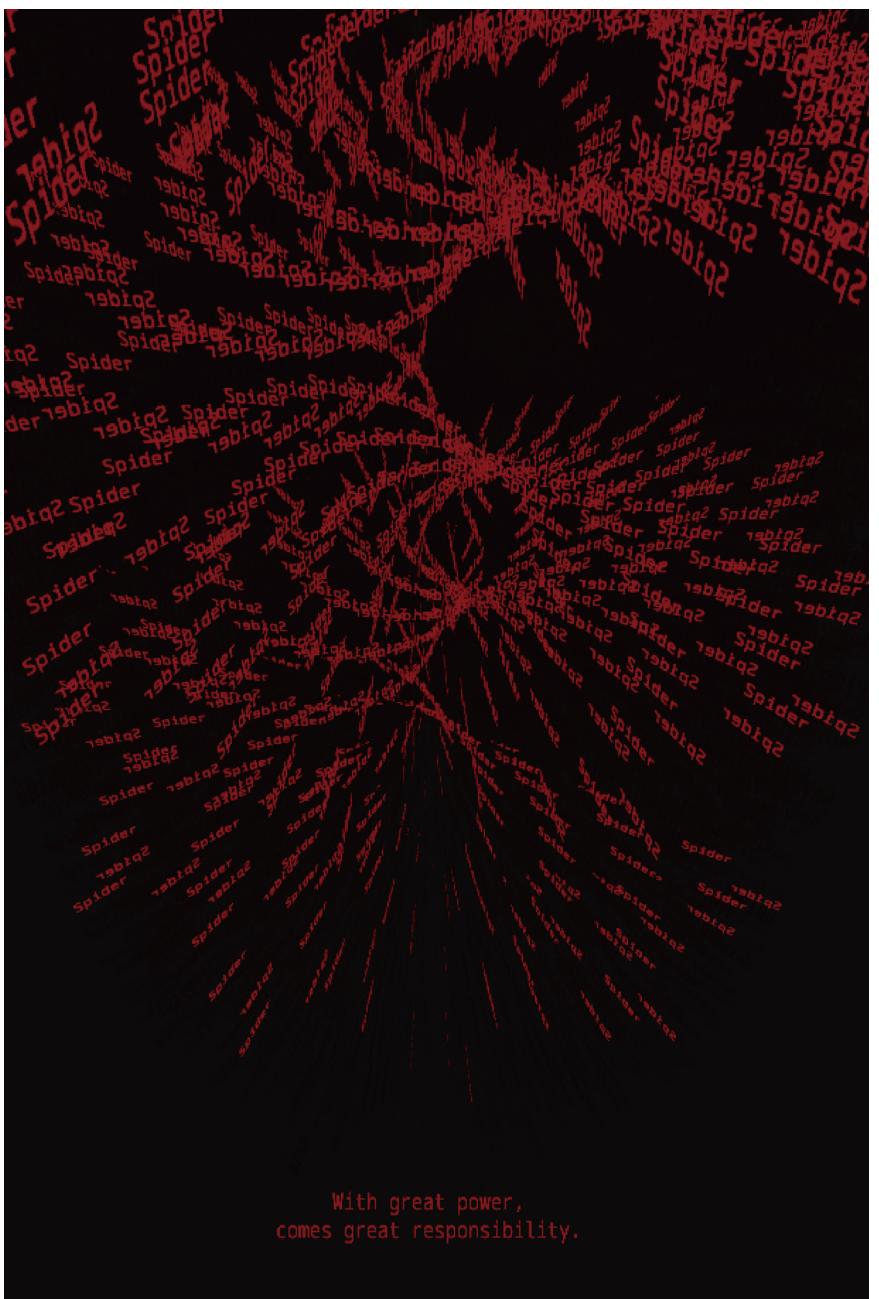
As an art student, I am very glad that I put time and effort to use Blender and learn to take and give. Try it, you'll find more then you thought.

Ching Zheng
April 27th 2018
At room 206 ~

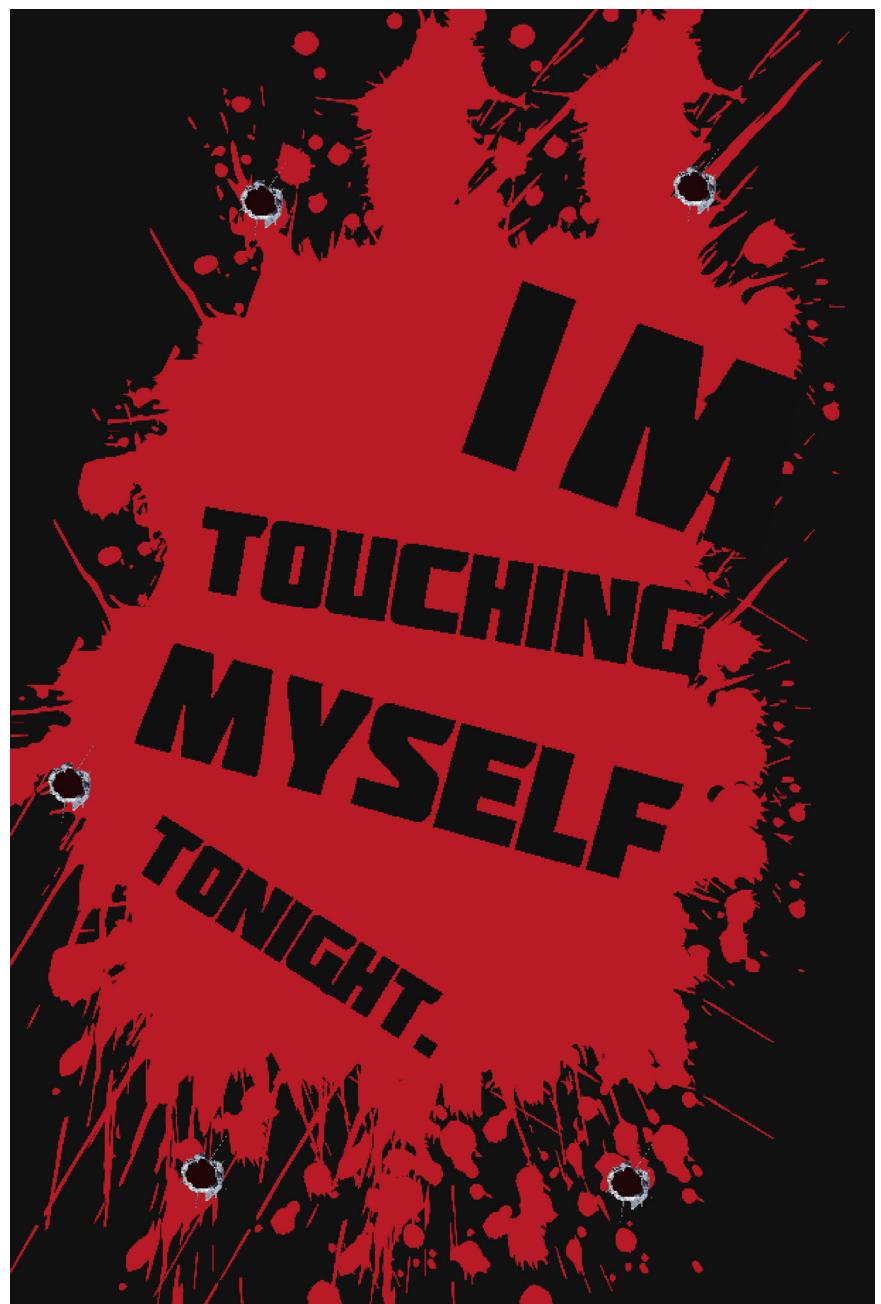
Generative Moving

Interactive Movie Poster

Spider-Man
The Matrix
Deadpool
Great Gatsby



Your mind makes it real.



You
cannot
repeat
the
past



F.Scott Fitzgerald

Great Gatsby

Statement

These are series of interactive movie poster that created with processing. I choose few of my favorite films and create an interactive type poster based on my own interpretation. I pay the most attention to how dynamic composition, animation and interactions can help storytelling. The use of slogan are help people to recognize which movie is that.

Spider-Man

The poster have a interactive double helix Spider, which represent the DNA change of Peter Parker. The slogan is with great power, comes great responsibility, which is the most famous quote from the movie.

The Matrix

The poster have a matrix rain drop animation which represent the core statement of the movie. Audience can interactive with their mouse, the slogan will be show after audience hit some points on the poster

| Deadpool

Poster are interactive with mouse. When audience click the mouse, the gun will shoot the poster and the blood from the gunhole will fill the background. The slogan will show after that.

| The Great Gatsby

You cannot change the past is the most famous quote from the movie. The quote will gradually fill with the bubble from the champagne animation.



STEPS MONO BOLD

DESIGNED BY CHRIS COREY

Steps Mono was designed by Raphael Bastide and Jean-Baptiste Morizot for the French magazine Etapes: to demonstrate a collaborative design on the same font. It is a libre / open source font released by the Velvetyne Type Foundry with two weights, regular and thin.

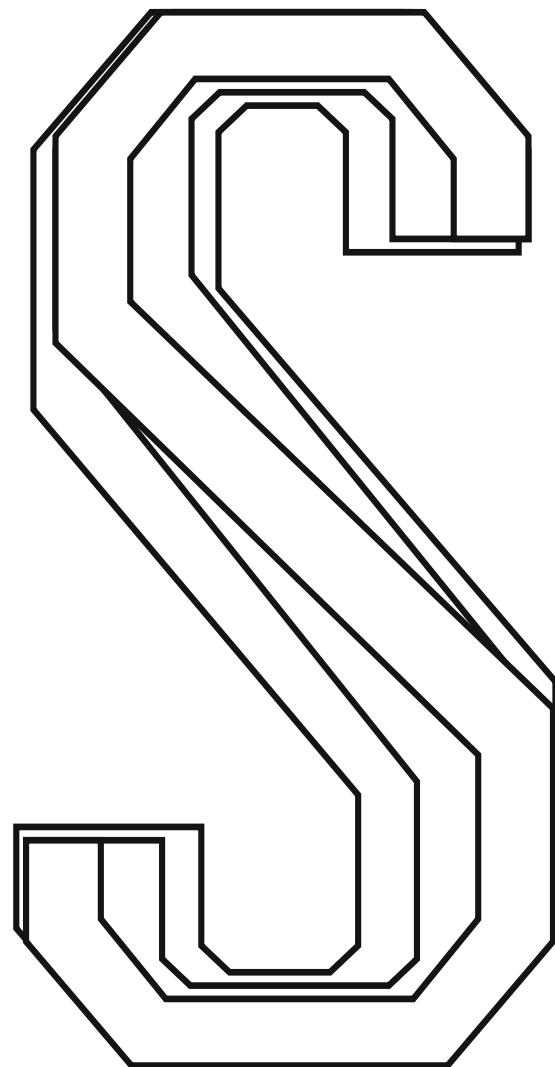
I've wanted to try a design a monospaced typeface for a while now and I thought designing Steps Mono Bold over the course of this semester would be a nice way to learn more about the style and gain experience before I start on my own project. It was a great way to explore a specific style of typography without a huge time commitment and I'll definitely play around more with libre fonts in the future. They're a great learning tool for new type designers like me because I don't have to start from scratch and I get to look at every glyph up close in FontForge.

A B C D E F G H I
J K L M N O P Q R
S T U V W X Y Z

a b c d e f g h i
j k l m n o p q r
s t u v w x y z

0 1 2 3 4 5 6 7 8 9

**HAND
CRAFTED
INFECTION
FREE BEER
THE INVASION OF
12 FLUID OUNCES
DESTROY YOUR PLANET ALTOGETHER**



THE FUNDAMENTAL FREEDOMS OF THE LIBRE FONT USER

1. Use them for every purpose.
2. Study and modify them.
3. Redistribute them.
4. Redistribute modified versions.

THE VELVETYNE FOUNDRY MANIFESTO

We, VTF, dauntless offspring of the Postscript generation, we chase at the speed of the electron, the course of typography and its attendant of wrestling writing warriors. Like Spartacus, we unchain glyphs from unfair laws and arbitrary conventions, like Ben Hur on a gigantic tank, we welcome script horsewomen and type charioteers and all tremendous, armed with bulletpoint pen and bold crossbows, wearing draw hats, we challenge here and now the tantalizing palimpsest of today.

Working with FontForge

I was impressed with FontForge considering it's free. I'm used to working in RoboFont and it was an easy transition. My main complaint was that I couldn't figure out how to test install my work for proffing which means every time I wanted to print it out I had to generate the typeface and install it on FontBook before using it. I still prefer RoboFont but FontForge did have a few features that I wish RF had. The main one being able to work on letters in a string of letters. In RF you can only work on a glyph in isolation which isn't ideal.

Working with Steps Mono

Steps Mono felt like the perfect monospaced typeface to start working with. It's a condensed face with provided a few challenges when making a bold weight, but at the same time, the typeface is all straight lines and angles, so I didn't have to deal with refining Bezier curves. This made it easy to really focus on getting the weight transition right while staying true to the original design.

Working with Scribus

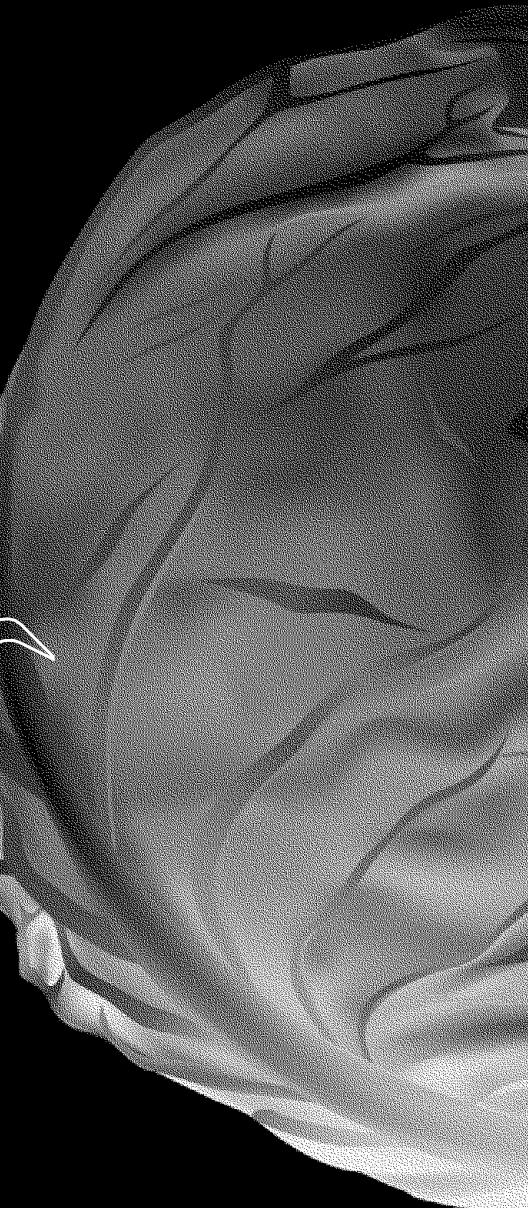
This was not a pleasant experience. I'm so used to working in InDesign and I just wanted things to work a certain way. I've only worked with Scribus for a few hours, so I haven't given it much of a chance, but I feel like it does pretty much everything worse than InDesign does. The UI isn't very refined, and the software feels a bit buggy. But most of my frustration probably stems from the fact that I spend so much time in InDesign and Scribus...isn't InDesign. It took me much longer to get things done with Scribus and it drove me a little crazy. I need to work on my patience I guess. Or just go back to InDesign.

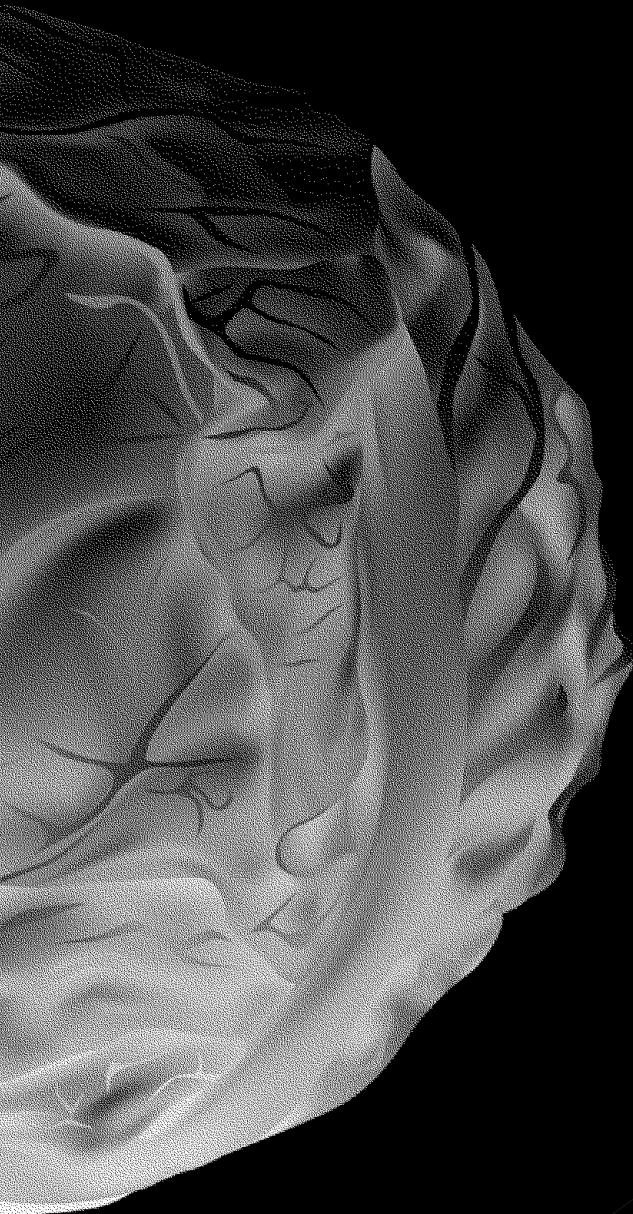
Agora

Making
Change

Agora

Making
Change



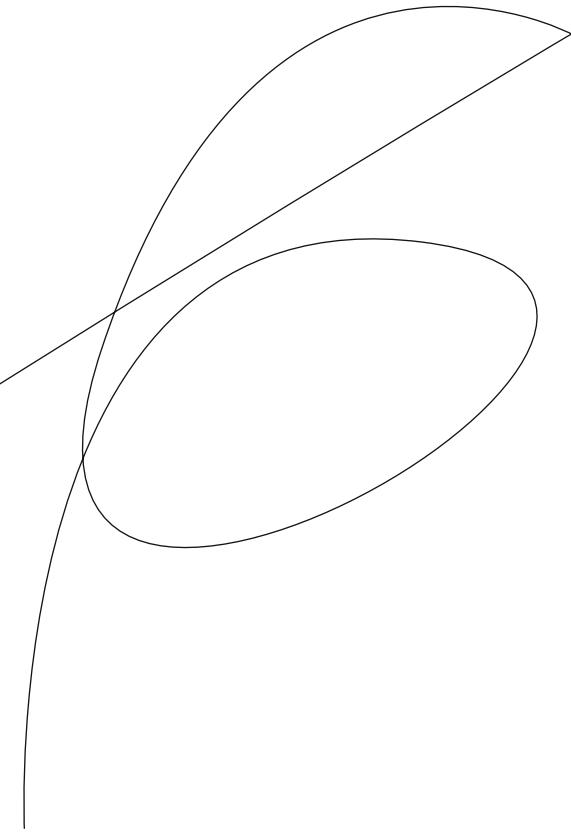


Calvin Hutcheon
Devin Halladay

hyperfruce

A Vision

radically public space



Brown 307 is an open studio. throughout the course of this semester, Devin Halladay and I have conducted a series of spatial interventions, attempting to open the room to agonistic discourse. Employing a variety of strategies, we used basic materials to demarcate and elicit conversation. Using pink tape, we opened space to students, asking people to post important thoughts, ideas and influences. Providing glass markers, we made tools available to brainstorming, messaging and communication. Emptying one shelf, we marked the level with pink tape declaring it a design Praxis library. Finally, we connected a raspberry pi a wall mounted tellivion, connecting the space to a public are.na channel (<https://www.are.na/open-for-discourse/mica-gd-open-discourse-1518547783>).

These strategies worked to varying degrees. Though participation was sometimes limited, in one instance discourse conducted within these spaces led to significant department wide conversations. Observing the space and the discourse conducted within it, I compiled a case study (available at: <https://github.com/millenniallglyphs/Open-for-discourse-casestudy>), coming to the conclusion that the intervention compelled only surface level interaction. People were sharing concerns, ideas and solutions, but there was no structure in place to encourage follow up. From my study I identified four key actions needed to elicit further organization. These actions are 1. identifying a good idea. 2. connecting people 3. connecting information 4. facilitating meet ups.

With these ideas in mind Devin and I began mocking up an app that would take the place of the are.na channel we had set up on the TV. Our intent was to create a noninstitutional, liminal space to which content could be posted. Rather than a grid, we developed the idea of an agora: a navigable cartesian space populated with content. Unlike a feed, content is populated organically throughout the space, encouraging an alternative mechanism for content consumption. The functions of the app are intentionally limited. We give users only the basic tools needed to develop organization, hoping to focus the platform to this intent. Though only in existence as a mockup, this app has the potential to be implemented into 307 and other spaces, encouraging agonistic discourse and the development of ongoing organization.



Public space is in a state of crisis: the agora is slowly suffocating. There are many reasons for this crisis, but the taproot is the cessation of agonistic political conflict due to the teleology of modern neoliberal democracy. In simple terms, the teleology, or end, of modern liberal democracy is twofold—to create a boundless polity (a global democracy), and to eliminate political conflict through the rational mediation of political settlements. Put differently, the end of democratic ideals is a post-political world in which the political is eradicated in favor of the rational. But these ideals, dating back to the Enlightenment, have become practico-inert and are failing to serve the needs of a modern networked public. We need to re-envision what it means to be a citizen of a democratic public. In order to do this, we need to rearticulate and assert the value of conflict as the central modality of a properly functioning democracy. We need to create more public spaces where citizens of the polis can be political every single day.

Hannah Arendt, in *On Revolution*, recognized that America's public space had shrunk both in size and number due to the very structure of the United States' democratic system. Given only one or two days a year to be political and exercise one's share in popular sovereignty, the political and revolutionary spirit of Americans began to atrophy. "If the ultimate end of revolution was freedom and the constitution of a public space where freedom could appear," Arendt writes, then "no one could be called happy without [their] share of public happiness, that no one could be called free without [their] experience in public freedom, and that no one could be called happy or free without participating, and having a share, in public power." Indeed, our representative democracy has internally failed to provide its citizens with proper participatory models.

Upon coming to a similar realization, Calvin and I decided to take matters into our own hands. No longer able to count on institutions to provide us with proper public space, we decided to create our own in the Brown 307 open studio. Brown 307 was intended expressly to be an "open studio", but we felt it never was truly open—there was no space for discourse, nor for conflict. Students used the room in a pseudopublic fashion, occupying the same space but ultimately isolating themselves in their own digital silos. We wanted to bring a sense of conflict, mediation, and discourse into the room, fulfilling the promise of a truly "open" studio.

A Conversation

chicken soup for the critical soul

<https://bit.ly/2HPPLLIP>

CH: I guess we'll start discussing the ~~open~~ toolkit we've been developing throughout the course of this semester.

DH: yeah, let's do it. we've managed to create a relatively successful ~~open~~ space in Brown 307. how can we use the strategies we developed for this room to ~~open up~~ other spaces?

CH: From the get go, we discovered the power of demarcating space. We used pink tape to signify that an area was ~~open~~ to discourse, and I think the simplicity of this strategy makes it suited to contexts beyond 307.

Furthermore, we identified the power of emptying spaces of institutional significance. We talked about creating anti-occupations. I think this is an aspect I want to explore further. How can we develop strategies to vacate?

DH: The only way to vacate a space is to first occupy it. Maybe a better question is, how can we get others to vacate? How can we create ruptures in the way others use space? I think one strategy is to provide tools for others to voice their opinions and shape their own spaces.

As you mentioned, in 307 we used pink tape to provide an infrastructure for discourse. Other things we did were less obvious but probably more consequential: inviting others to participate, posing questions to be answered, providing food and tea, etc.

So I guess the question now (sorry for sidestepping your question a bit) is: how do we

create infrastructures for others to occupy? what mechanics make an ~~an~~ *~~open~~* infrastructure different from a *closed* infrastructure?

CH: A solid reframing of the question. One key distinction is the idea of invitation. At one point we discussed how Occupy Wall Street would have been different if instead of congregating in Zuccotti Park, protesters had made private office buildings the focus of their occupation.

The key distinction here is that a park does not need to be deinstitutionalized. One does not need an invitation to enter a park. On the other hand, an office building is purposefully closed. One way of ~~opening~~ spaces is simply to invite people in.

So generosity becomes a central tool. How this translates into infrastructure, I'm not sure. Does it mean propping ~~open~~ a door, serving free food, developing a kind of hobo code that functions as an invitation? Perhaps. Still this only attempts an answers for spaces we inhabit.

DH: Yeah, generosity and invitation are key ideas here. Those are methods of forming communities, which in themselves are a kind of infrastructure for shared action. Maybe we first need to look at what makes a community effective and participatory.

As you suggested, I think the way to do this is via small actions—the endemic, as we once called them. Things like propping ~~open~~ doors, writing contribution guidelines, asking questions, developing ~~open~~ file formats, letting others have influence over our spaces.

CH: True. These endemic strategies are important, particularly ~~open~~ file formats, yet they are only one step towards developing agonistic spaces.

When I think of infrastructure, I think of something that is ubiquitous, something with high capacity. Is it possible to implement macro strategies alongside the endemic. I want to prop open doors. At the same time, I want to plaster your public domain logo over billboards.

DH: It's possible—we need both. Little actions, big movements. Those things are relational; they inform each other. I think F/LOS is a good model of this IRL. lots of people making small changes for their own benefit, while at the same time building better worlds for many other users

CH: Indeed. No matter the setting, it seems to be a relational issue. It is also an issue of reward systems. People in FLOS communities find value in participation—making things people want to use. The result is an ecosystem that incentivizes development based on community need.

One could further argue that the success of FLOS has to do with infrastructure. Github is an incredible platform. It makes me wonder if feature such as the issue queue and pull request could be integrated into platforms trying to leverage ~~open~~ collaboration in other fields

DH: Yea I think those mechanics are super valuable for real life. Makes me think of our “actionable content” strategy on the Agora platform—means of establishing actionable steps to

follow up/follow through on ideas. Similar things must be applicable to fields other than design & eng

This is where openness is really really important. When everyone in a community is willing to share and collaborate, it becomes much easier to take action and follow through on things that need to be done. ~~Open~~ things are more inclusive and have more potential than closed things

CH: It becomes a recursive process. The more ~~open~~ spaces that are created the more opportunity there is to source solutions that may ~~open~~ others.

And that's exactly the idea behind the "actionable content" strategies on Agora. Not only can they be used to ~~open~~ institutional space, they are intended to promote student organization.

CH: In fact, one of the big motivations behind this project was to develop infrastructure that would enable rapid student organization in response to ongoing events.

When the Department of Homeland Security made a surprise visit to campus people were caught off guard. There was no unified student response. No one knew what to do. Fortunately this was a purely administrative visit, but one can easily imagine such an event escalating.

Clearly there is an urgent need for ~~open~~ tools that enable community sourced organization.

DH: Yeah, the Homeland Security visit was a pretty stark reminder of the reality we're facing: our institutions are too opaque and our communities are not organized enough to ~~open~~ them up. I think one of the reasons we are so bad at organizing is that we

now try to organize online

This is evidently an ineffective way of communicating and organizing groups of people, for the most part. Compare our generation to that of students during the Vietnam war, who were able to cause drastic change through massive and quick organization.

They did all their organization in the trenches (so to speak), person to person. But I think we can reshape our online tools to facilitate real-world meetings and organization

And I think we have to do this — online communities are great, but they rarely cause lasting change in our institutions (though maybe I'm wrong about this)

CH: I think this is perhaps a false equivalency. This year there have been significant student protest regarding the ownership of assault weapons. Arguably, the internet helped facilitate these actions.

However, I strongly agree that the only effective form of organization is one which is centered around engagement offline.

To this end, I think to Obama's 2008 campaign and the controversy surrounding Cambridge Analytica. Clearly these are not open tool kits, but they demonstrate the inherently political nature of networks and the challenges of maintaining open, agonistic spaces.

It seems that networks can easily influence offline behavior. The challenging thing is to ensure that influence is planar.

Is it possible to develop tools that both enable offline mobilization and challenge the authority of server-power?

DH: The only way to achieve both of these things is to mobilize people using custom tools—tools built by the community, operated by the community, and owned by the community.

This is why things like open source communities (not just FLOSS) and peer to peer networks are so valuable: they're able to operate outside of existing, problematic network-power relations. Building an open community is like building a new world.

CH: Yes to custom tools and peer to peer networks. I want to better understand methods for keeping networks open. I want to understand the incentives and value systems that make open networks succeed. this is what I am trying to ask: can we make tools that encourage these values?

Can these tools also help open spaces that are currently closed? We attempted to answer this in the context of 307, but can we extrapolate from Agora's tool kit and develop a more universal system that can be infinitely customized and iterated to fit other spaces?

DH: I don't know if there's any way to generalize & create a universally adaptable toolkit. There are lots of types of closed spaces and intensities of hegemony/power. I really think the solution lies not in the tools we use but in the intents with which we act in our daily lives

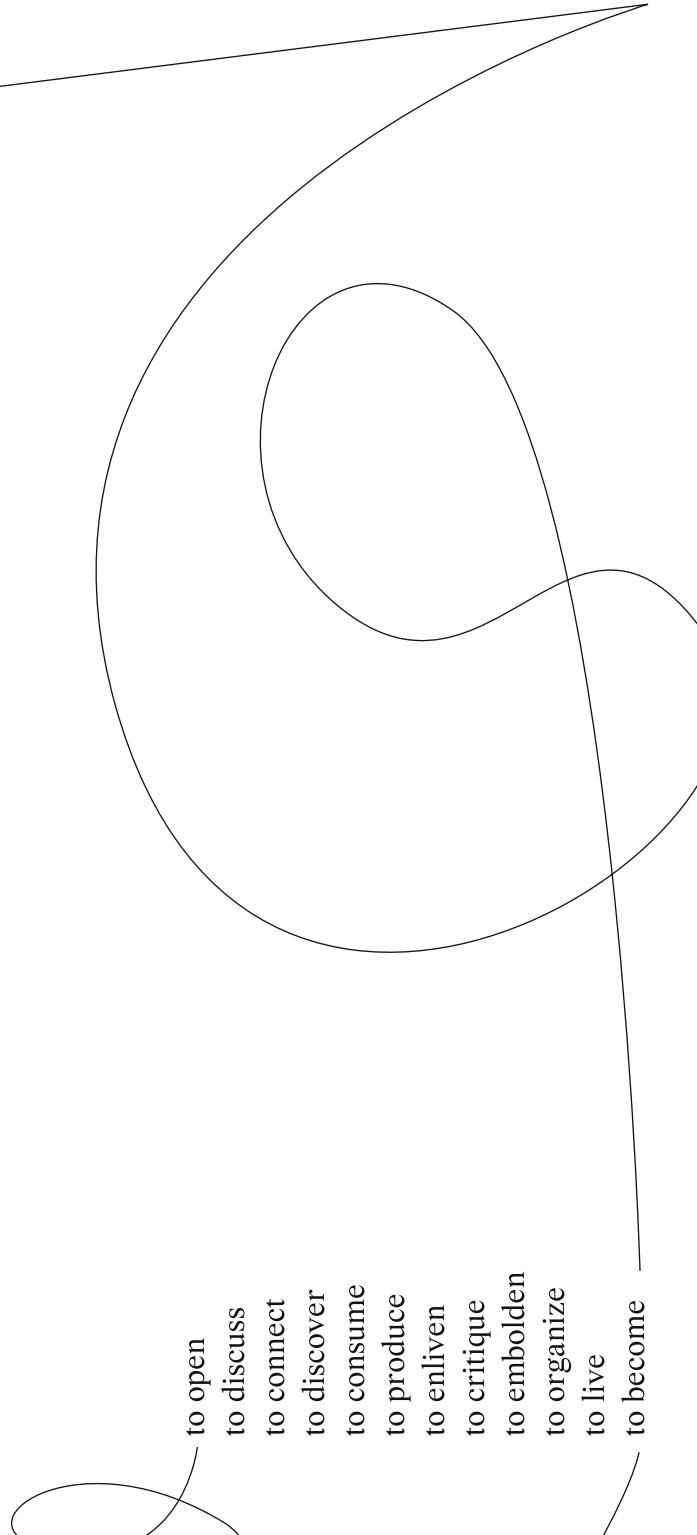
For example, we can't create a more open culture just by putting Public Domain stickers all over and inviting others into a space for a time. It needs to be constant. We can't just be open, we need to live open

CH: Yes. Tools don't have to be actions, or stickers. They can be behavioral. Maybe what we are getting at is the need for an open philosophy.

DH: Right. What we need is a lot of people, all at once, living and acting in ~~open~~ ways, inviting others to collaborate on and discuss the things that move society forward. This can mean lots of things: propping doors, making ~~open~~ file formats, sharing source files, providing tea...

...building software, provoking action, stacking rocks, organizing strikes, designing fonts, sharing food, criticizing institutions, making salads.

An Application

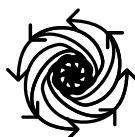


to open
to discuss
to connect
to discover
to consume
to produce
to enliven
to critique
to embolden
to organize
to live
to become

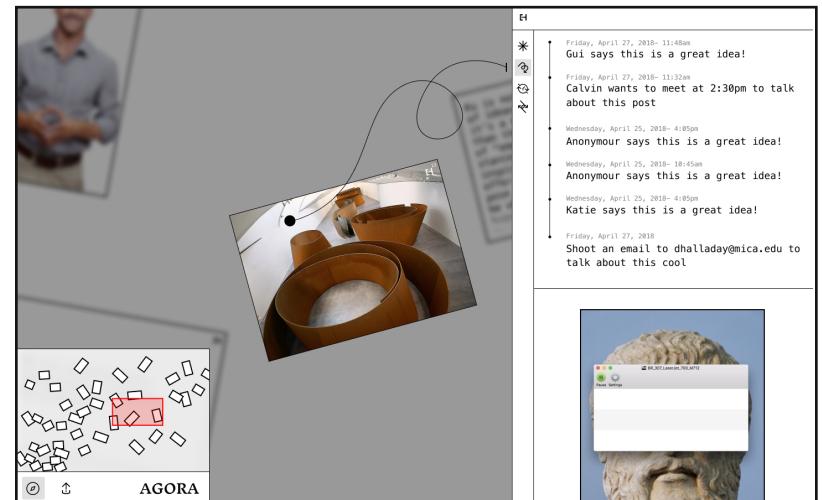
AGORA



A experimental tool for generating discourse



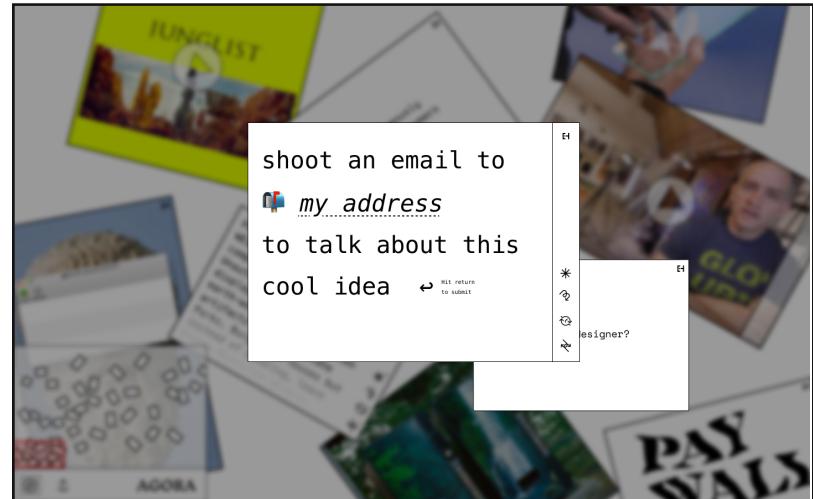
Agora is an experimental tool for the generation of discourse and passive discovery and connection of content.



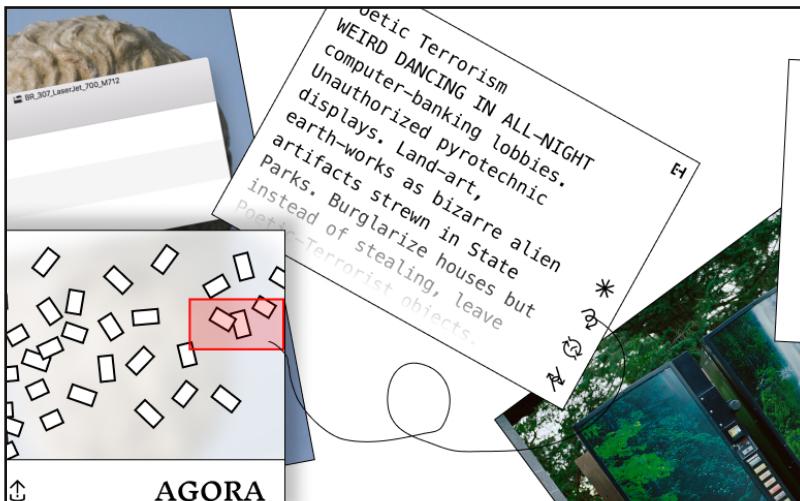
Its primary motivation is to allow agonistic discourse to happen naturally in the digital space, while encouraging physical organization and connection in the real world through the creation of meetings and messages.



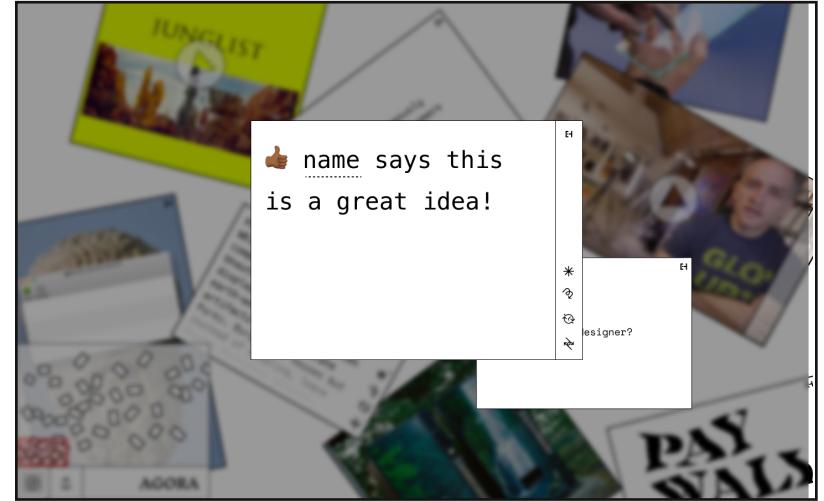
Navigate the digital space by dragging with your mouse, or by using the mini-map in the lower left corner.



Create real-world action by sending the poster of an object an invitation to meet and discuss the object's contents



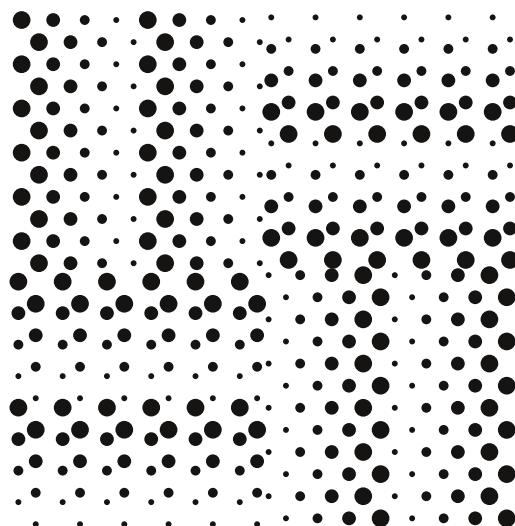
Connect one post to another using the pickwhip feature, which allows a user to literally "draw" connections between disparate pieces of content.



Add value to objects by marking them as great ideas. A user's identity is attached to every action in the product to encourage real-world association.

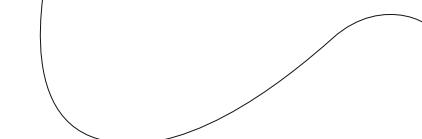
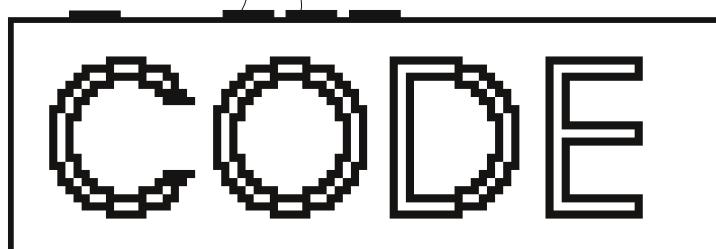
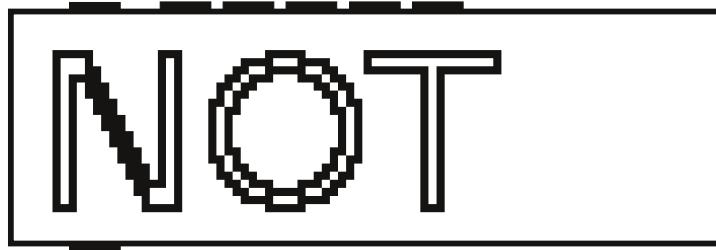
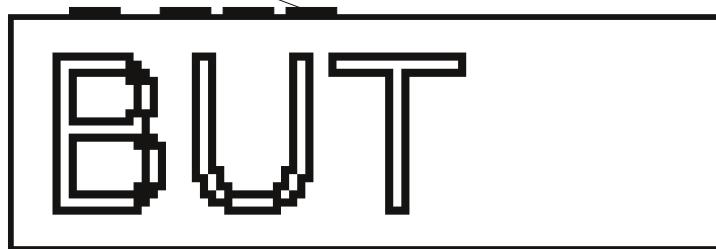
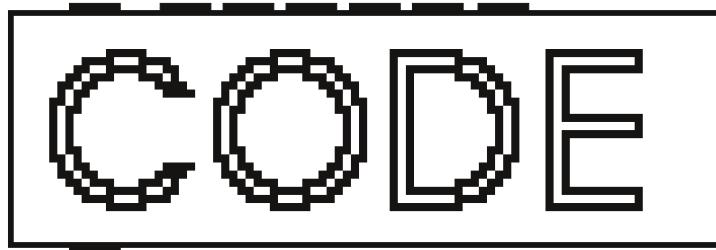
NODEBOXING

By Pragun Agarwal



Code can be intimidating for anyone. Dabbling with the idea of writing functions and commands to generate a visual rather than crafting one is a unique workflow for a designer.

Nodebox is an open source software that tries to bridge this gap in a effective and intuitive manner. Through simple connections between a selection of pre-set commands, it provides the user with limited yet a diverse set of tools and opportunities to make generative art.



Each node has a single output and multiple input options like size, number, range etc. This allows the user to select what parameters they would want to connect and modulate with.

NO WAY! NO WAY!

N N N N N N N N N N N N
W W W V V V V V V V V V V
O O O O O O O O O O O O

W W W W W W W W W W W W

NO WAY! NO WAY! NO WAY!

A A A A A A A A A A A A
N N N N N N N N N N N N
V V V V V V V V V V V V
Y Y Y Y Y Y Y Y Y Y Y Y

There was a sense of surprise and wonder, each time a different connection was made between these nodes. The user can see visual output as and when they make any kind of changes.

A A A A A A A A A A A A
W W W W W W W W W W W W
Y Y Y Y Y Y Y Y Y Y Y Y

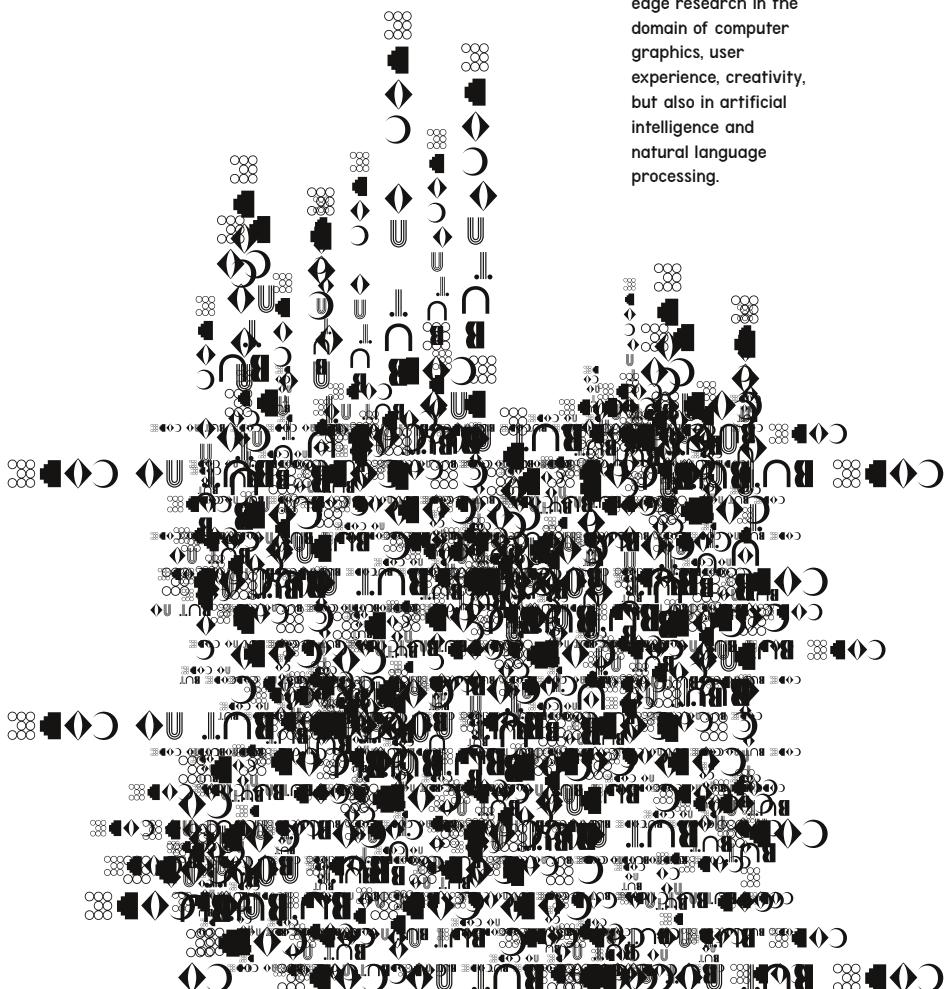
W W W W W W W W W W W W

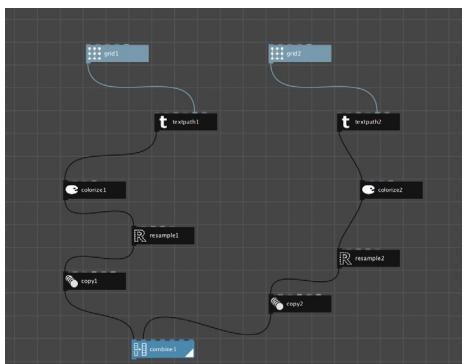
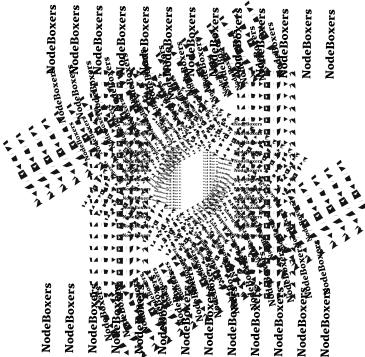
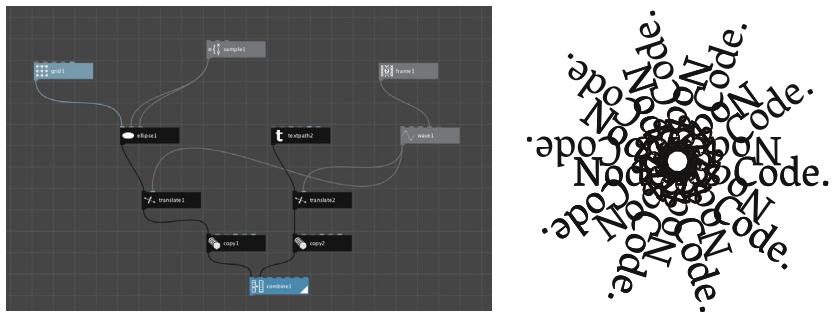
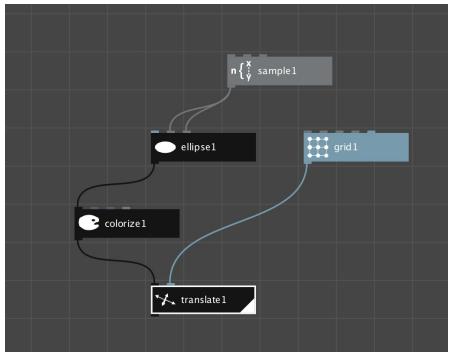
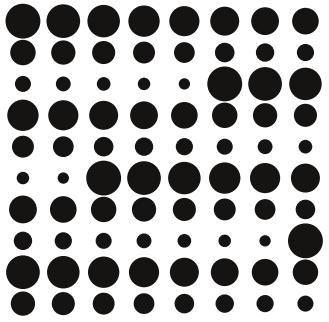
A A A A A A A A A A A A
Y Y Y Y Y Y Y Y Y Y Y Y
! ! ! ! ! ! ! ! ! ! ! !



NodeBox is developed by the Experimental Media Research Group, a cross-domain research group associated with the Sint Lucas School of arts of the Karel de Grote-Hogeschool (Antwerp, Belgium).

EMRG has been active since 2004 developing NodeBox and doing cutting-edge research in the domain of computer graphics, user experience, creativity, but also in artificial intelligence and natural language processing.





N O D E
D E B O
D E B Q
B O X
L B O X
X

B O X

! 1

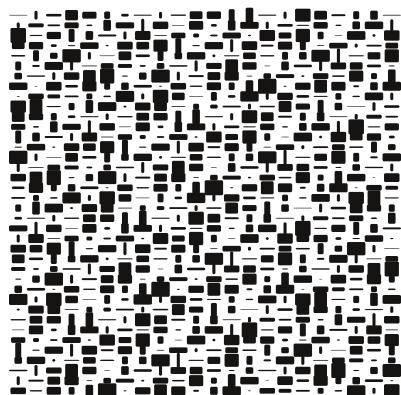
卷之三

卷之三

This is nodeB05. This is nodeB05. This is nodeB05. This is nodeB05.

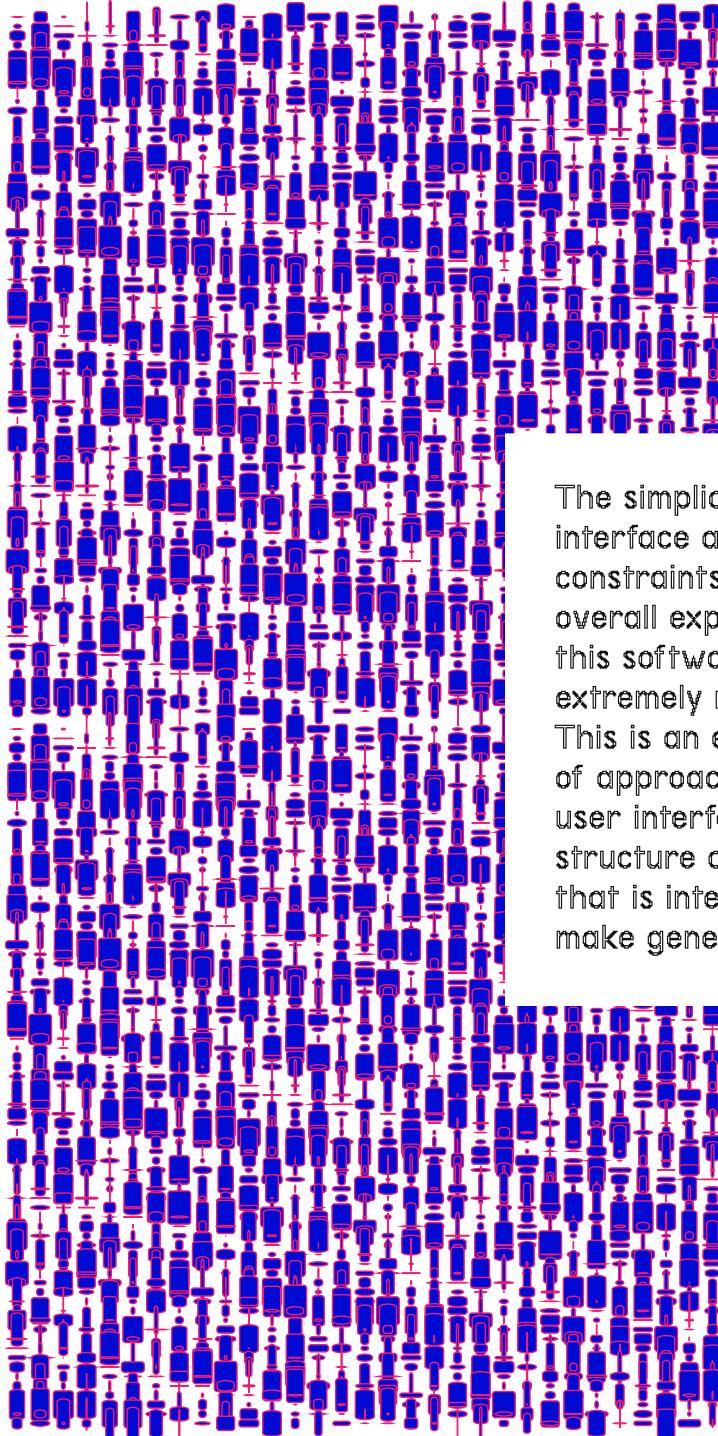
S N O D D S

E B O X
O X
C X
Z O D
Z O D



Nodeboxing

97



98 GD 399.01: Open Sourcery

The simplicity of the interface and constraints makes the overall experience with this software extremely rewarding. This is an effective way of approaching the user interface and structure of a software that is intended to make generative art.

A Running List

MAZZY BELL

I am and have always... jotting down mi...

I am and have always been a habitual note taker and list maker, constantly jotting down miscellaneous pieces of information that I think will be useful in the future. For the most part, these seemingly useful facts are not quite useful at all; I rarely really look back at them and when I do I find them unorganized and relatively incomprehensible. I found a note recently that just read "magician's 25th" and I truly have no idea what that means, nor do I ever think I will.

One note that has gotten particularly extensive and out of hand, is an open sourced themed document, which was started in September of 2017. I had just learned the meaning of open source and felt fervent in reading all that I could on the subject, potentially overly enthusiastic as I often am when I find something new to fixate on. The initial entry reads "open source!!!!!! I was doing this already and I didn't even know" followed by like 6 links, some of which do actually deal with open source things, others that do not. For the record, I would like to add that I am not quite sure what I meant by "doing" open source, but you get the gist.

The note continues on with a lot more links that I will probably never click on and some more stream of consciousness ramblings. It wasn't until January of this year, 2018, when I entered this class, Open Source Design, that I found my note out of control and off the

I am and have always... jotting down mi...

that my note spun out of control and off the rails.

I think in my own investigation of what it meant for something to be open source, in the few months leading up to January, I was discovering things that probably didn't even know they were open source, a lot of hardware and general instructions to building things that most of the time predated the phrase itself. I certainly don't develop software and I only very recently finally figured out how to use Github. While I was engaging with open source content I wasn't really integrating, or trying to integrate myself in the "open source community". I am not even sure I knew there was an "open source community" until this class began.

At some point, I'm going to say a third through the semester, my note turns sour. What started out as frivolous fragments and one liners, quickly snowballed into frenzied rhetoric brimming with f bombs occasionally typed out IN ALL CAPITAL LETTERS, just to emphasize the metaphorical steam that was blasting through my ears.

My initial planned contribution to this book was an essay, a mashup of all of the writing I had so feverishly been typing away at all semester. However, when I pulled the paragraphs out of my note it was 16 pages long and each chunk of writing had a huge disconnect to the next. It was also one of the grumpiest things I had ever written,

**I am and have always...
jotting down mi...**

highly unproductive at the very least.

In the spirit of my instinctive list making, and knowing that it would be borderline torturous for anyone to read 16 pages of me yelling at them, I have gone through The Note and made a pros and cons list based on all of the weird things I have typed out on the benefits and downfalls of what it means for something to be open source. Obviously, the list is coupled with this long winded explanation of how I got here and a little bit of reflection. I hope you enjoy, and remember, if you get bored, because who am I kidding, this will probably be rather boring for most people, just be grateful you didn't have to read the other thing.

*the part in quotation marks are pulled directly from my open source note

Pros

- If you really wanted to build a house on your own, you could definitely do that without having to pay for the instructions or come up with them on your own
- If you really wanted to build all of the furniture in your house (the one you built on your own), you could definitely do that without having to pay for the instructions or come up with them on your own
- "there are so many dank fonts that are open source that I already had on my computer that I didn't even know were open source"

**I am and have always...
jotting down mi...**

- Inkscape is a better/more user-friendly vector creating software than illustrator
- If you are using something that is open source, and you don't like something about it, you can (in theory) just modify and amend
- Free/cheaper than things (software) that aren't open source
- "inherently collaborative :)" (I suppose this could be a con but for me it's a pro)
- "design/software that are open source are never really 'finished'; they can always be ameliorated into better, more helpful versions of themselves. While this could be true with other design/software that isn't open source, it is theoretically more likely for things to be improved based on the nature of what it means for something to be open source... anyone that finds a problem with something can fix it"
- Projects and iterations of things are far more documented and logged than things that are not open source

Cons

- "I really can't figure out how to use Github, I feel really goofy for not understanding it and I feel like normally it wouldn't matter but it feels quintessential to being involved in the open source community"

I am and have always... jotting down mi...

- While the term open source is seemingly universal, it isn't. Most people don't know what it is and, in my opinion, that takes away from so many aspects of what would otherwise be highly beneficial. This feels especially true since I often have a difficult time explain what open source is without simultaneously critiquing it

-The phrase 'too many cooks in the kitchen' is probably grossly applicable

- In regards to the open source community, "my assumption— because it is called 'open source' people (of privilege) who historically have never had issues speaking up and voicing their opinions, provide little room for people who have never had the space to speak when they feel necessary."

- In reference to the previous con, "it feels arduous to integrate yourself into the open source community, it is intimidating and while it seems that this community should be centered around learning, it doesn't really seem like it is."

-Most websites that I have come across that deal with hardware/software that are open source were certainly not easy to find and for the most part not very easy to navigate.

While I'm sure I could continue on, these were the things that felt the most veracious. In the spirit of open source I'm housing this list on a google doc (more accessible) for

I am and have always... jotting down mi...

-The phrase 'too many cooks in the kitchen' is probably grossly applicable

- In regards to the open source community, "my assumption— because it is called 'open source' people (of privilege) who historically have never had issues speaking up and voicing their opinions, provide little room for people who have never had the space to speak when they feel necessary."

- In reference to the previous con, "it feels arduous to integrate yourself into the open source community, it is intimidating and while it seems that this community should be centered around learning, it doesn't really seem like it is."

-Most websites that I have come across that deal with hardware/software that are open source were certainly not easy to find and for the most part not very easy to navigate.

While I'm sure I could continue on, these were the things that felt the most veracious. In the spirit of open source I'm housing this list on a google doc (more accessible) for anyone to add to. I may even put it on Github as well but that might take me a minute to figure out how to do...

<https://bit.ly/2rlgyHf>