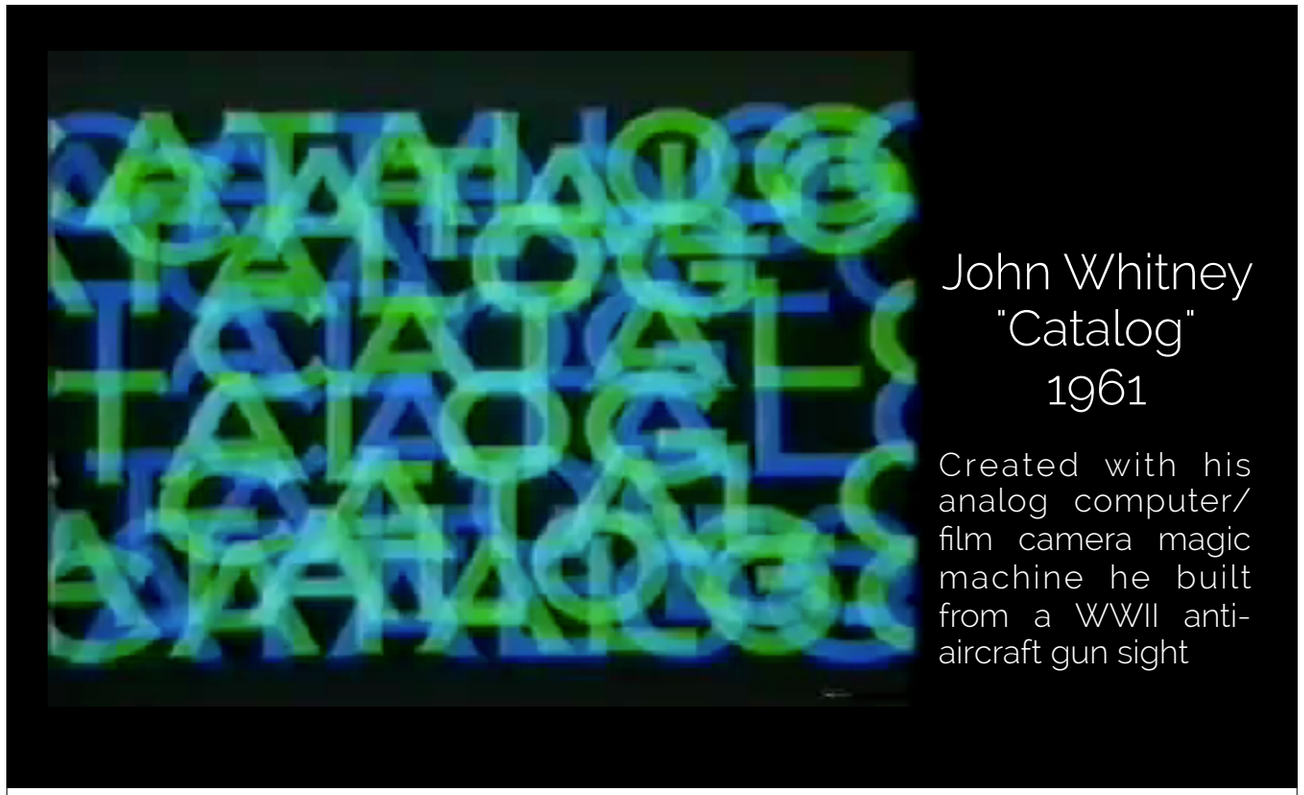


Talk by Andrew Ringler coming at 6pm

How Different Software Languages & Libraries Affect Your Creative Process

John Whitney's demo reel of work created with his analog computer/film camera magic machine he built from a WWII anti-aircraft gun sight. Also Whitney and the techniques he developed with this machine were what inspired Douglas Trumbull (special fx wizard) to use the slit scan technique on 2001: A Space Odyssey

https://www.youtube.com/watch?v=TbV7loKp69s

Andrew Ringler

How Different Software Languages & Libraries Affect Your Creative Process

—MIT IAP Jan 2015

I will post slides at http://andrewringler.com/

Hi. I am Andrew Ringler.

I am going to talk to you about creating art by programming.

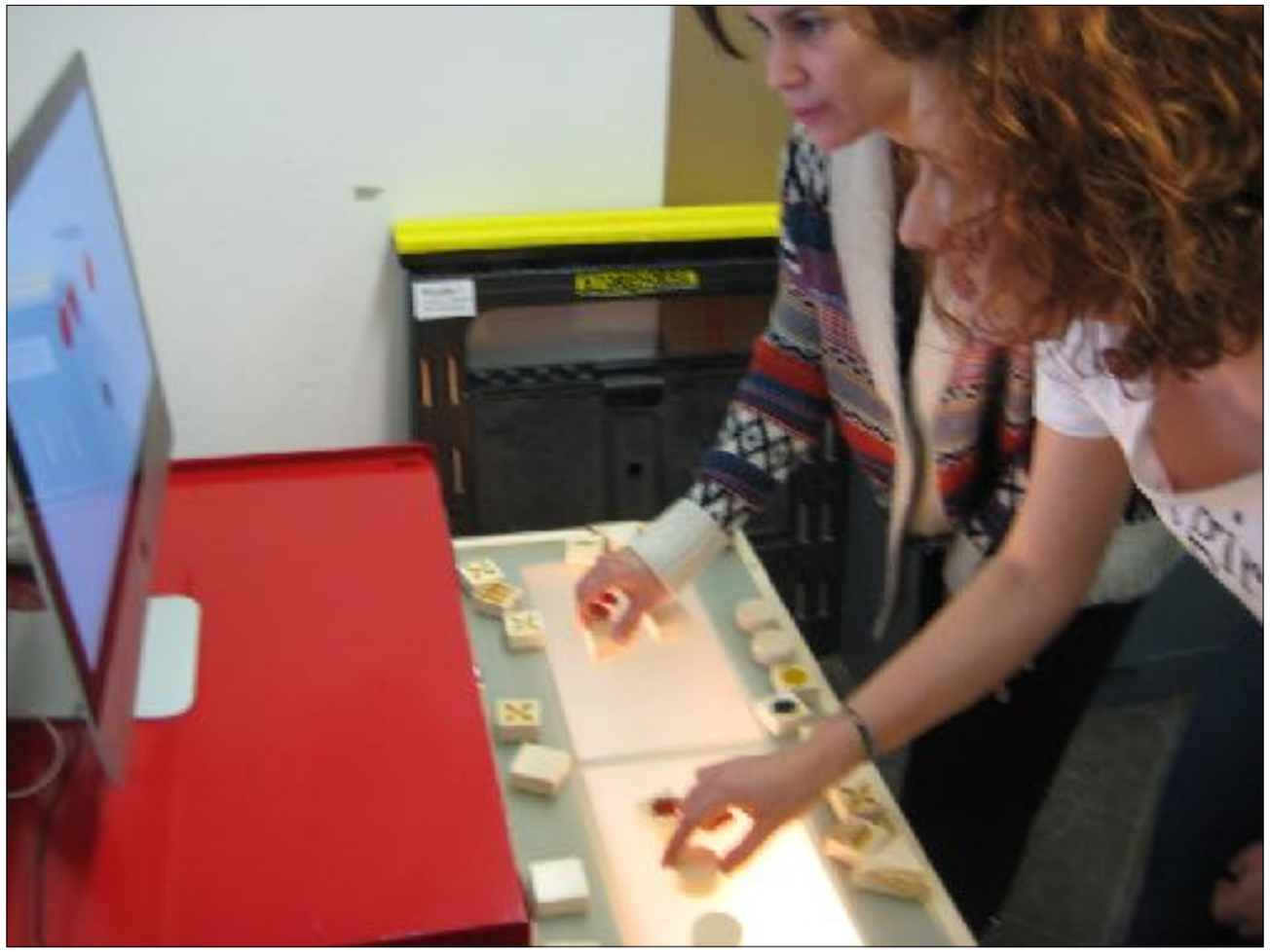
So, first a little bit about me, Andrew Ringler.

I do Improv.

Which means I can stand on a stage with 2 chairs.

I often make charcoal drawings

I make short films.



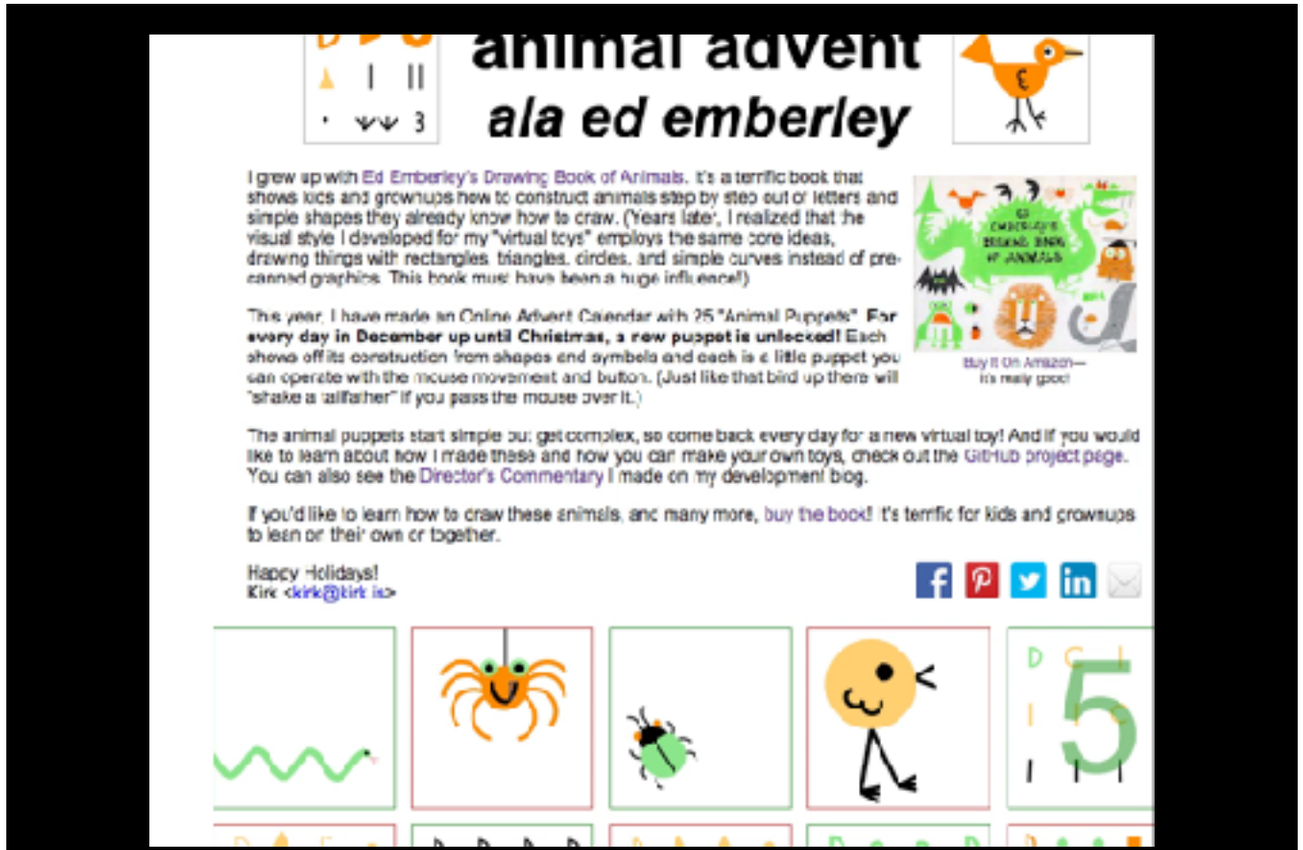
I make interactive, tactile & collaborative digital art

Programming to make Art

And all of these experiences and interests influence my work when I program to make art.

So. What do I mean by programming to make art.

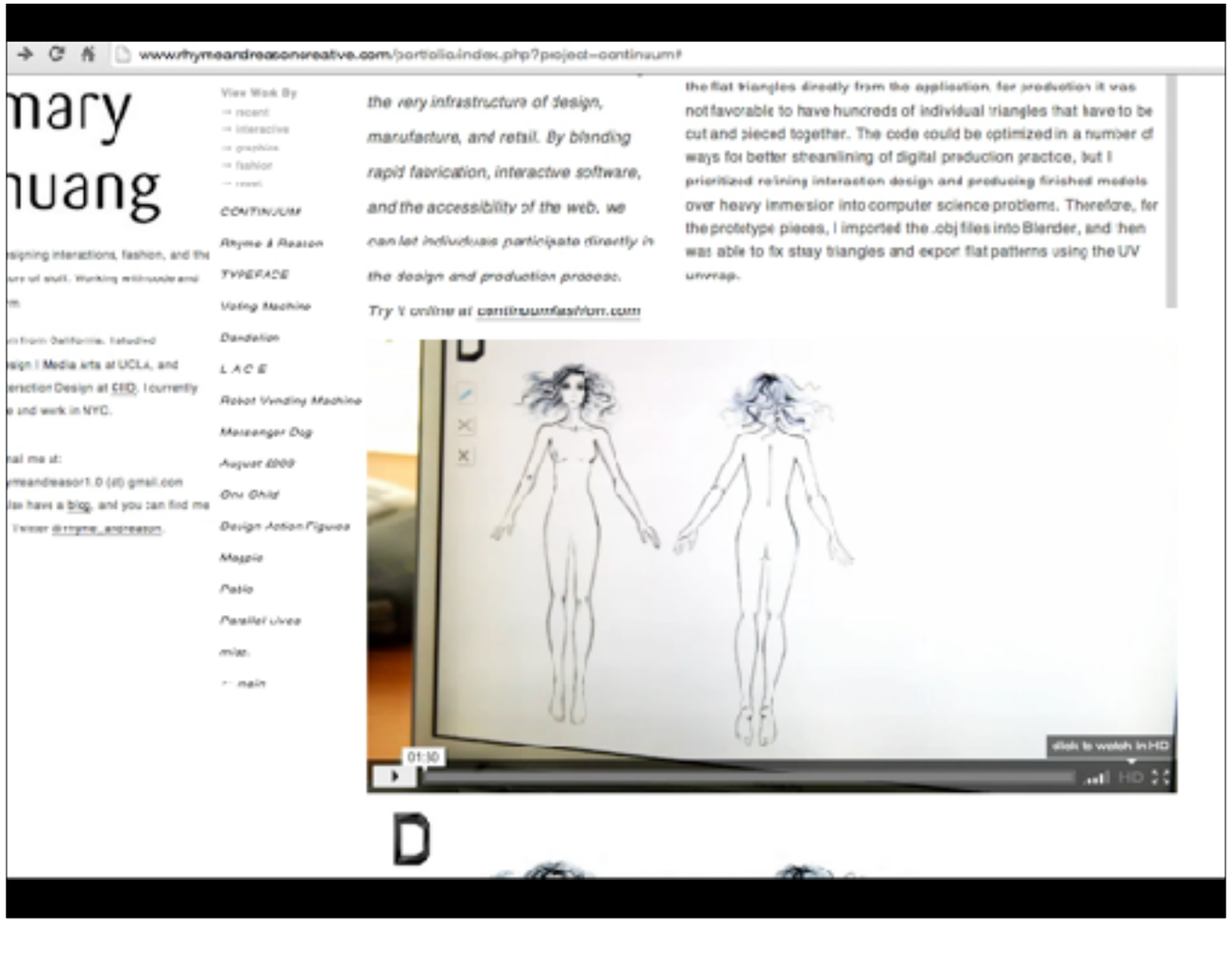
Here are a few examples of art created through programming:



Kirk Israel - http://advent.kirk.is/

An interactive Animal Advent calendar created by Kirk Israel coded in Processing.js

http://advent.kirk.is/



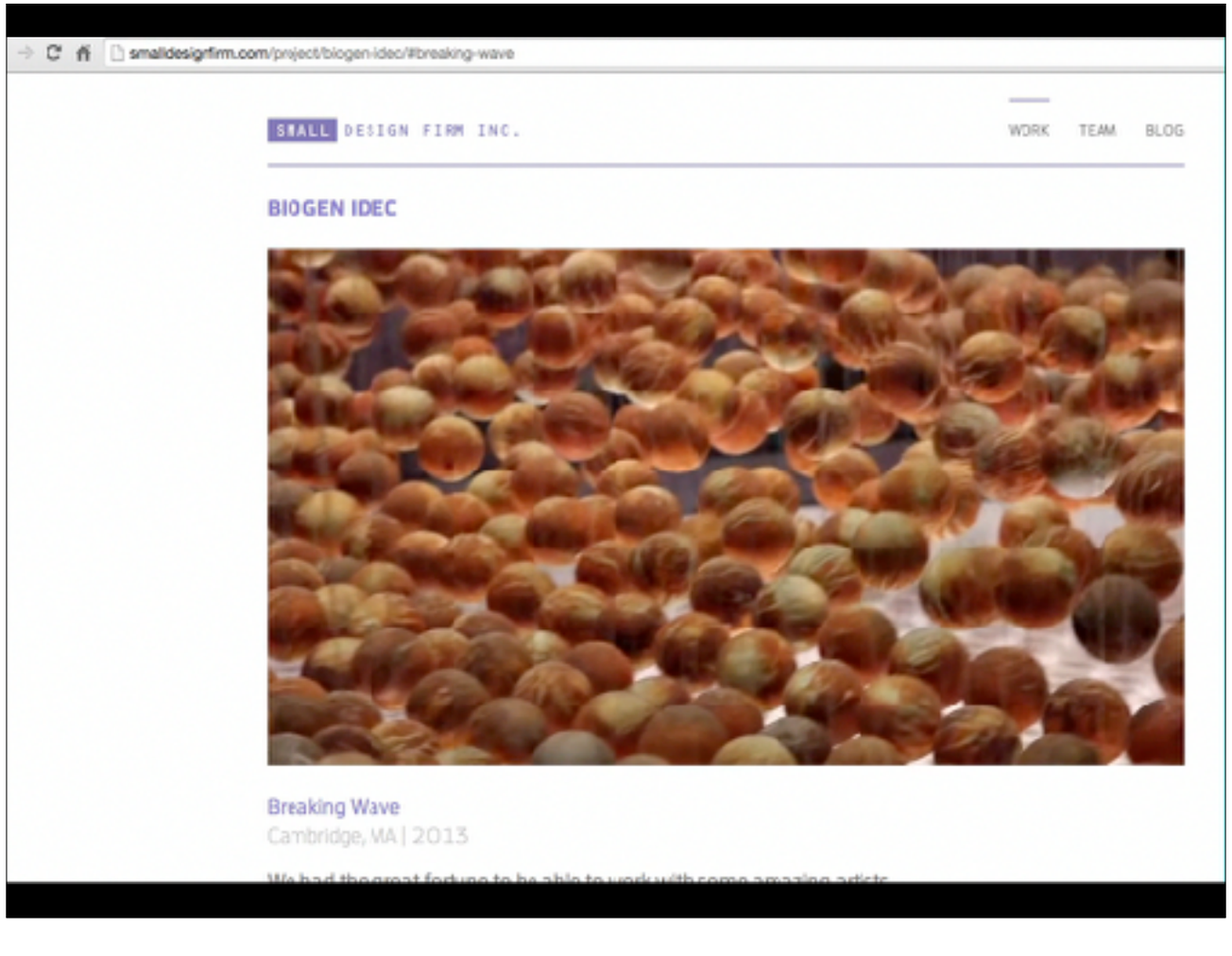
Interactive and algorithmic assisted dress generation by Mary Huang, written in Processing.

The software application was created in Processing. The generated dress models can be exported as .obj files. While you could export the flat triangles directly from the application, for production it was not favorable to have hundreds of individual triangles that have to be cut and pieced together. The code could be optimized in a number of ways for better streamlining of digital production practice, but I prioritized refining interaction design and producing finished models over heavy immersion into computer science problems. Therefore, for the prototype pieces, I imported the .obj files into Blender, and then was able to fix stray triangles and export flat patterns using the UV unwrap.

http://www.rhymeandreasoncreative.com/portfolio/index.php?project=continuum#

MIT media lab branding by TheGreenEyl and E Roon Kang made in Processing.

The new visual identity of the MIT Media Lab inspired by the community it comprises: Highly creative people from all kinds of backgrounds come together, inspire each other and collaboratively develop a vision of the future.

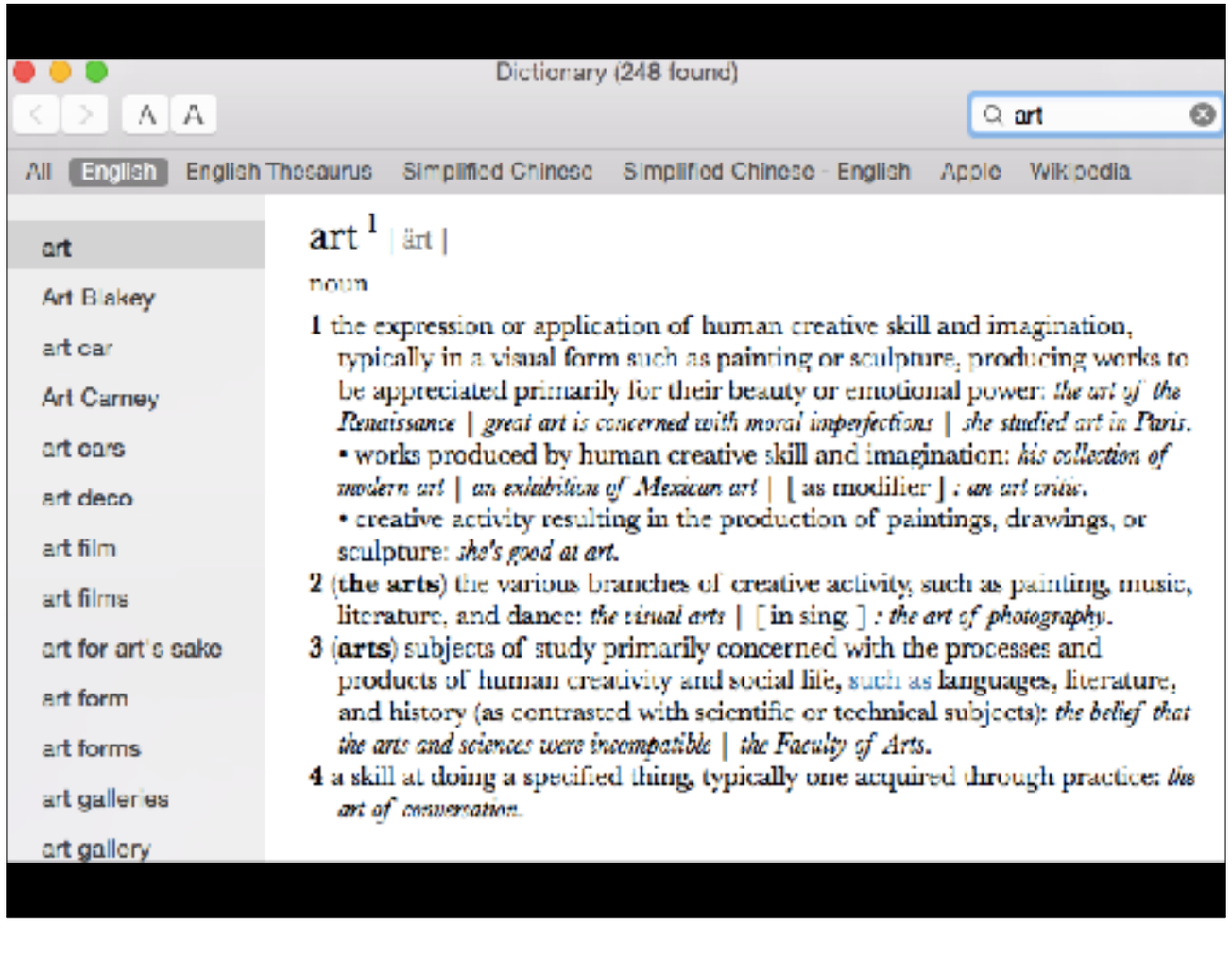


Small Design Firm, Plebian Design and Hypersonic Engineering for the Biogen IDEC headquarters lobby

http://www.hypersonic.cc/projects/breakingwave

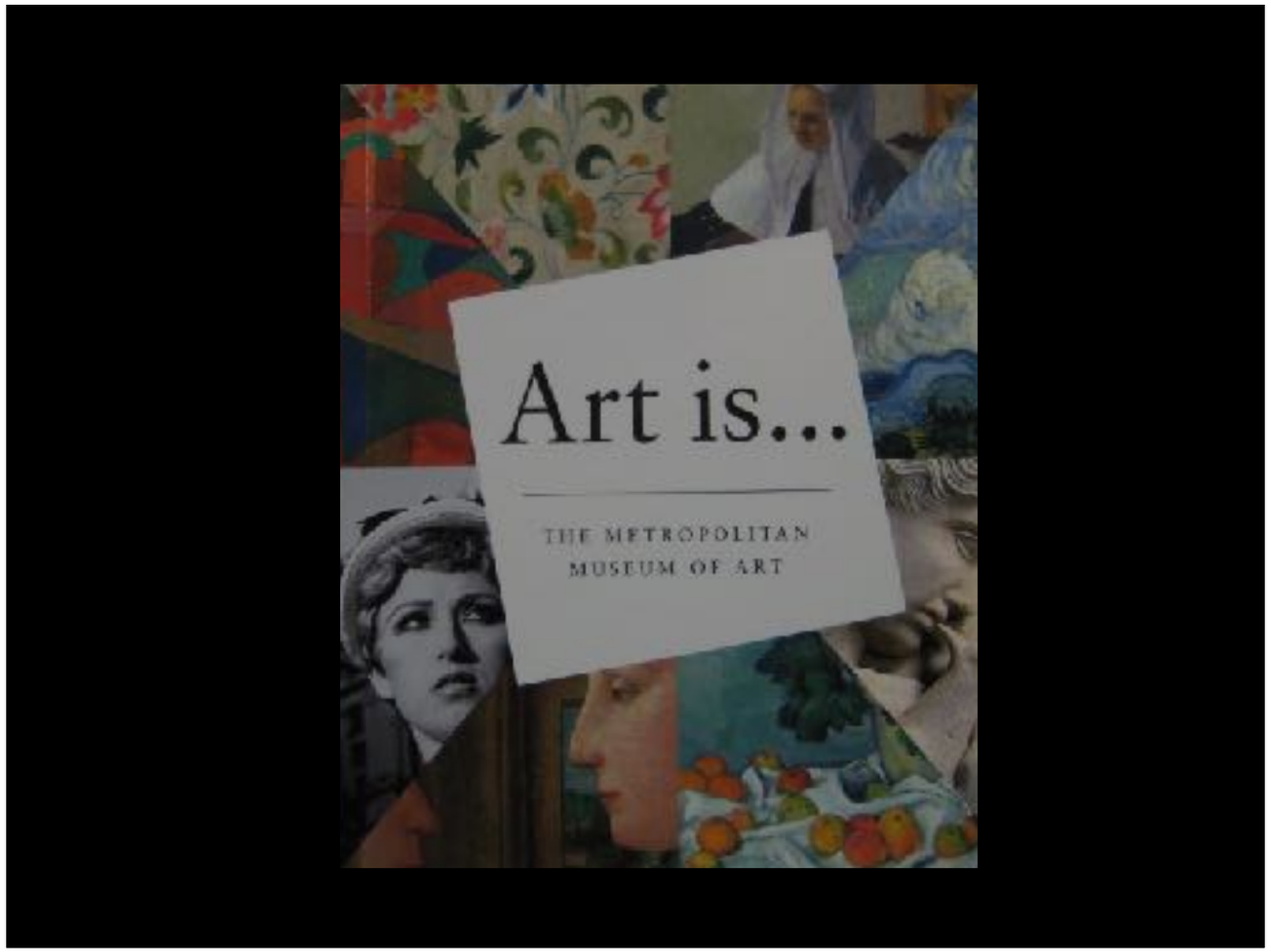
http://smalldesignfirm.com/project/biogen-idec/#breaking-wave

What is Art?

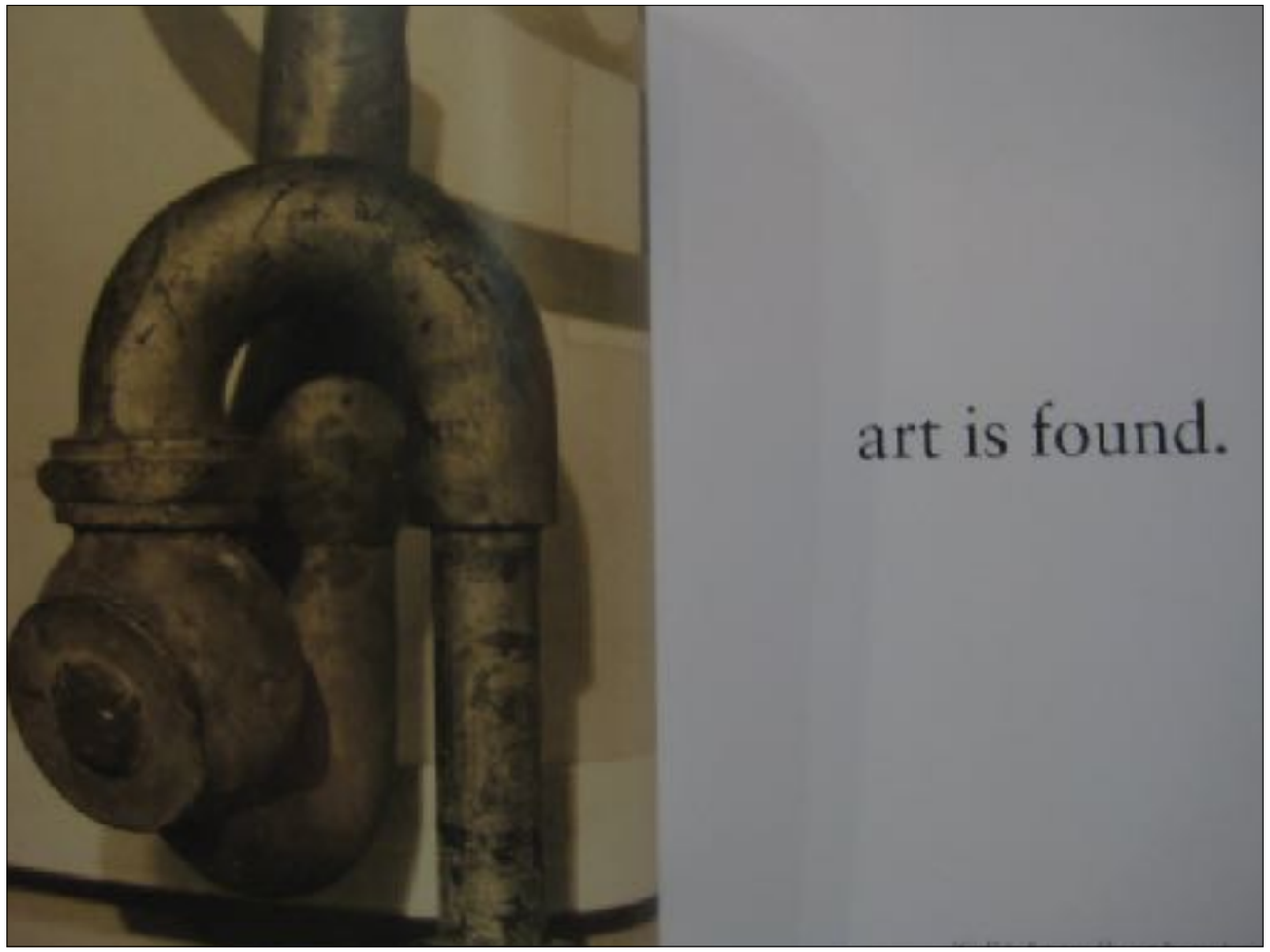


Art - from Apple Dictionary

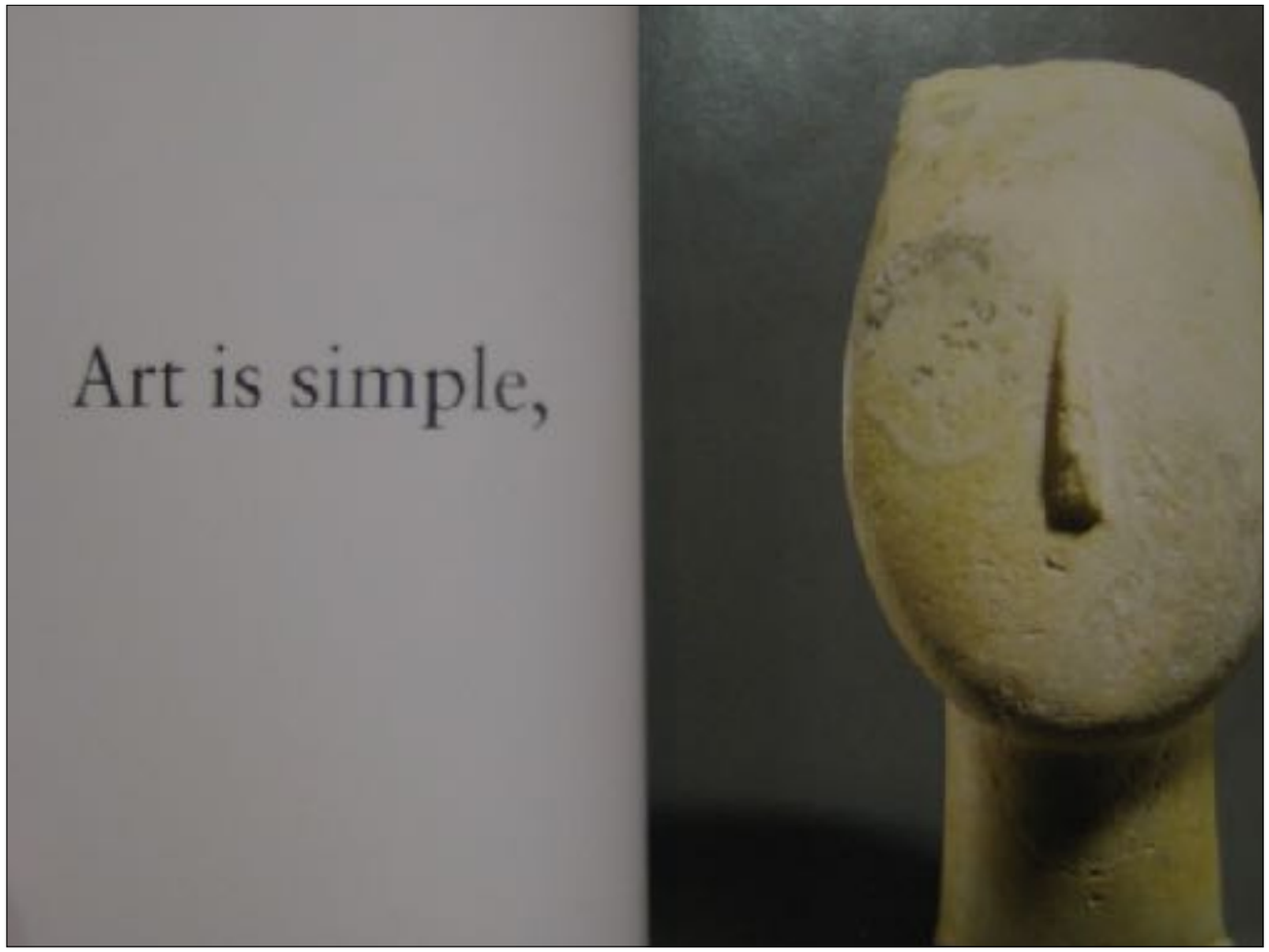
the expression or application of human creative skill and imagination, typically in a visual form such as painting or sculpture, producing works to be appreciated primarily for their beauty or emotional power



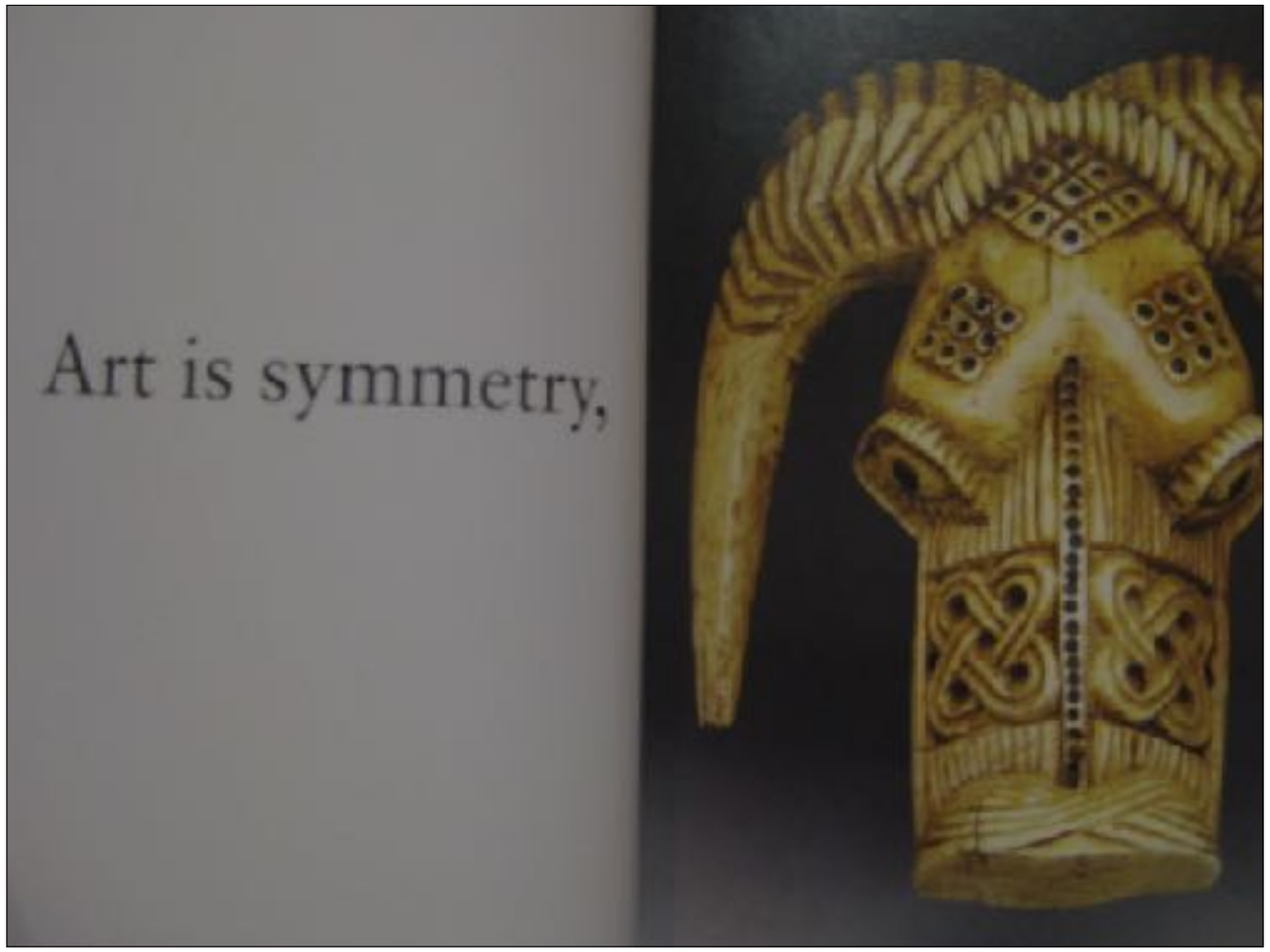
I have a book called Art is… from the Met



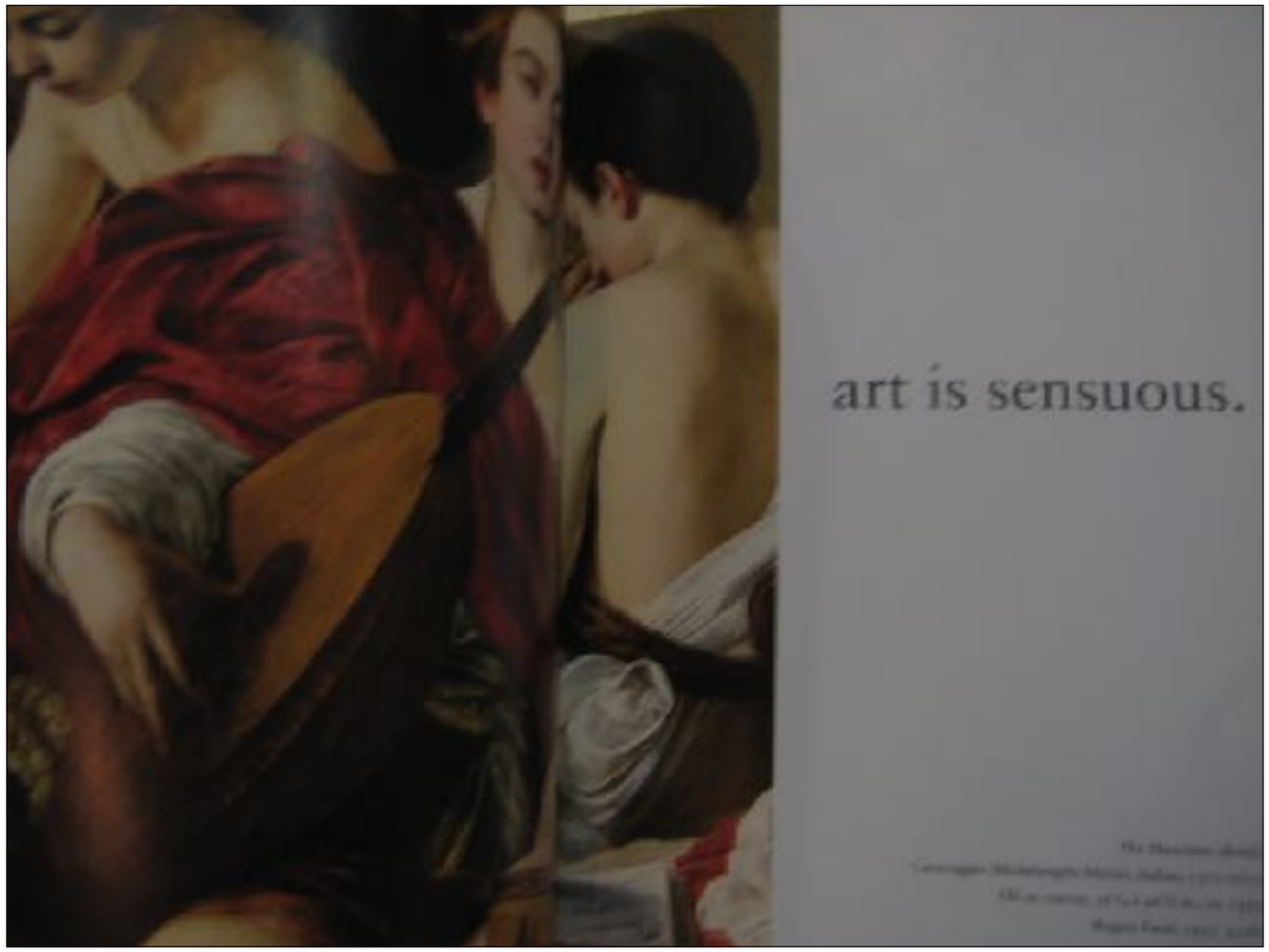
art is found



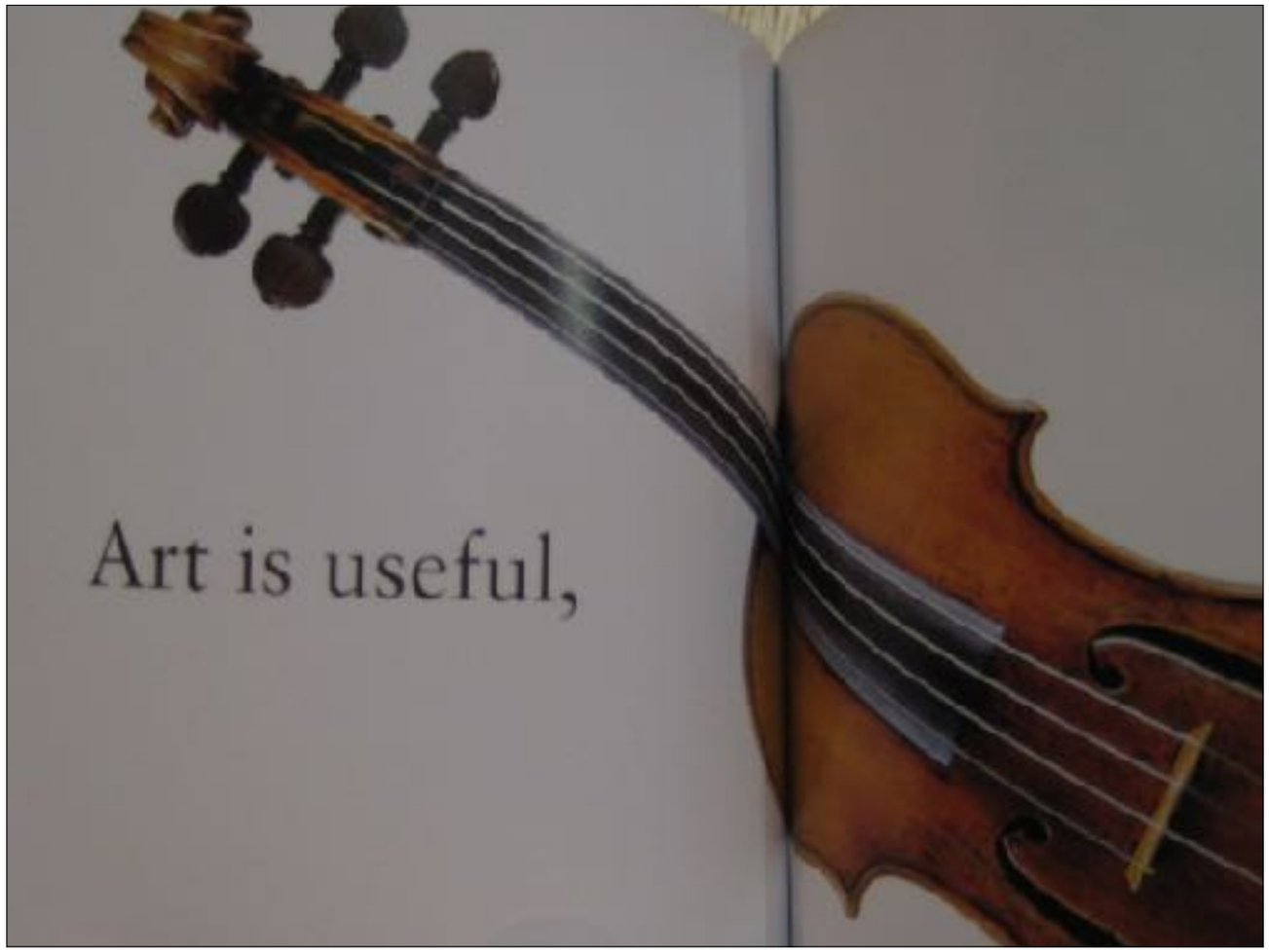
Art is simple



Art is symmetry

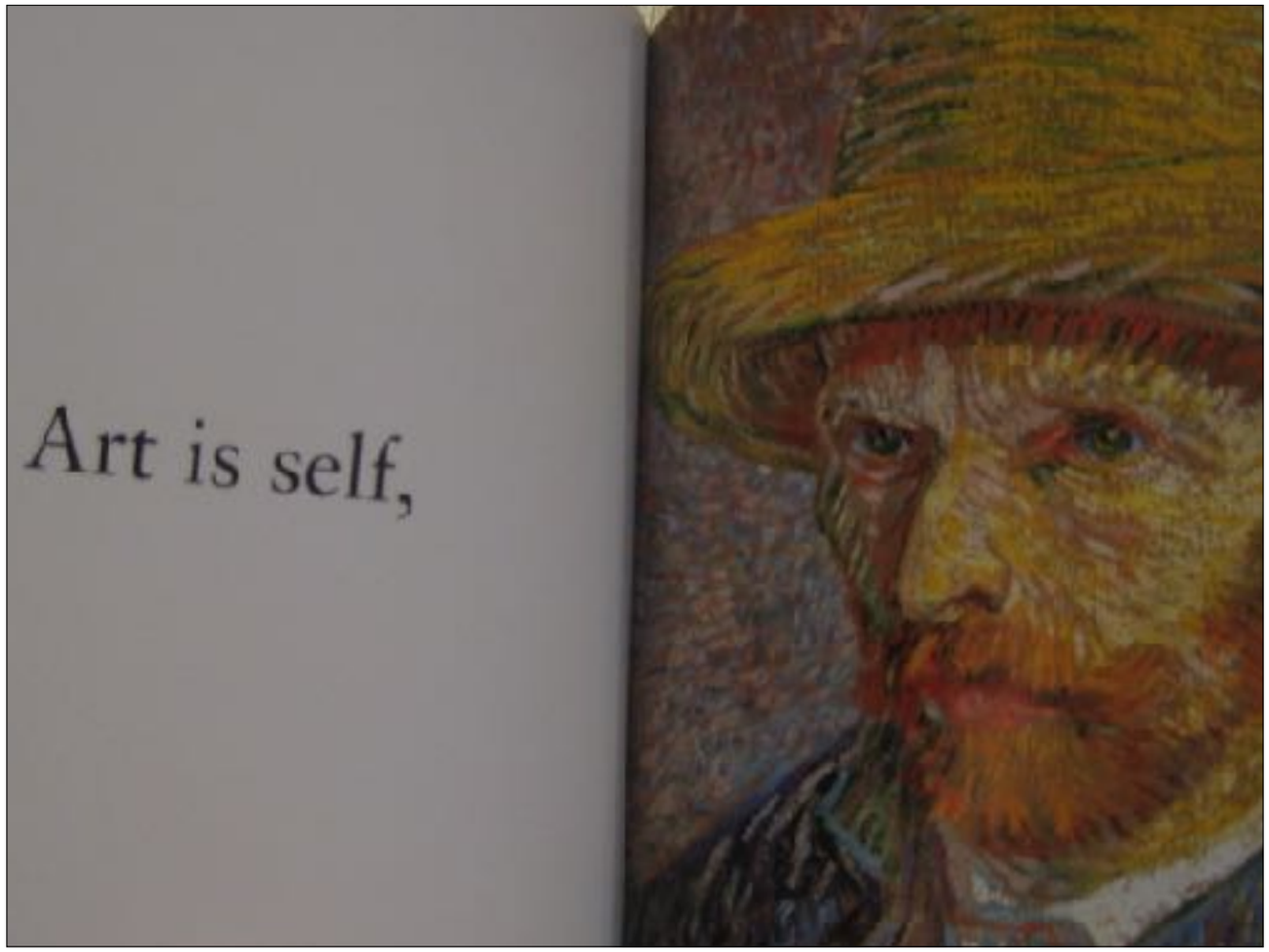


Art is sensuous

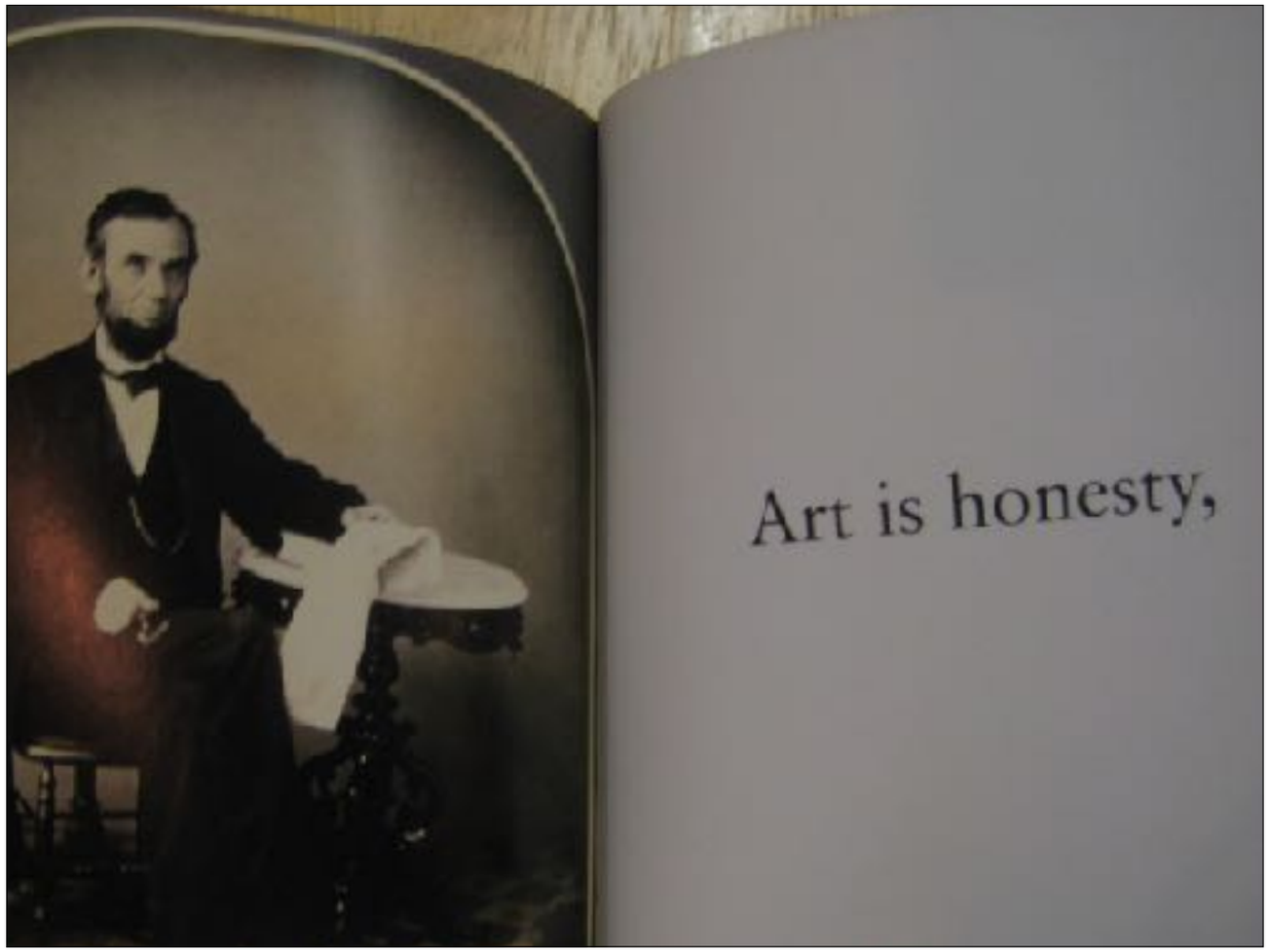


Art is useful

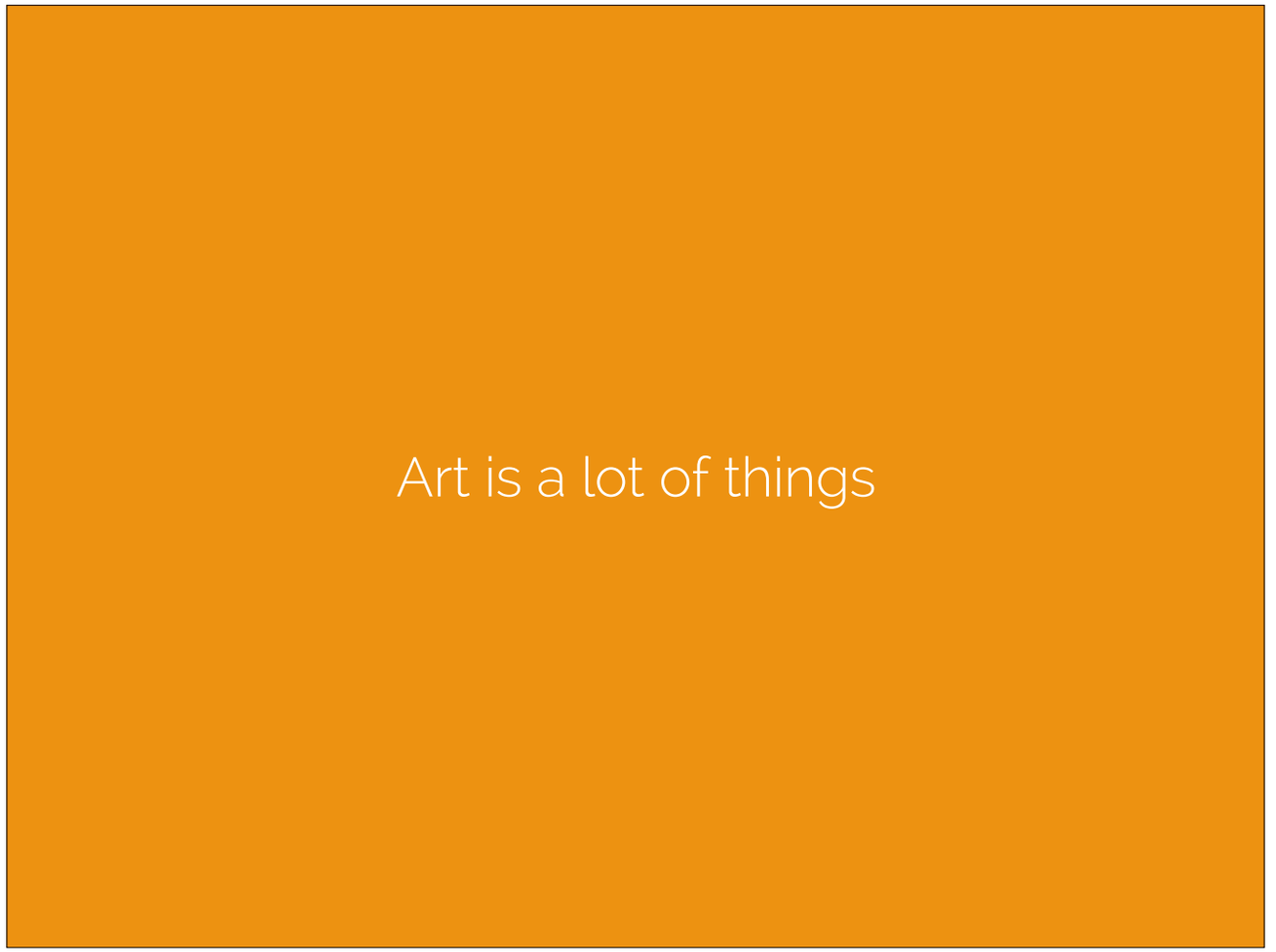
Art is expressive



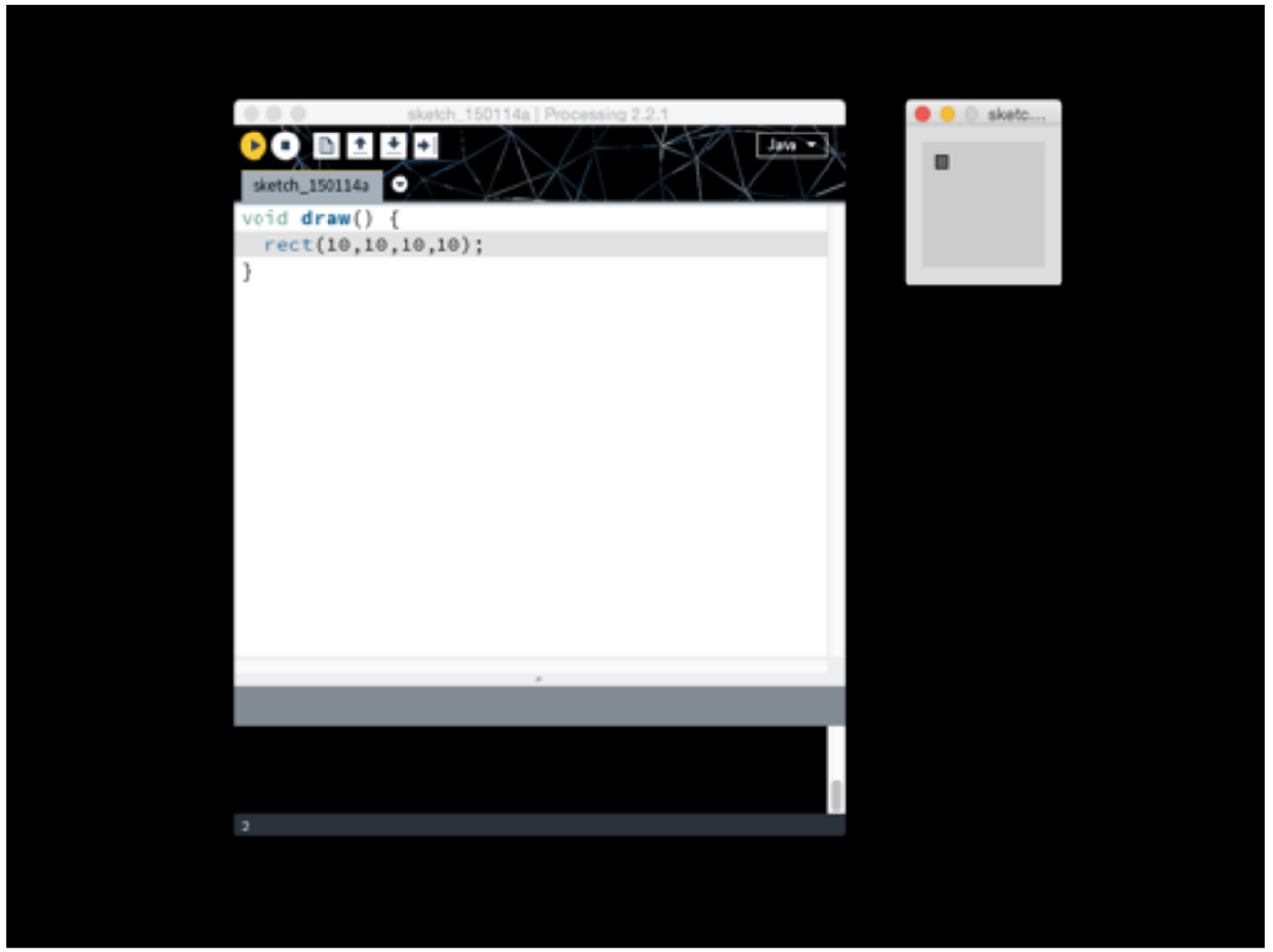
Art is self



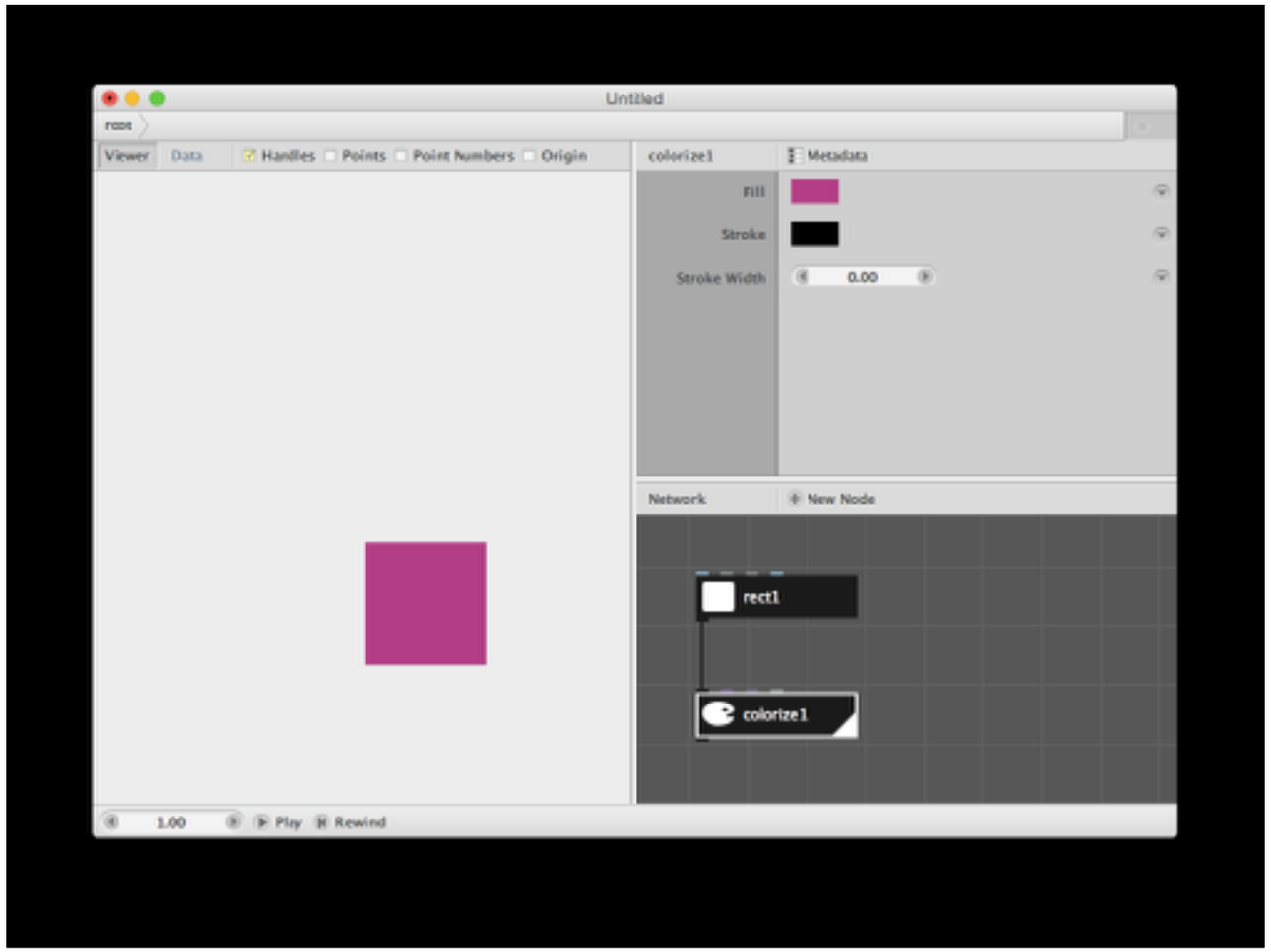
Art is honesty



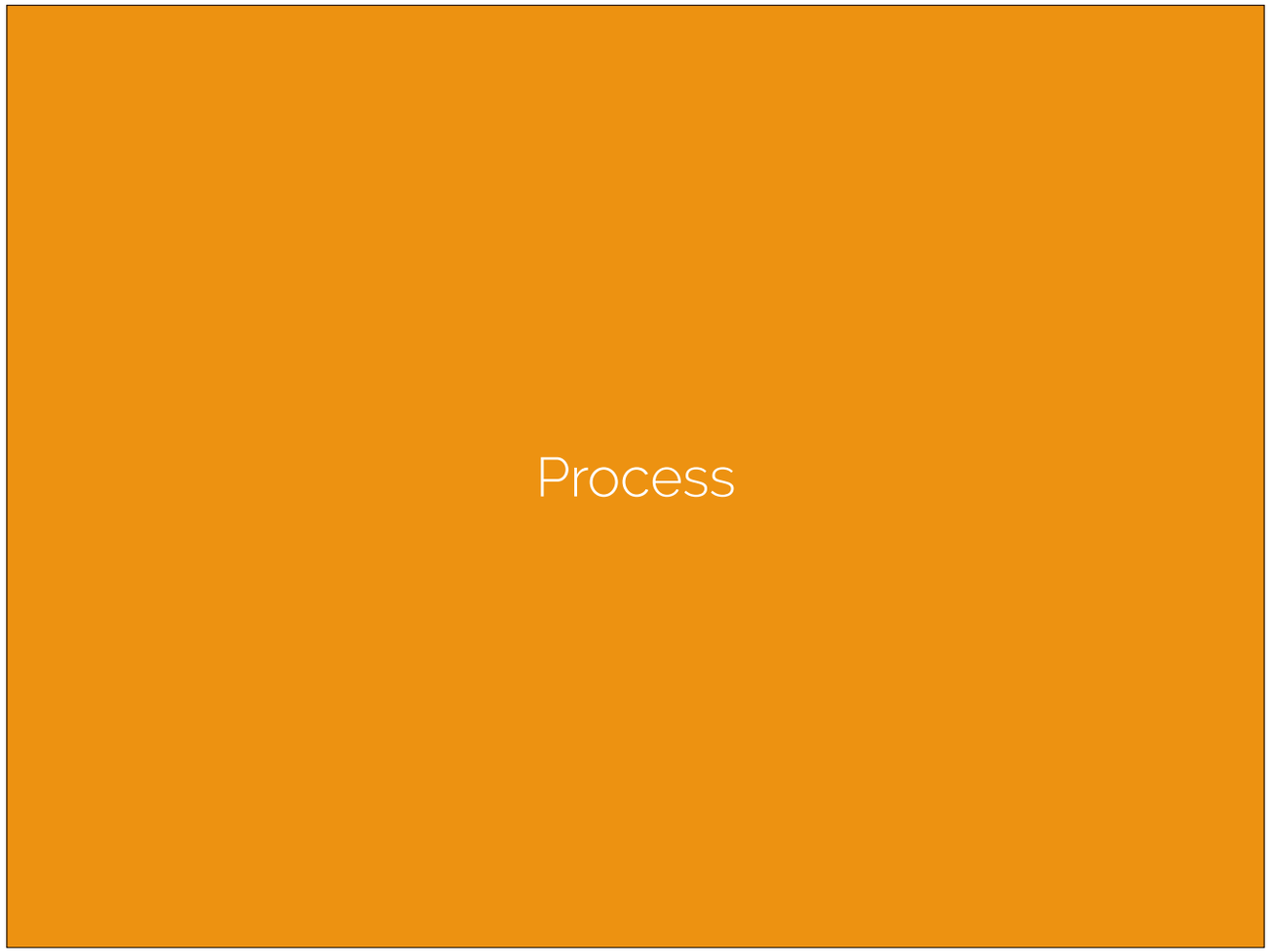
So art is a lot of things. So, here I will be talking about creating art by coding.



Using textual programming languages (like the language called Processing shown here)



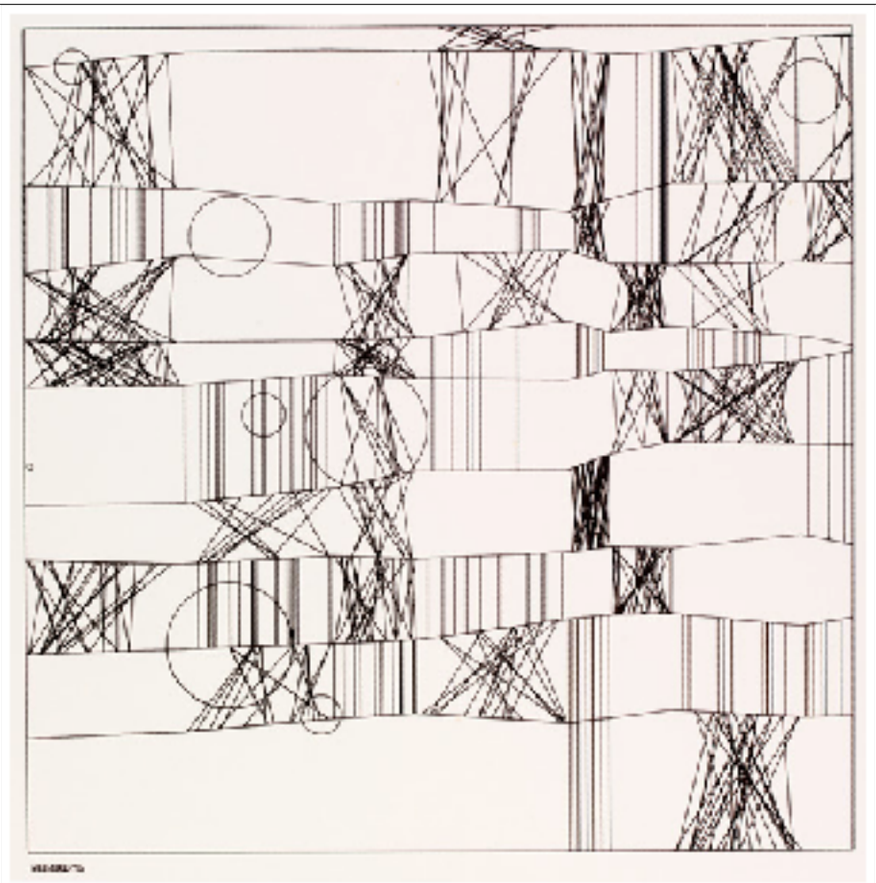
Or using a visual programming language (like the language called Nodebox shown here)
to make things. to make art.



And I will show you what a creative process looks like while coding. Ok. So how do we do this? First we start with some tools.

So just as an artist working with paints has his tools. Paints, paintbrush, palette, canvas. So does an artist working in code need tools.

http://upload.wikimedia.org/wikipedia/commons/e/e9/Brush_and_watercolours.jpg



Frieder Nake, Hommage à Paul Klee 13/3/65 Nr.2, 1965. Museum no. E.851-2008. Given by the American Friends of the V&A through the generosity of Patric Prince

So early programmed art was written in the same computer languages as software for business and science. Frieder Nake in 1965 wrote a program to send commands to a plotter, so here is one such output.

http://dada.compart-bremen.de/item/artwork/414
http://www.vam.ac.uk/content/articles/a/computer-art-history/

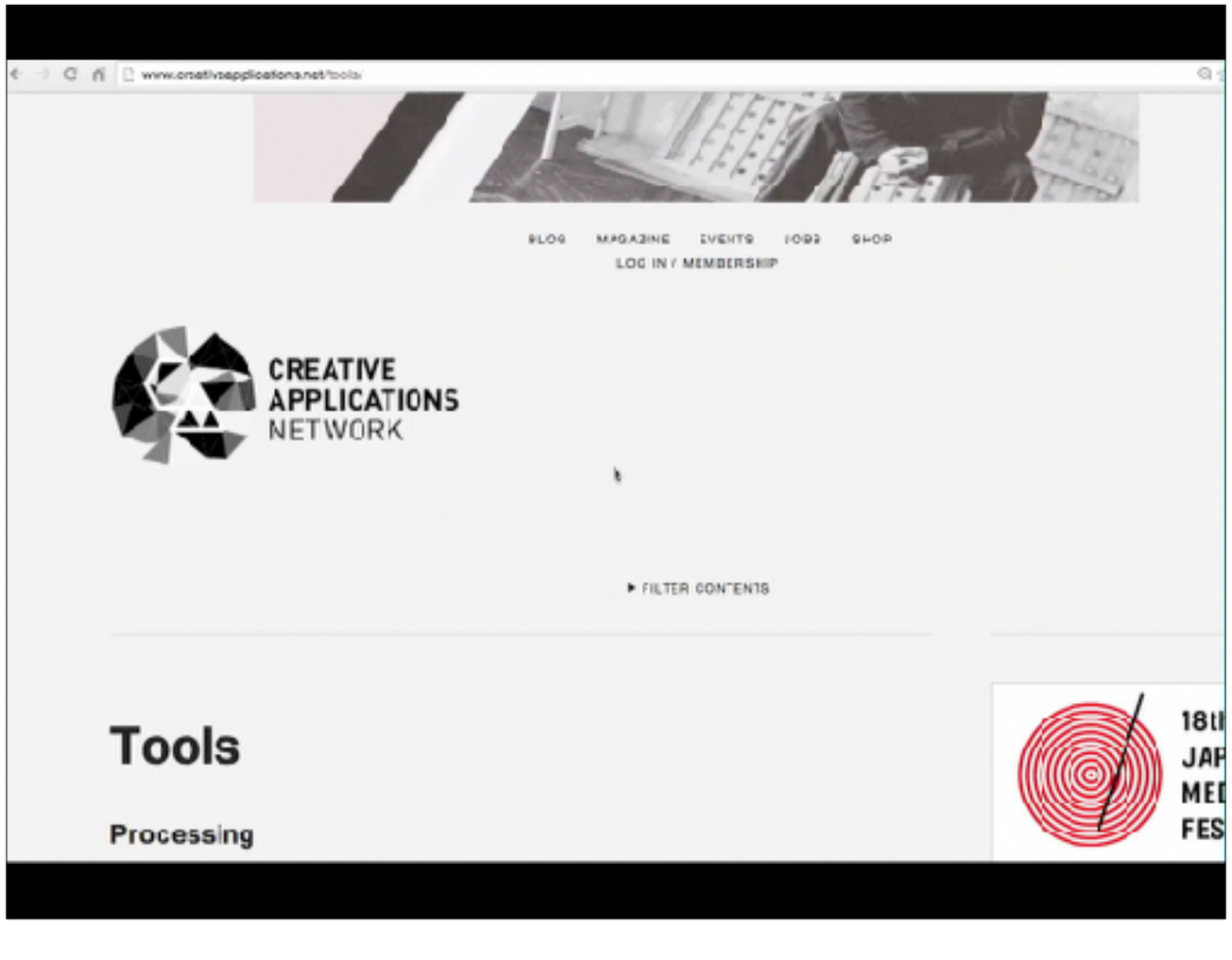
Symbolic instructions	Instructions in machine code	Storage addresses for the instructions
LS = > (1001)	1001 0 67	(0600)
0000 = > I 8	0000 8 91	(0601)
(1001) = > A	1001 0 41	(0602)
-> (1002/8) + A	1002 8 45	(0603)
I 8 + 0001	0001 8 93	(0604)
I 8 ? 0049	0049 8 98	(0605)
-- S, ≠	0603 0 14	(0606)
A = > (0610)	0610 0 42	(0607)
(0610) = > LS	0610 0 69	(0608)
Stop !	7900 0 00	(0609)
	0000 0 00	(0610)
	9999 9 99	(0611)
	9999 9 99	(0612)

So Nake wrote his program on the ER56 and his program would have looked something like this in 1965.

But for artists today we have a much easier time than digital artists did 50 years ago. Not only have program languages become easier to use, but we have languages designed especially for artists and their needs.

We state it first in symbolic language and then in the code Script of the "ER 56." In the third column we note the addresses in which we want to store each program instruction. We start by selecting an address at random; it turns out to be address (0600). We could take any other address; our computer has plenty of them.

http://www.lauftext.de/cybernetic-computer/language.htm



createapplications.net has a short list of programming tools created especially for creative pursuits.

So here we have:

Processing, Openframeworks, Cinder, vvvv (v4), MaxMSP, Pure Data, SuperCollider, QuartzComposer, NodeBox, Polycode, Three.js, Processing.js, Paper.js, d3 and Raphael

http://www.creativeapplications.net/tools/

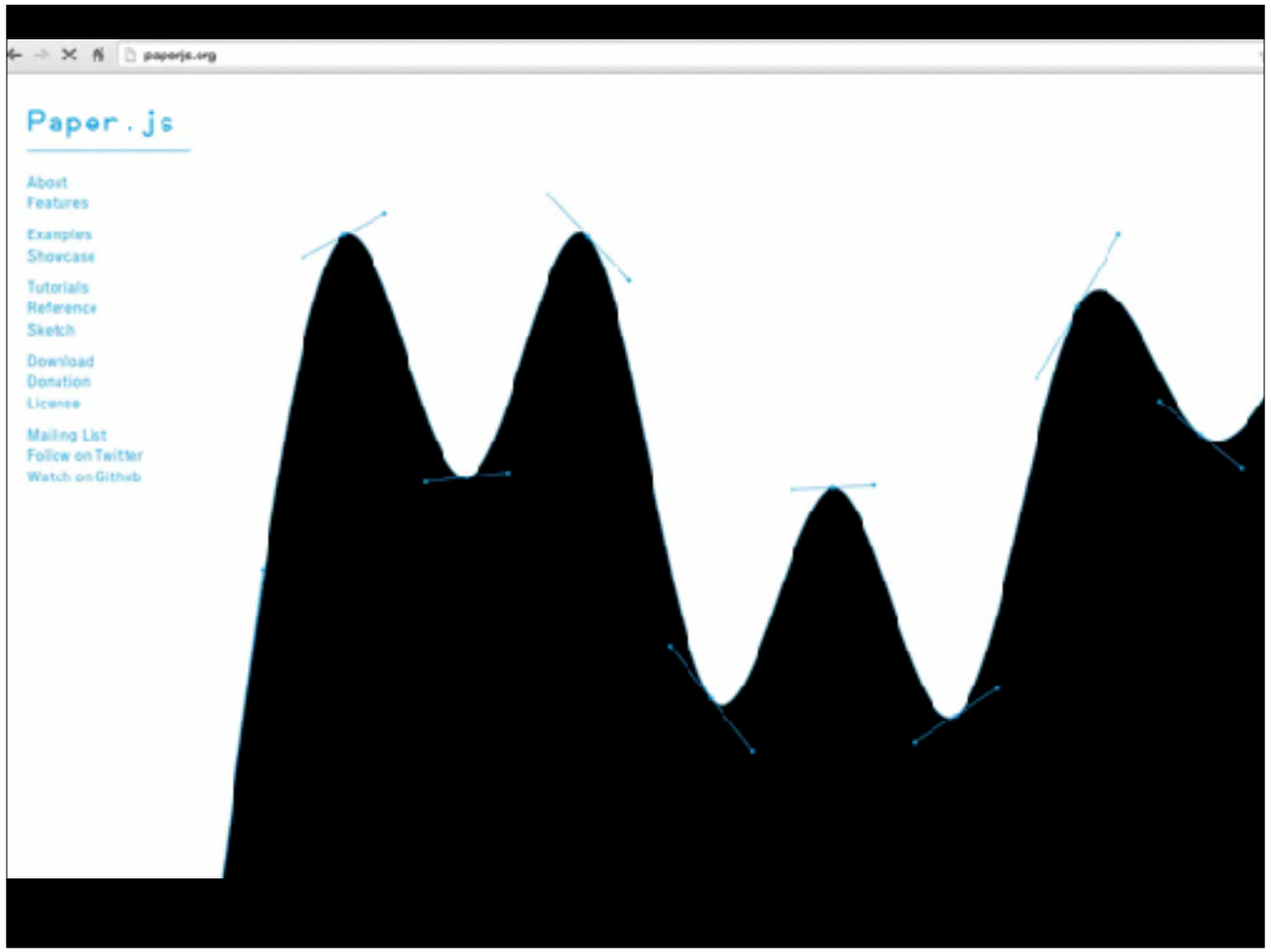
Software for Creative Programming

Processing / Processing.js / P5JS
Openframeworks
Scratch
d3 / raphael
vvvv (v4), MaxMSP, Pure Data, SuperCollider
Cinder, QuartzComposer
Paper.js
Polycode, Three.js

resources:
http://hackingforartists.com/
http://www.creativeapplications.net/tools/

So each of these were created to make creative programming easier. But that is a large field, as we saw Art is a lot of things. So each system here is created by a programmer who has specific needs and desires, and each system here makes some things easier to accomplish and other things harder.

So I think the best way to get an idea for what a system is good for is to read the about page and look at their gallery of examples.



So here we have Paper.js

about: Paper.js is an open source vector graphics scripting framework that runs on top of the HTML5 Canvas. It offers a clean Scene Graph / Document Object Model and a lot of powerful functionality to create and work with vector graphics and bezier curves, all neatly wrapped up in a well designed, consistent and clean programming interface.

So lets look at the gallery

Paper.js

Showcase

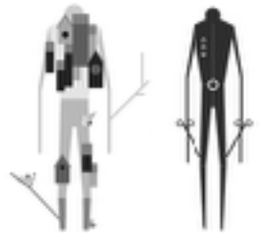
About
Features

Examples
Showcase

Tutorials
Reference
Sketch

Download
Donation
License

Mailing List
Follow on Twitter
Watch on Github



Les métamorphoses de Nr. Kalia

This Google DevArt winner combines Paper.js with body tracking in wonderful ways.

NY Times: You Made That

The New York Times turned to Paper.js for this fun abstract art tool.

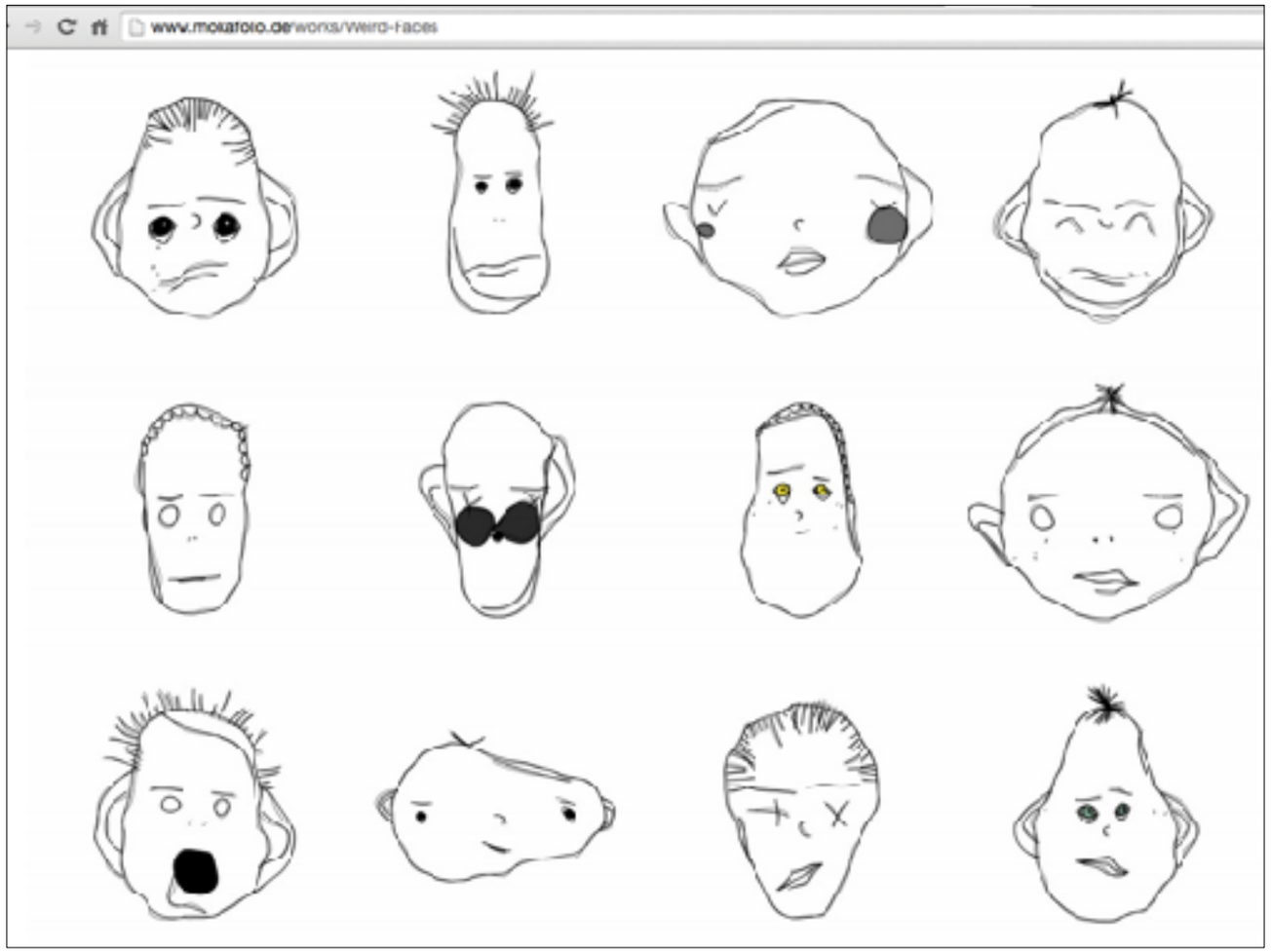
Foursquare's Timemachine

Four Square uses Paper.js to allow their users to visually relive their check-in history.

Google Chrome Racer

Turbulence

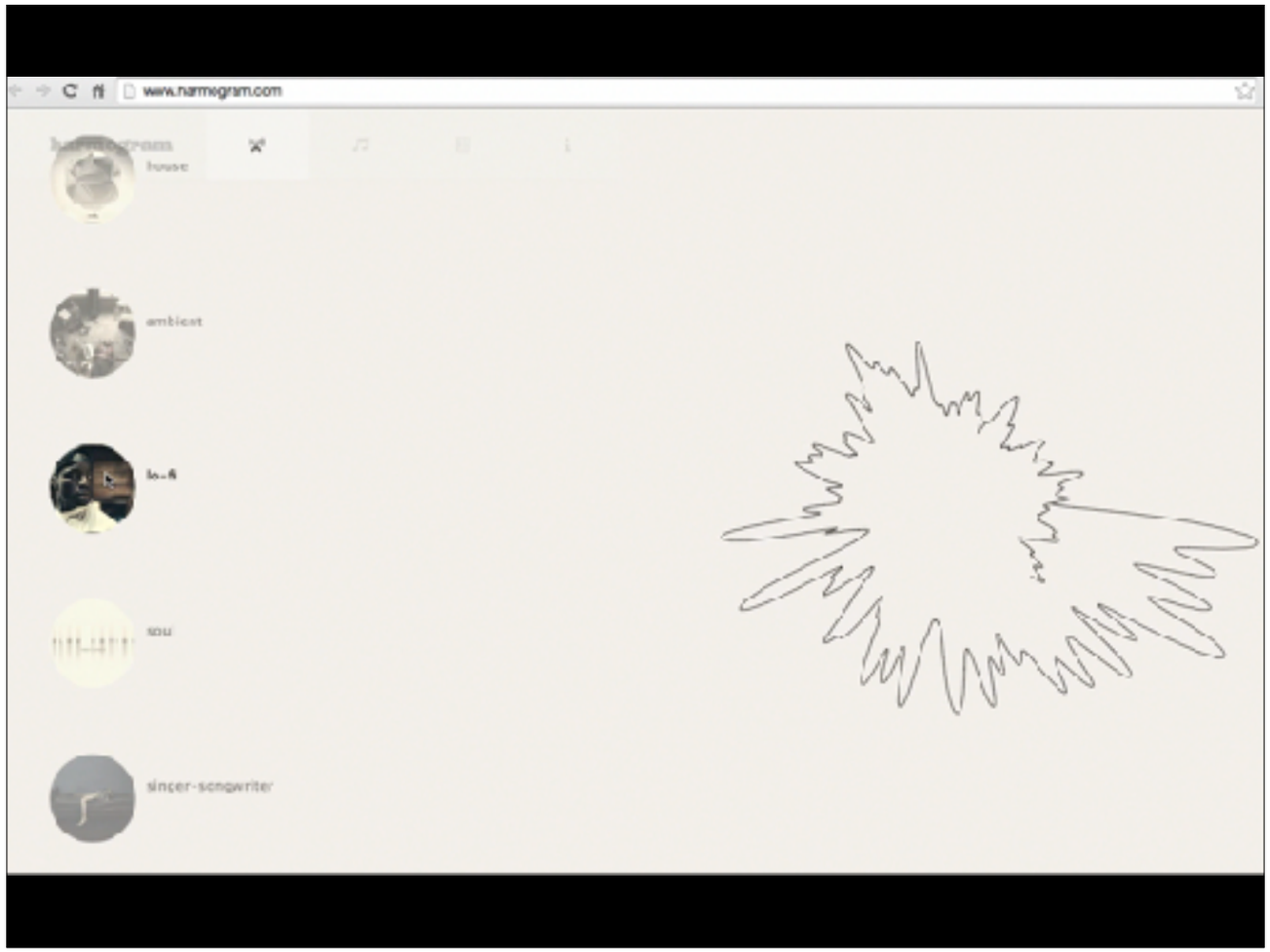
Kolber



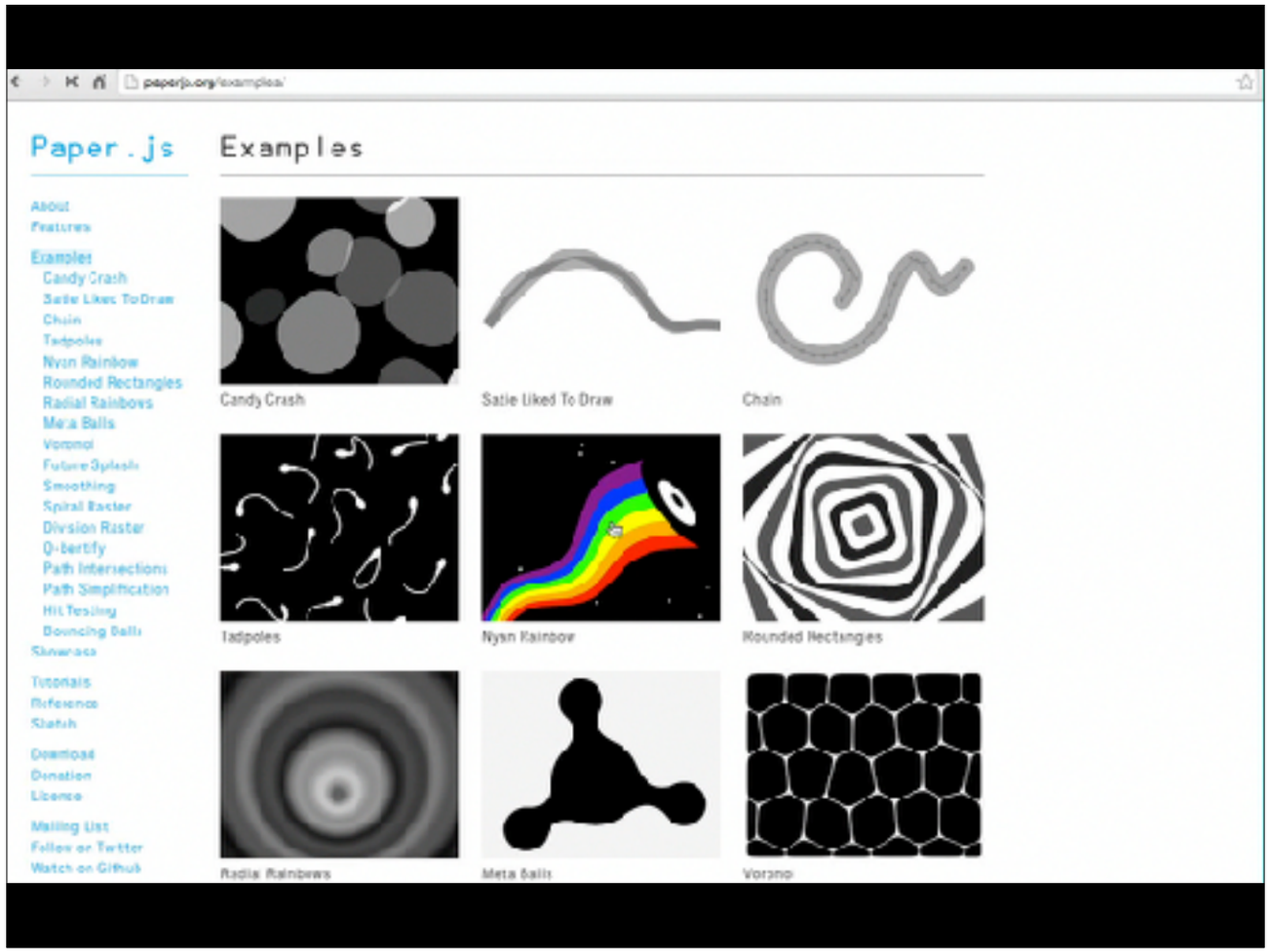
Weird faces by Matthias Dorfelt created in paper.js

http://www.creativeapplications.net/javascript-2/weird-faces-study-by-matthias-dorfelt-using-paperjs/

http://www.mokafolio.de/#!project=21



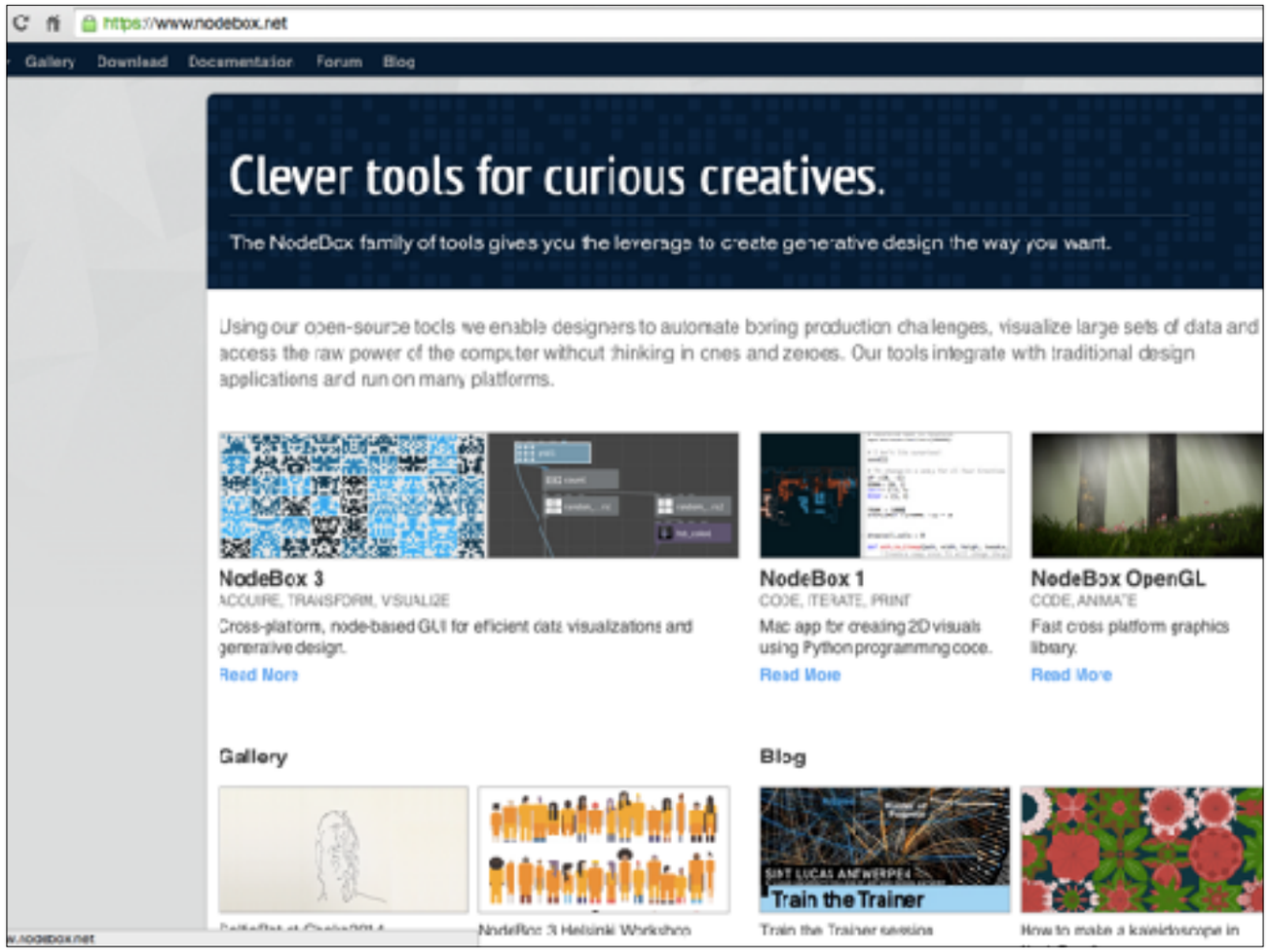
http://www.harmogram.com/ made in paper.js



Nyan rainbow

http://paperjs.org/examples/nyan-rainbow/

inspired by http://en.wikipedia.org/wiki/Nyan_Cat I suppose.

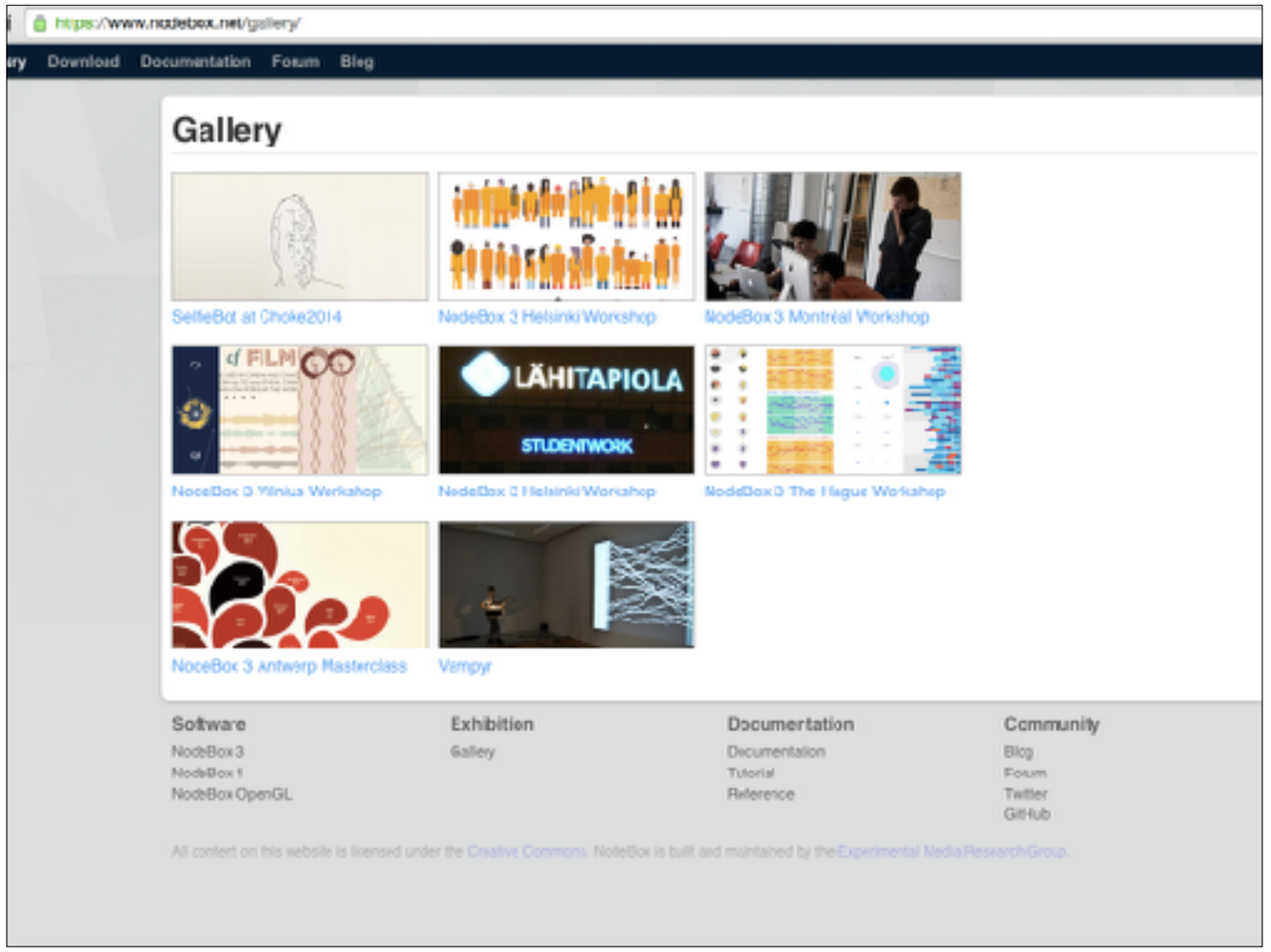


OK. Here is another. This one is called Nodebox.

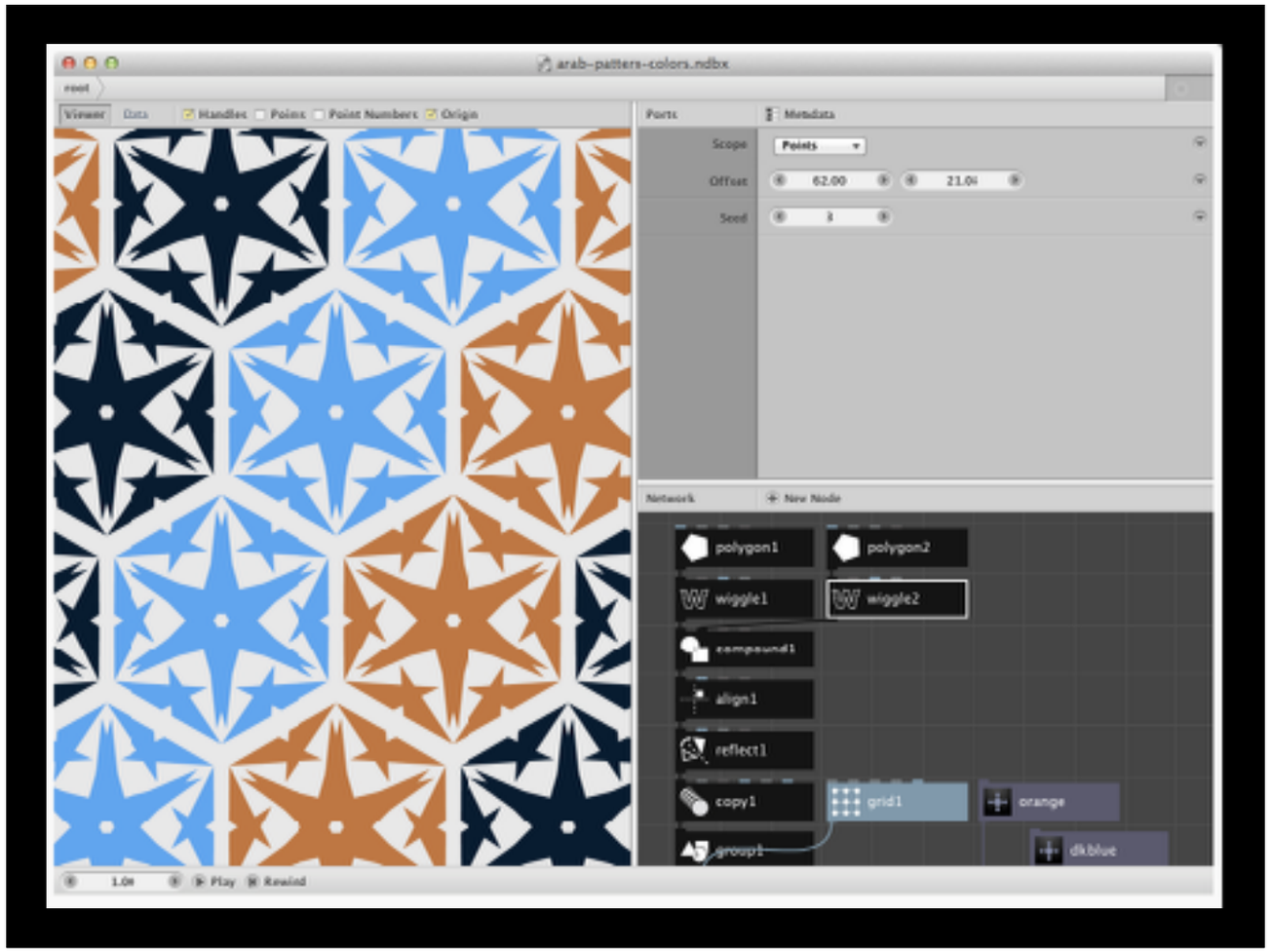
NodeBox makes it easy to do data visualisations, generative design and complex production challenges.

NodeBox is a node-based software application for generative design. It's built from the ground up by designers to be easy-to-use, efficient, and fast.

https://www.nodebox.net/node/

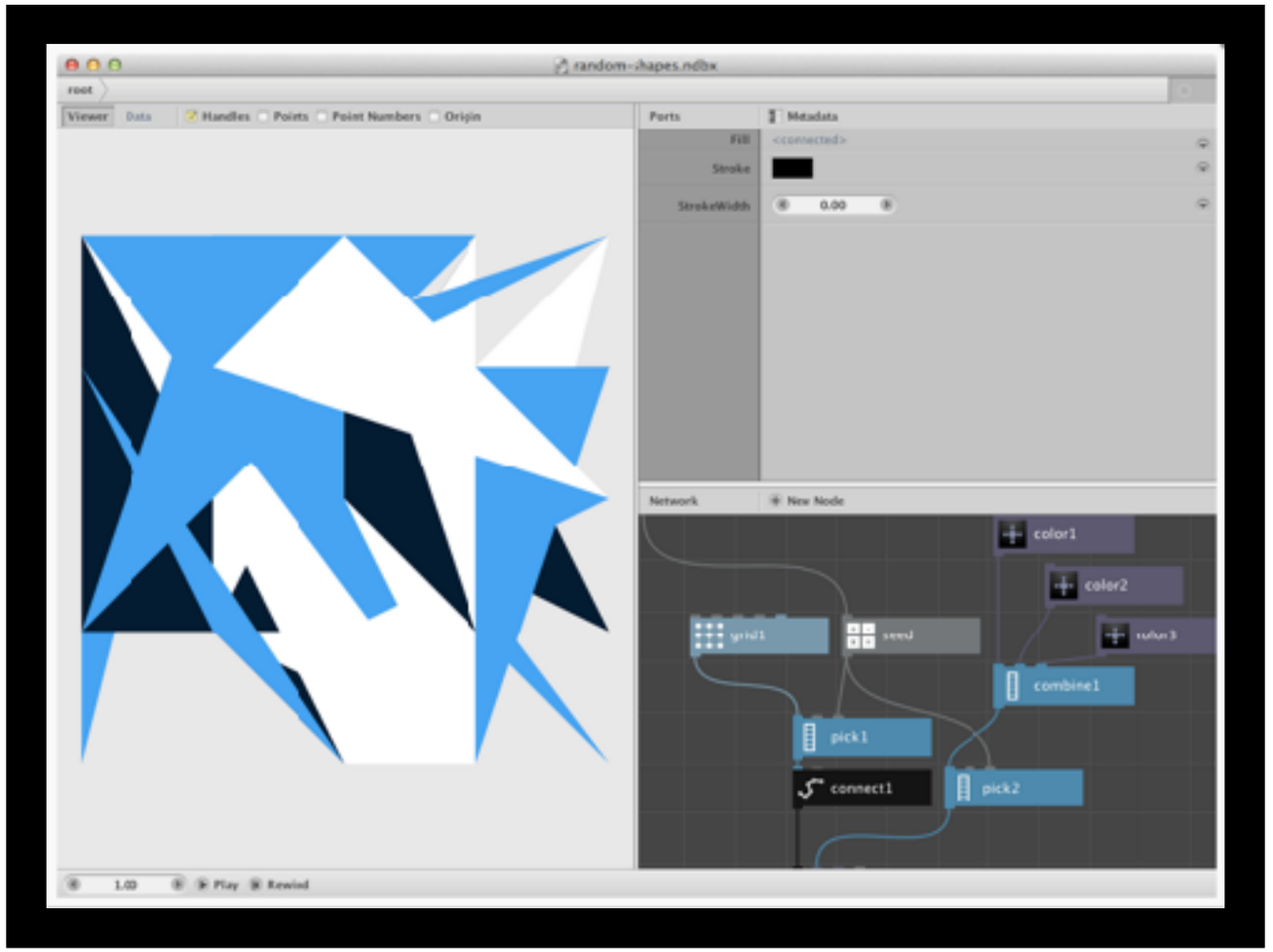


Nodebox gallerypage



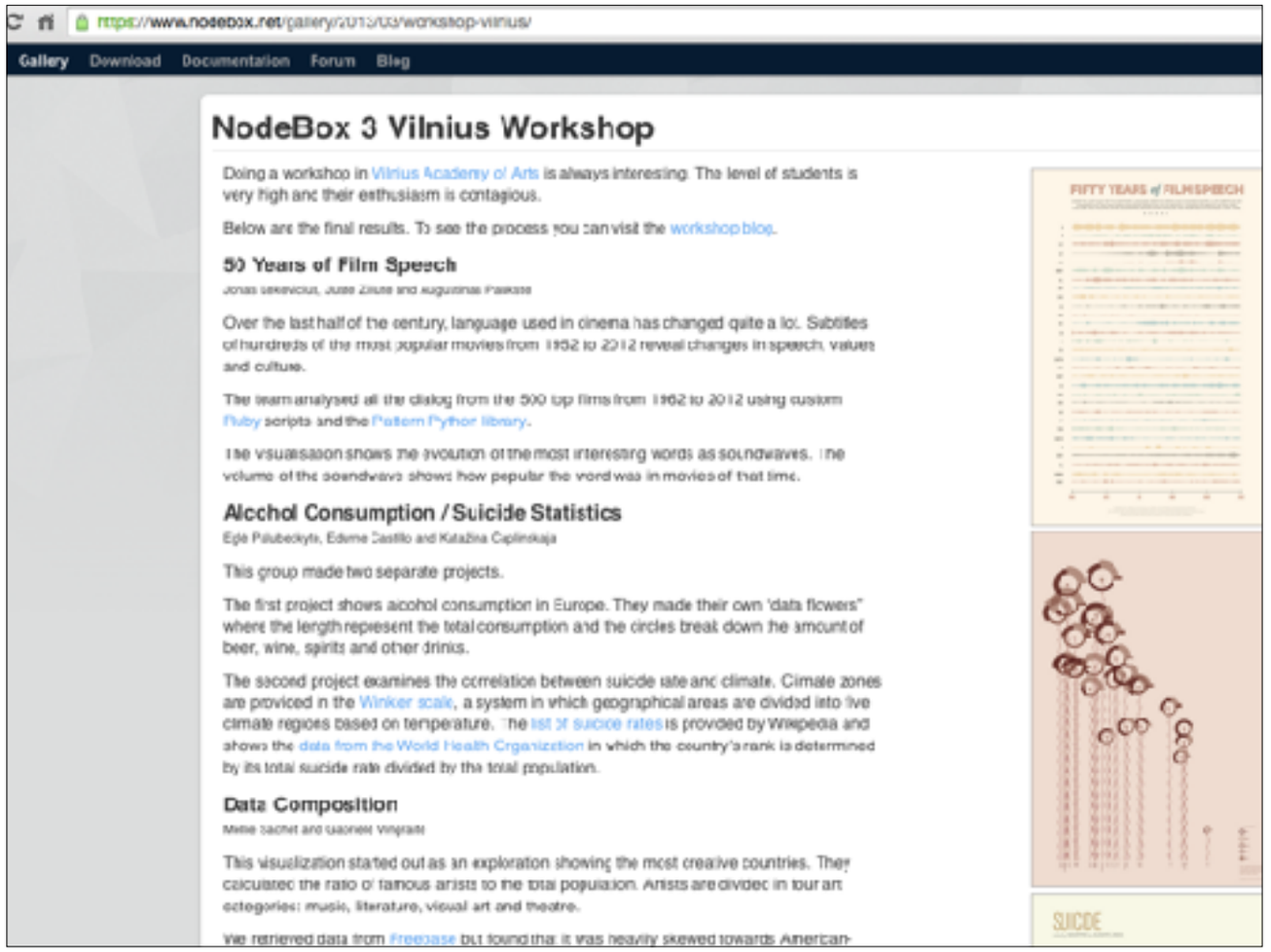
Repeating patterns

from https://www.nodebox.net/node/



Abstractions

from https://www.nodebox.net/node/



https://www.nodebox.net/gallery/2013/03/workshop-vilnius/

NodeBox 3 Vilnius Workshop
Doing a workshop in Vilnius Academy of Arts is always interesting. The level of students is very high and their enthusiasm is contagious.

Below are the final results. To see the process you can visit the workshop blog.

Summary

We believe NodeBox is a useful and accessible tool for data visualization. The results of the workshop, done in one week, show that students are able to create quality work using new tools with the right amount of creativity, a bit of guidance and a lot of hard work.

In addition to being useful for students, NodeBox workshops are a useful instrument for research as we examine how new users learn and use generative software. As always, this feedback guides our direction of the project. This was also the first workshop where we used Grasp, our in-house tool for usability research. Results of this will be published soon.

Over the last half of the century, language used in cinema has changed quite a lot. Subtitles of hundreds of the most popular movies from 1952 to 2012 reveal changes in speech, values and culture.

The team analysed all the dialog from the 500 top films from 1962 to 2012 using custom Ruby scripts and the Pattern Python library.

The visualisation shows the evolution of the most interesting words as soundwaves. The volume of the soundwave shows how popular the word was in movies of that time.

Alcohol Consumption / Suicide Statistics

Eglė Palubeckytė, Edurne Castillo and Katažina Čaplinskaja
This group made two separate projects.

The first project shows alcohol consumption in Europe. They made their own “data flowers” where the length represent the total consumption and the circles break down the amount of beer, wine, spirits and other drinks.

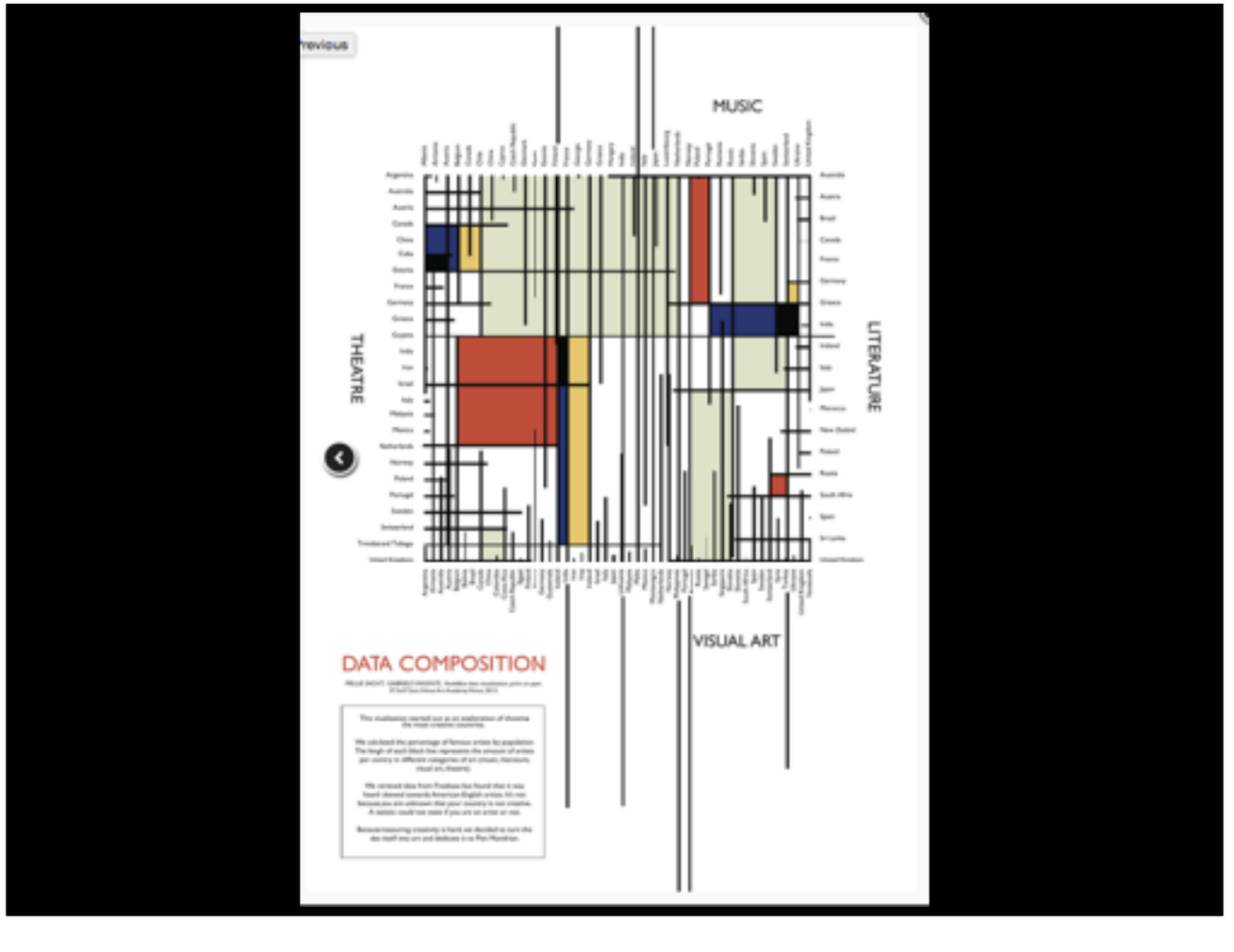
The second project examines the correlation between suicide rate and climate. Climate zones are provided in the Winkler scale, a system in which geographical areas are divided into five climate regions based on temperature. The list of suicide rates is provided by Wikipedia and shows the data from the World Health Organization in which the country’s rank is determined by its total suicide rate divided by the total population.

Alcohol Consumption / Suicide Statistics

Eglė Palubeckytė, Edurne Castillo and Katažina Čaplinskaja
This group made two separate projects.

The first project shows alcohol consumption in Europe. They made their own “data flowers” where the length represent the total consumption and the circles break down the amount of beer, wine, spirits and other drinks.

The second project examines the correlation between suicide rate and climate. Climate zones are provided in the Winkler scale, a system in which geographical areas are divided into five climate regions based on temperature. The list of suicide rates is provided by Wikipedia and shows the data from the World Health Organization in which the country’s rank is determined by its total suicide rate divided by the total population.



Data Composition

Mellie Sachet and Gabrielė Vingraitė

This visualization started out as an exploration showing the most creative countries. They calculated the ratio of famous artists to the total population. Artists are divided in four art categories: music, literature, visual art and theatre.

We retrieved data from Freebase but found that it was heavily skewed towards American-English artists and not representative.

Because it was so hard to measure, the project shifted towards turning the data itself into art, dedicated to Piet Mondrian.

here is an actual Mondrian

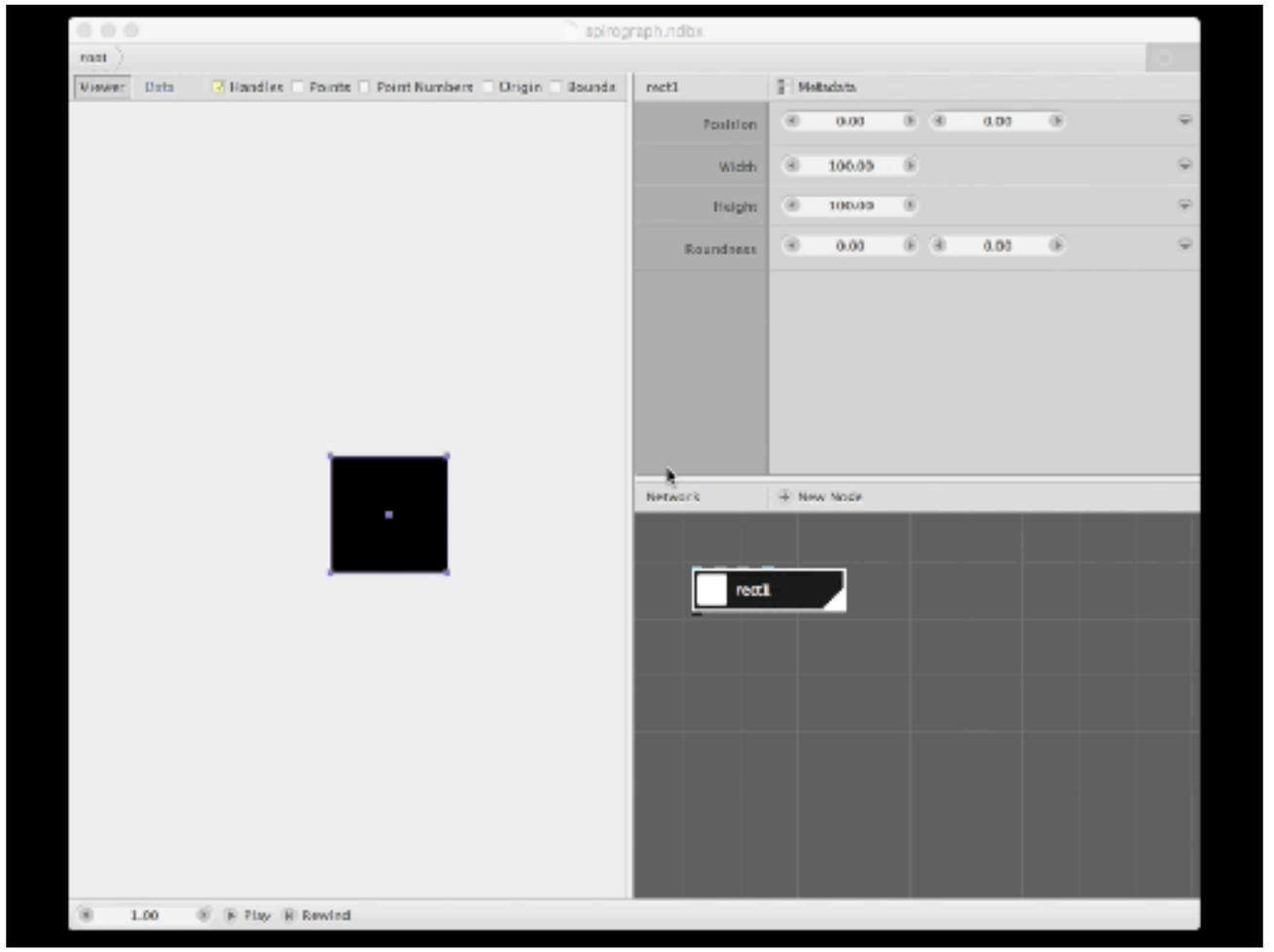
https://www.google.com/culturalinstitute/asset-viewer/tableau-yellow-black-blue-red-and-grey/GwHz_0nmyg4GGw?projectId=art-project

Eurovoices

Dalia Kemeklytė and Viktorija Pampuščenko
The Eurovision Song Contest would not be the Eurovision Song Contest without its predictable voting process. Rather than voting for the best songs, neighbouring or politically sympathetic countries vote for each other as a block.

The group decided to explore the relationships between the countries by revealing the sum of points they gave to each other throughout the period of 1975-2003.

Each country is represented as a planet with its satellites (other countries) – the bigger the satellite, the stronger their relationship.



And here is what it looks like to actual code in Nodebox.

Here is me creating a spirograph (following the Nodebox tutorial online actually).

About 8min. sped up to just a few.

https://processing.org/exhibition/

Processing | p5.js | Processing.py

Processing 2

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books
Handbook

Overview
People
Foundation

Shop

- Forum
- GitHub
- Issues
- Wiki
- FAQ
- Twitter
- Facebook

Exhibition. A curated collection of projects created with Processing. New software added each month.

Curated by Filip Visnjic of CreativeApplications.net

Non-Linear Code
by Dextro

Dextro writes 'non linear code' drawing inspiration from nature. The results are non fractal or random programs that iterate without change, with equal rules for all objects. Most of the scripts rely on trigonometry and could be seen as sets of wave generators interacting with one another. Some of these pieces take years to develop but the code is usually short but complex.

Links: Dextro, Vimeo

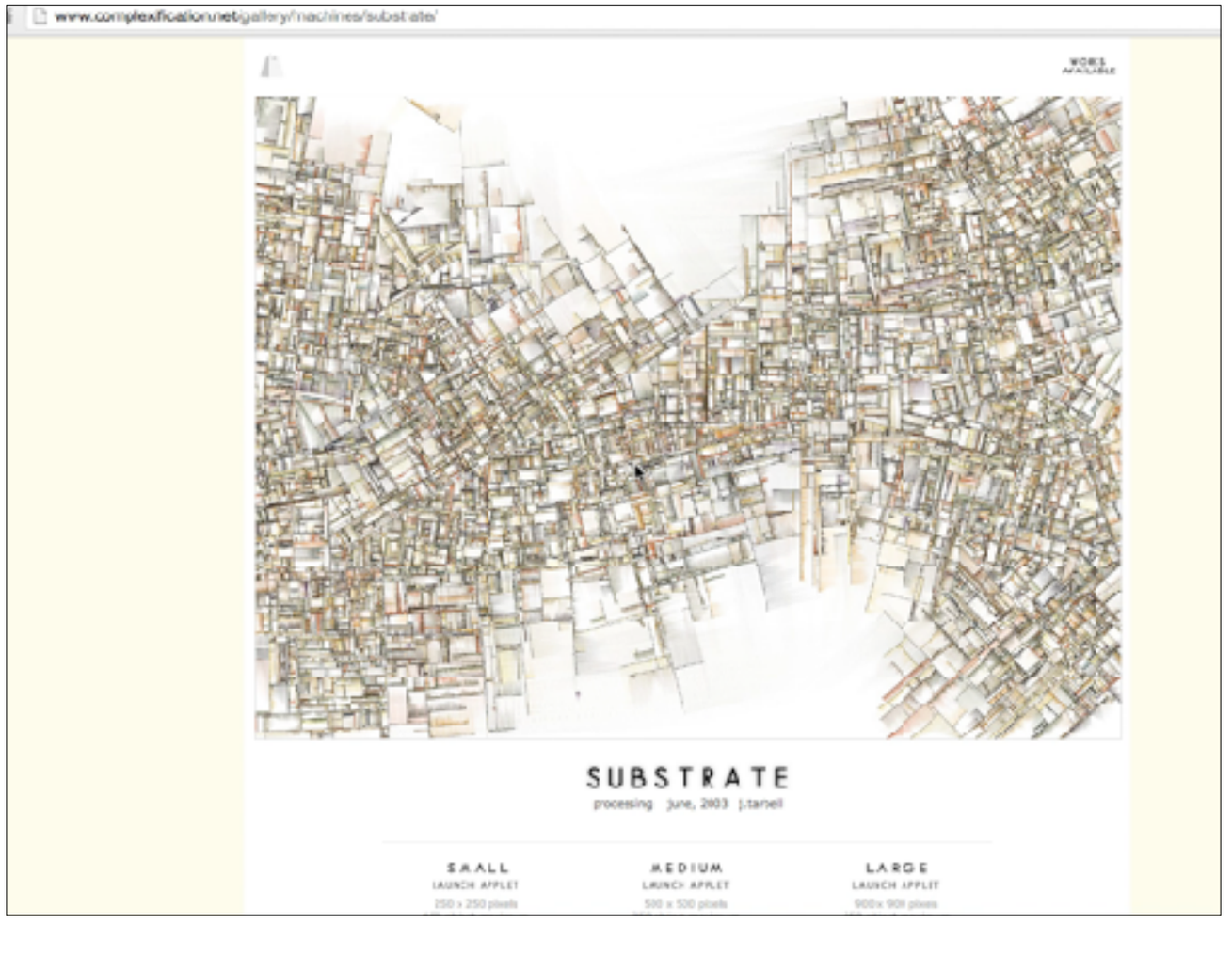
Computational design methodologies for large-scale 3D printing
by GAD – RC4

With an exponential increase in the possibilities of computation and computer-controlled fabrication, architecture is now facing a novel challenge Bartlett School of Architecture's RC4 in London researches computational design methodologies for large-scale 3D printing with industrial robots, taking logistical, structural and material constraints as design opportunities.

Filament Sculptures
by Lia

For about a year now generative artist lia hasbeen exploring 3d printing by analysing filament and the movements of the printhead. Ratherthan just having 3d models printed out, Lia hasbeen interested in the possibilities of the process by defining the location of the printhead, the speed of the movement and the amount of filament that should be extruded.

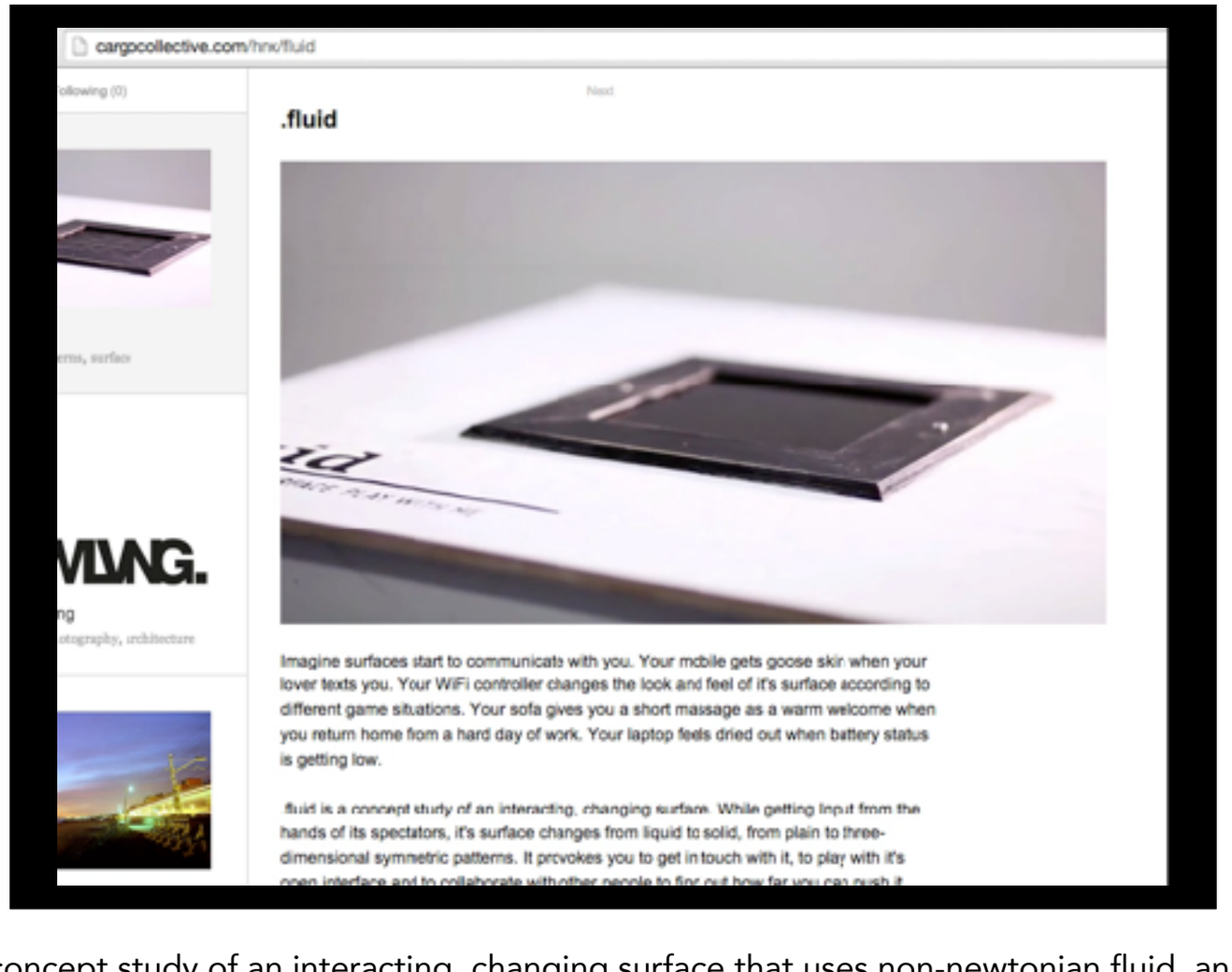
Links: Lia, liasomething.tumblr.com



http://www.complexification.net/gallery/machines/substrate/

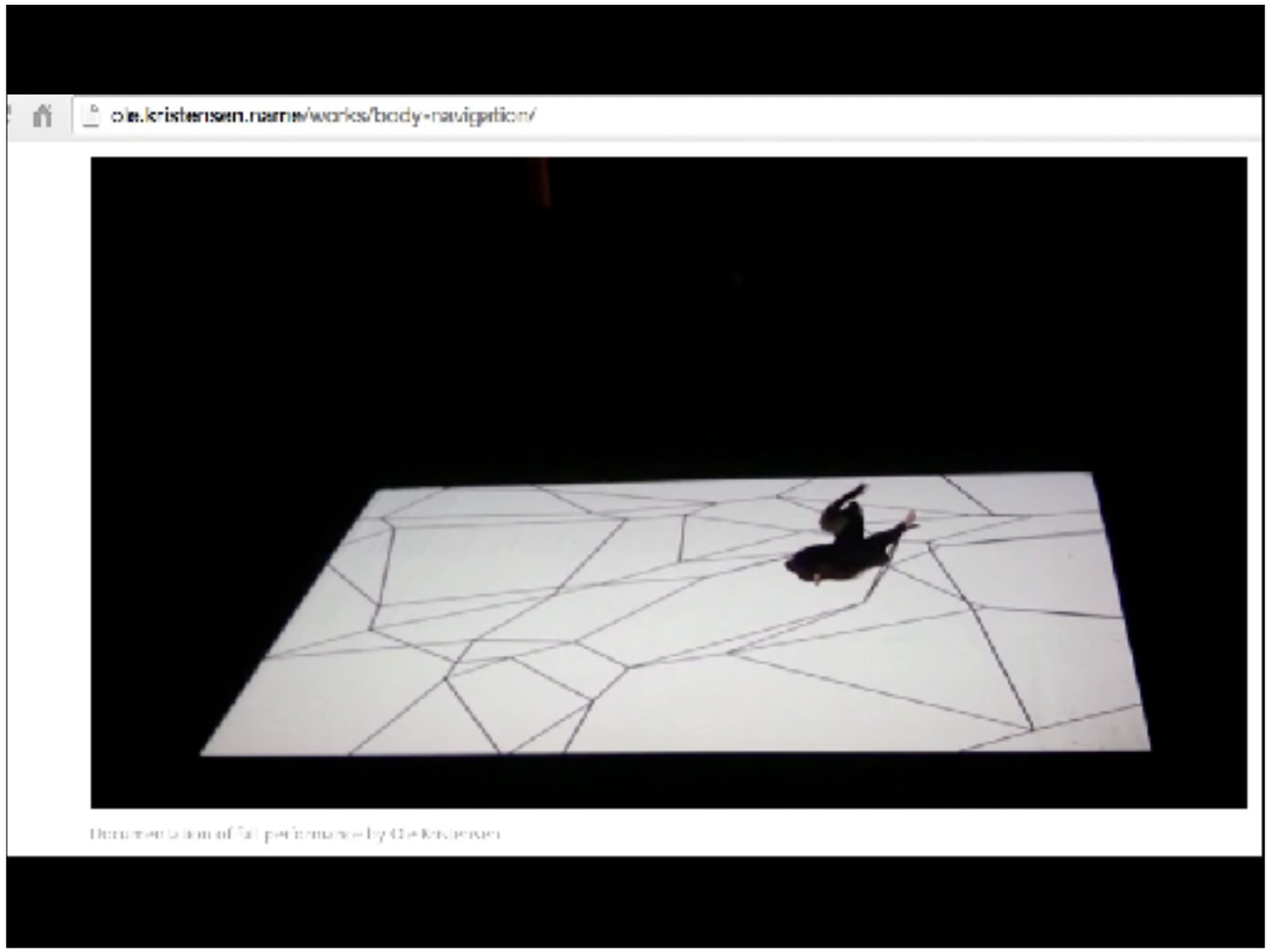
from “Processing” the book

Substrate beings similarly with an empty rectangular region. It has been compared to crystal formation and the emergent patterns of urban landscapes. A single line (known internally as a “crack” since the algorithm was inspired by sunbaked mud cracks) begins drawing itself from some random point in some random direction. The line continues to draw itself until it either (a) hits the edge of the screen of (b) hits another line, at which point it stops and two more lines begin. The one simple rule used in the creation of new lines is that they begin tangents to existing lines. This process is repeated until there are too many lines to keep track of or the program is stopped.



Created by Hannes Jung, .fluid is a concept study of an interacting, changing surface that uses non-newtonian fluid, an Arduino board, a speaker and Processing to allow surface to change from liquid to solid, from plain to three-dimensional symmetric patterns

https://processing.org/exhibition/
http://cargocollective.com/hnx/fluid

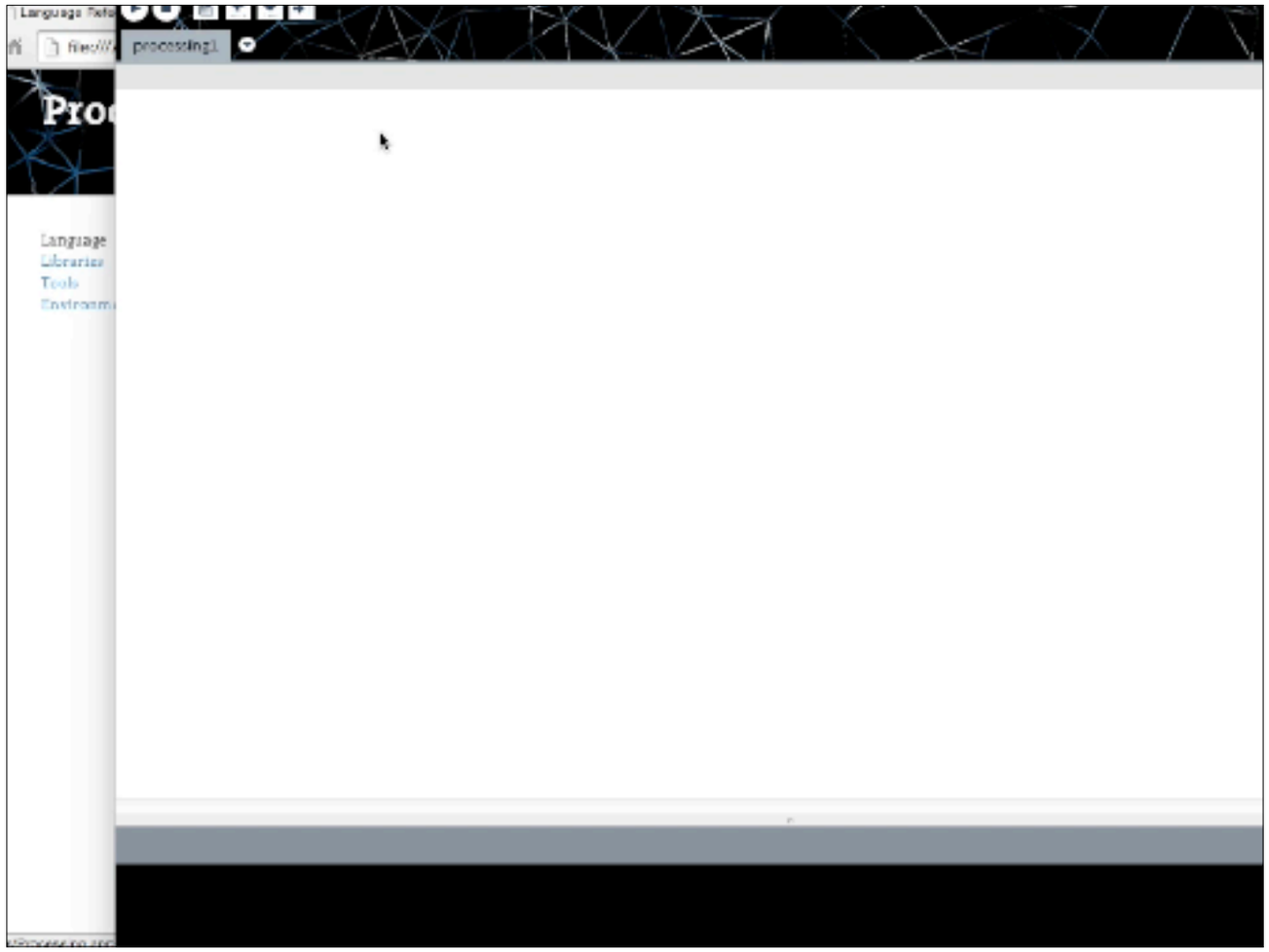


Body Navigation by Jonas Jongejan and Ole Kristensen for Recoil Performance Group

Infrared motion tracking and floor projection installation for a Danish Dance Theatre production with choreographer Tina Tarpgaard.

We used processing for the infrared blobtracking of the dancers and drawing the open gl graphics. During the performance Tina controlled the whole thing live from an Isadora-based interface via osc.

http://ole.kristensen.name/works/body-navigation/

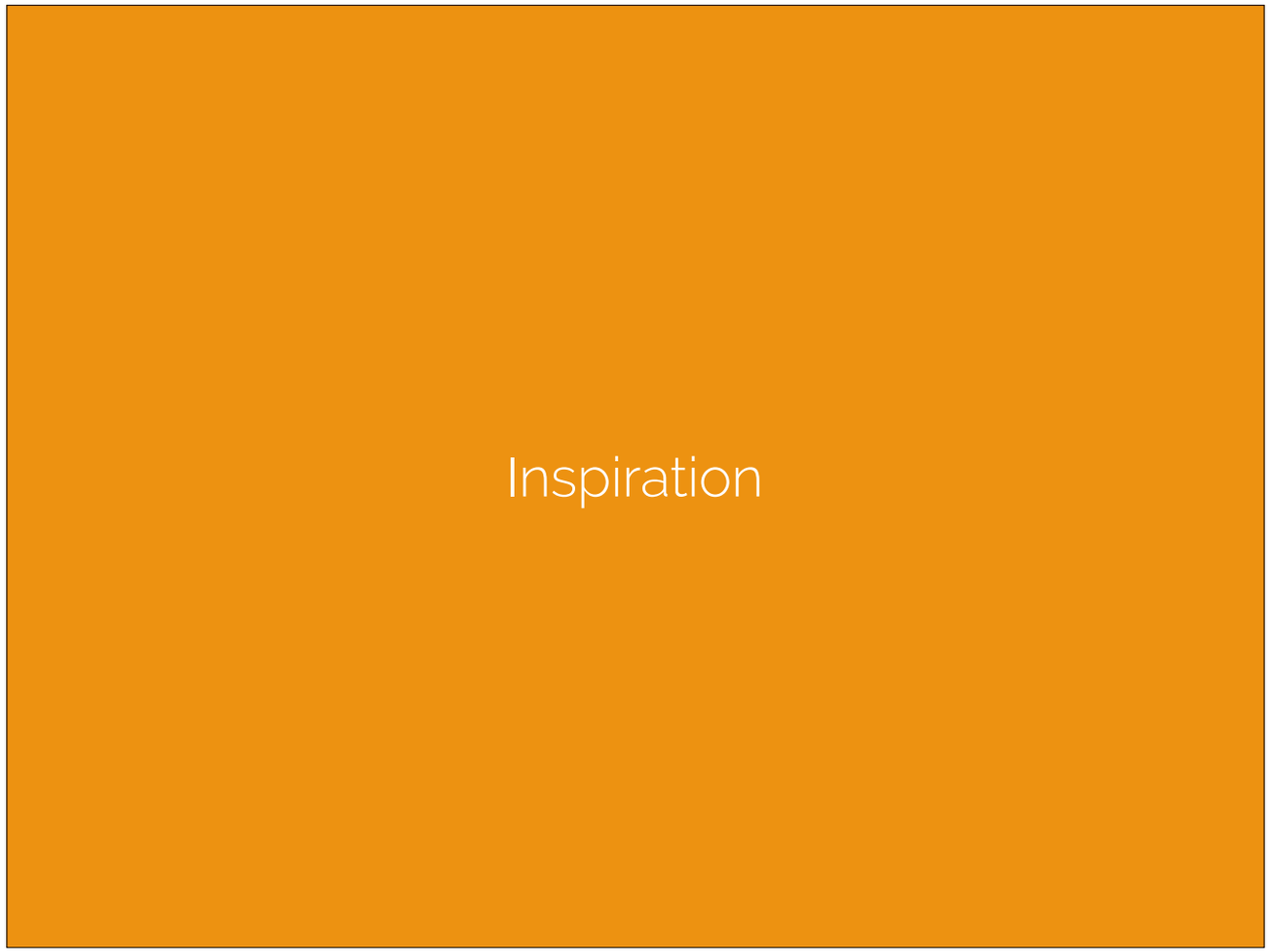


This is me coding in Processing for about 35mins., very undirected, just playing around. Here we have the video sped up 10x.

Experimentation, play, that is one way to create art. Inspiration comes from the medium.

Inspiration can also come up-front before starting a project.

Circles in Processing by Andrew R.



Inspiration can also come up-front before starting a project.

animal advent
ala ed emberley

I grew up with Ed Emberley's Drawing Book of Animals. it's a terrific book that shows kids and grownups how to construct animals step by step out of letters and simple shapes they already know how to draw. (Years later, I realized that the visual style I developed for my "virtual toys" employs the same core ideas, drawing things with rectangles, triangles, circles, and simple curves instead of pre-canned graphics. This book must have been a huge influence!)

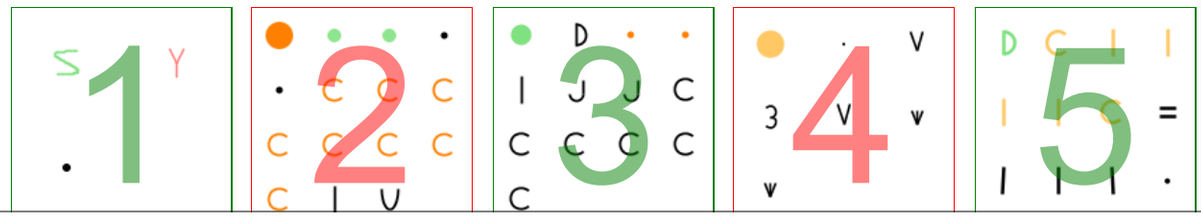
Buy It On Amazon—
It's really good!

This year, I have made an Online Advent Calendar with 25 "Animal Puppets". **For every day in December up until Christmas, a new puppet is unlocked!** Each shows off its construction from shapes and symbols and each is a little puppet you can operate with the mouse movement and button. (Just like that bird up there will "shake a tailfather" if you pass the mouse over it.)

The animal puppets start simple but get complex, so come back every day for a new virtual toy! And if you would like to learn about how I made these and how you can make your own toys, check out the GitHub project page. You can also see the Director's Commentary I made on my development blog.

If you'd like to learn how to draw these animals, and many more, buy the book! It's terrific for kids and grownups to lean on their own or together.

Happy Holidays!
Kirk <kirk@kirk.is>

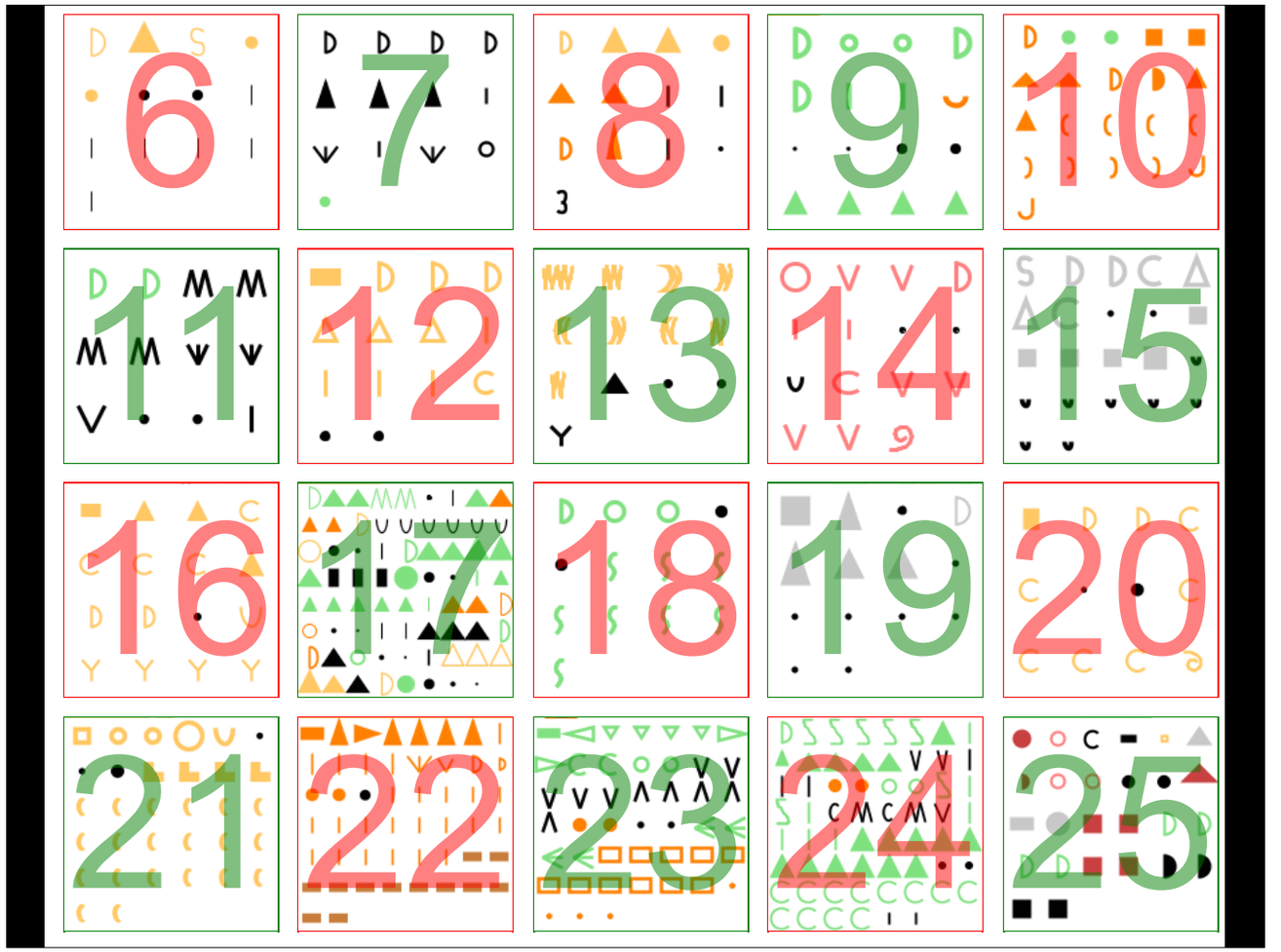


We already saw Kirk Israel’s Animal Advent written in Processing
http://advent.kirk.is/

Kirk Israel to boston-creative-coders
Show more
12/3/14
My latest bit of "tech poetry"----

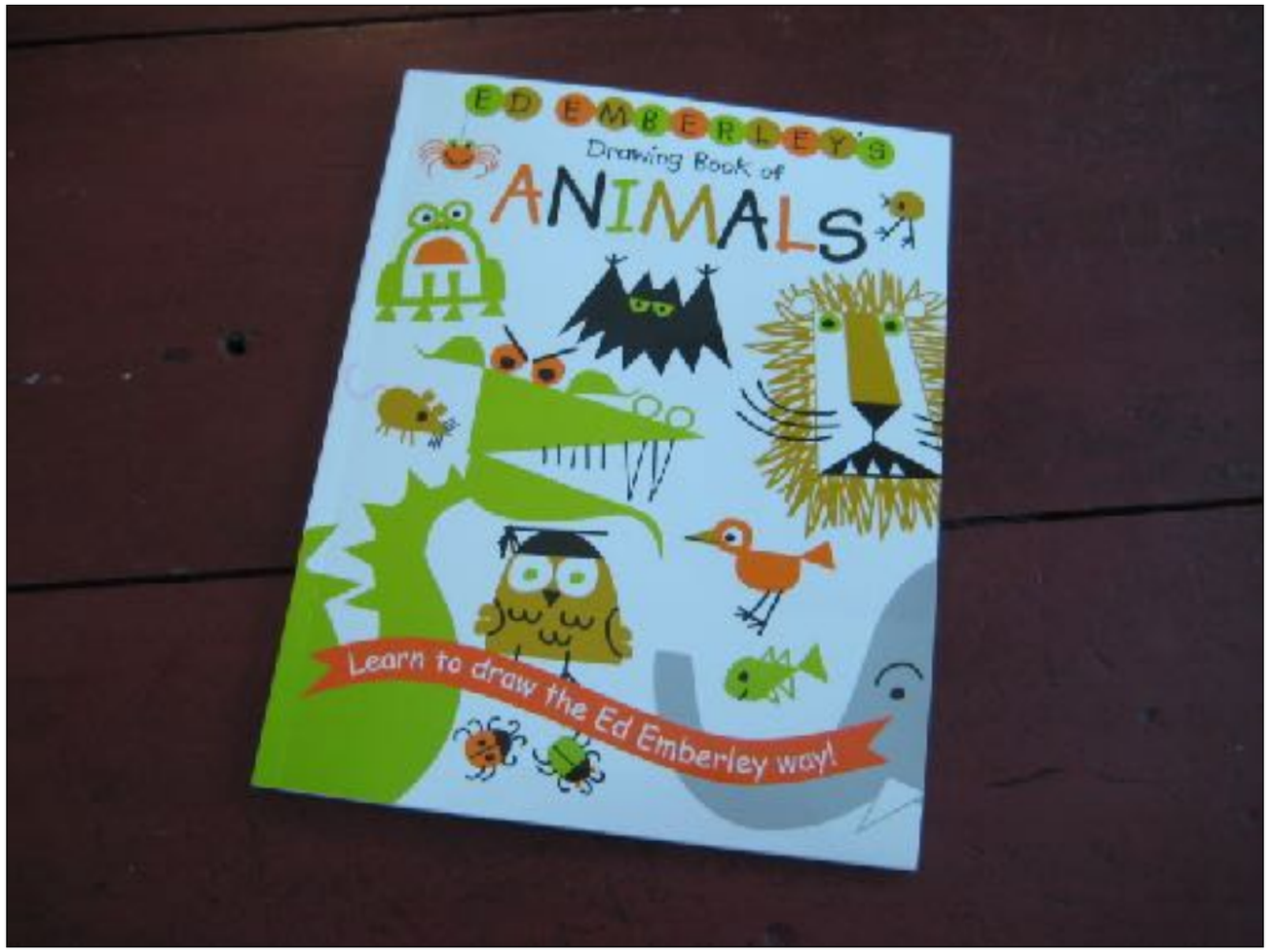
Growing up, "Ed Emberley's Drawing Book of Animals" was very important to me and my dad. It teaches kids that they can make animals of all types just by drawing letters and symbols that they already know how to make.

This year, I decided to make an Advent Calendar for the Days of December, 25 in all. Each day has an animal that Ed Emberley taught me



We already saw Kirk Israel's Animal Advent written in Processing

http://advent.kirk.is/

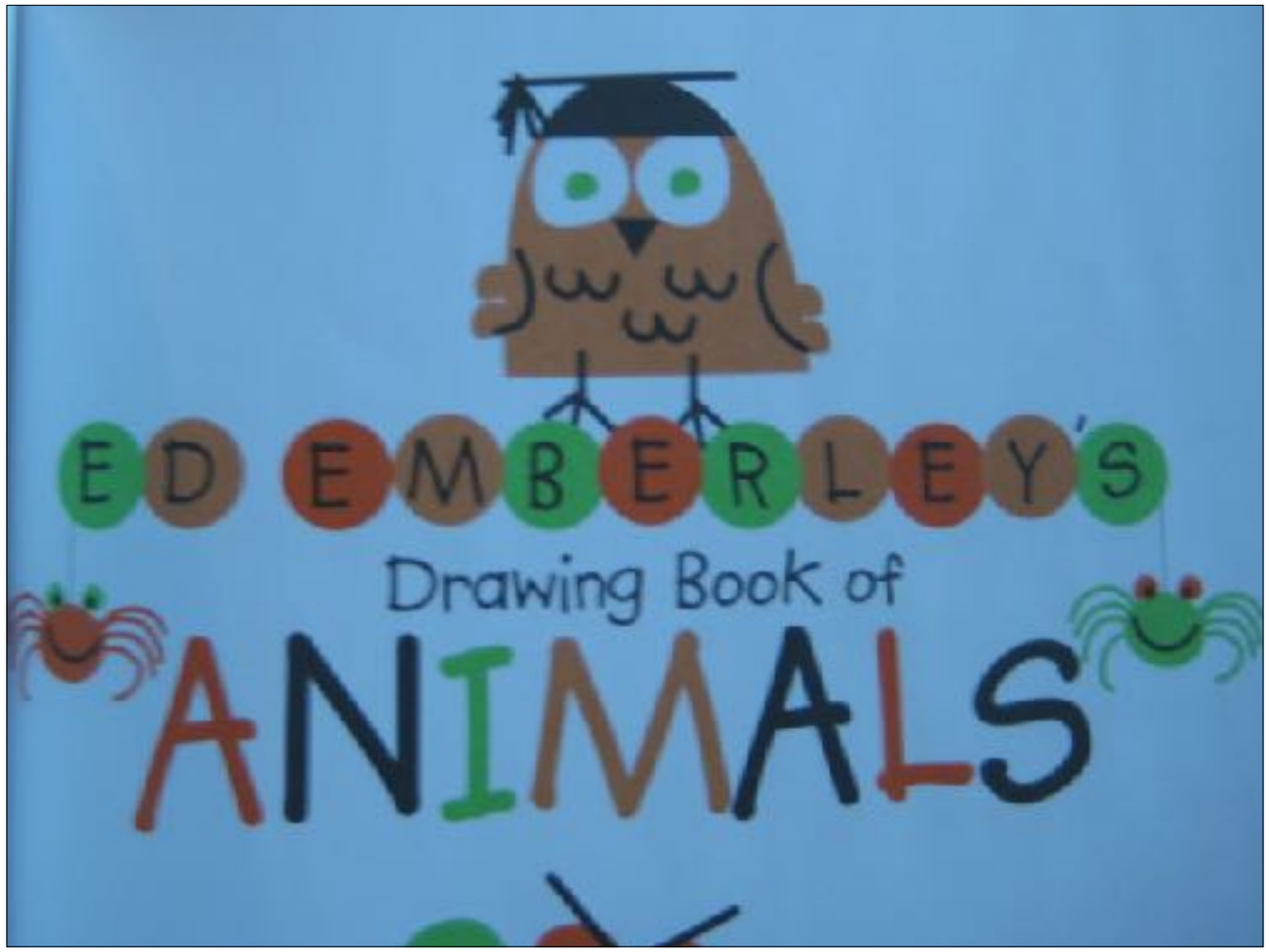


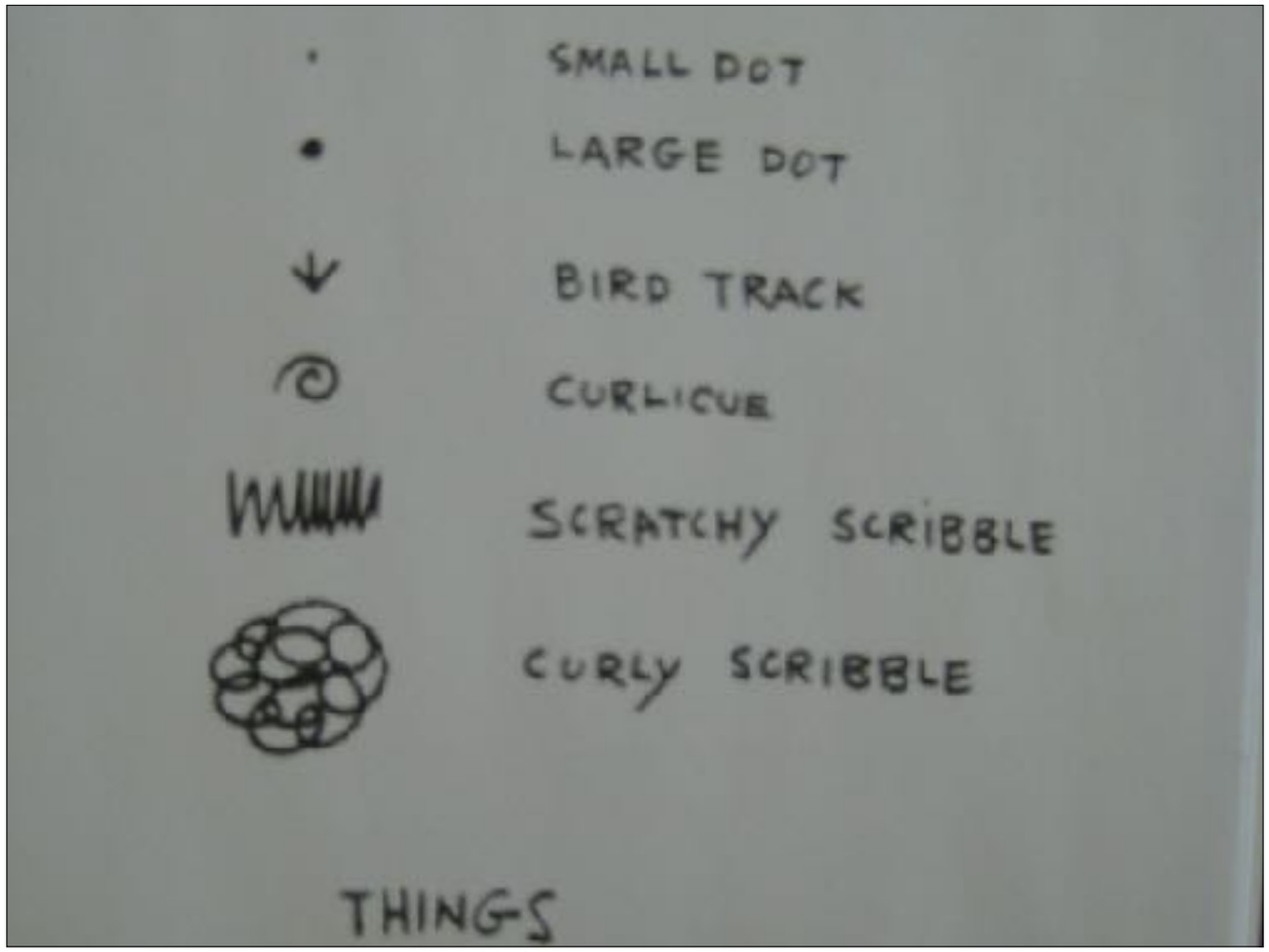
in Kirk’s words

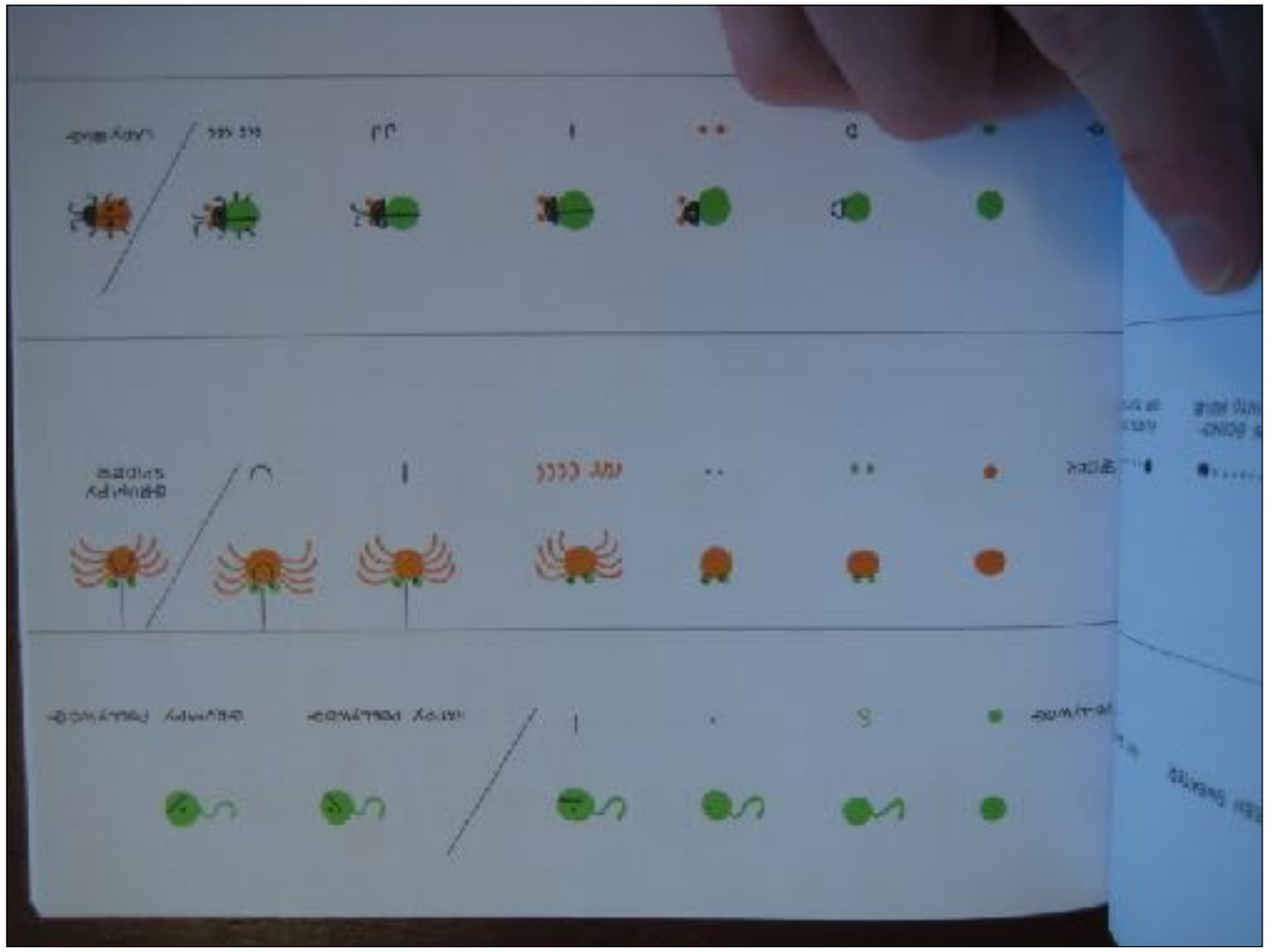
Growing up, "Ed Emberley's Drawing Book of Animals" was very important to me and my dad. It teaches kids that they can make animals of all types just by drawing letters and symbols that they already know how to make.

This year, I decided to make an Advent Calendar for the Days of December, 25 in all. Each day has an animal that Ed Emberley taught me to draw and that I taught a computer how to make- but each is also a little puppet that bobs and weaves or interacts with the mouse some other way.

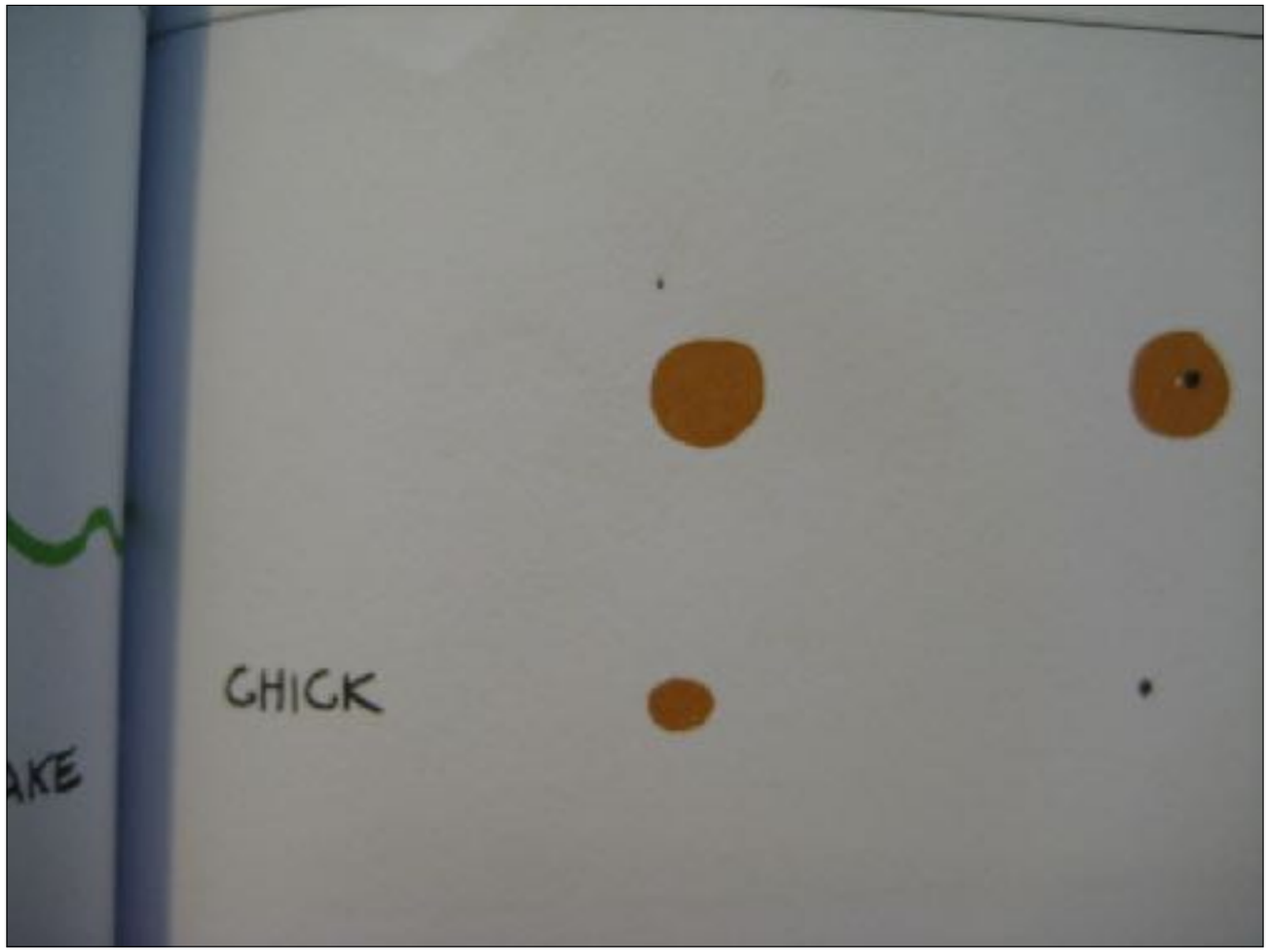
http://advent.kirk.is/
A new animal puppet is unlocked every day!

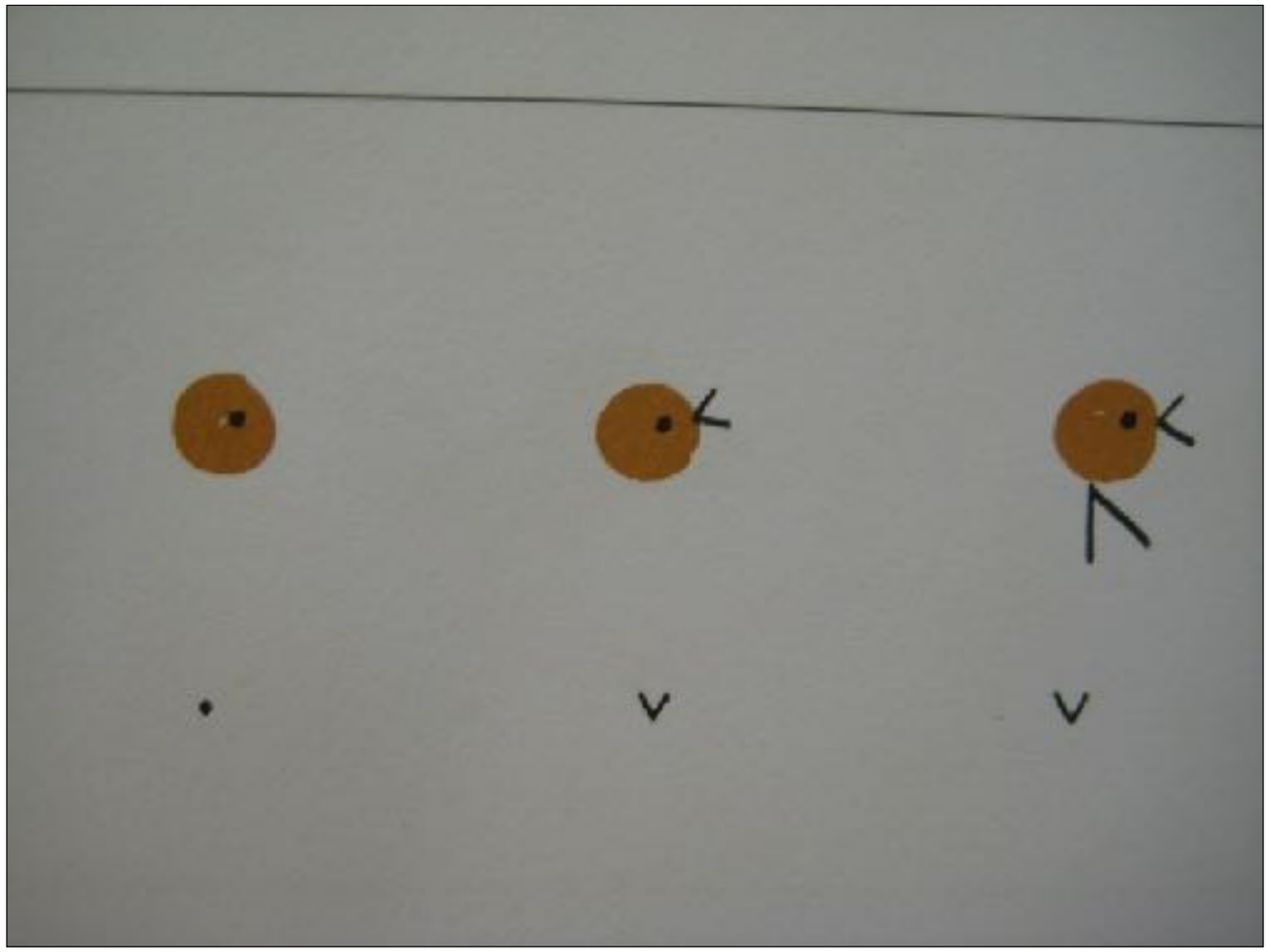

ED EMBERLEY'S
Drawing Book of
ANIMALS

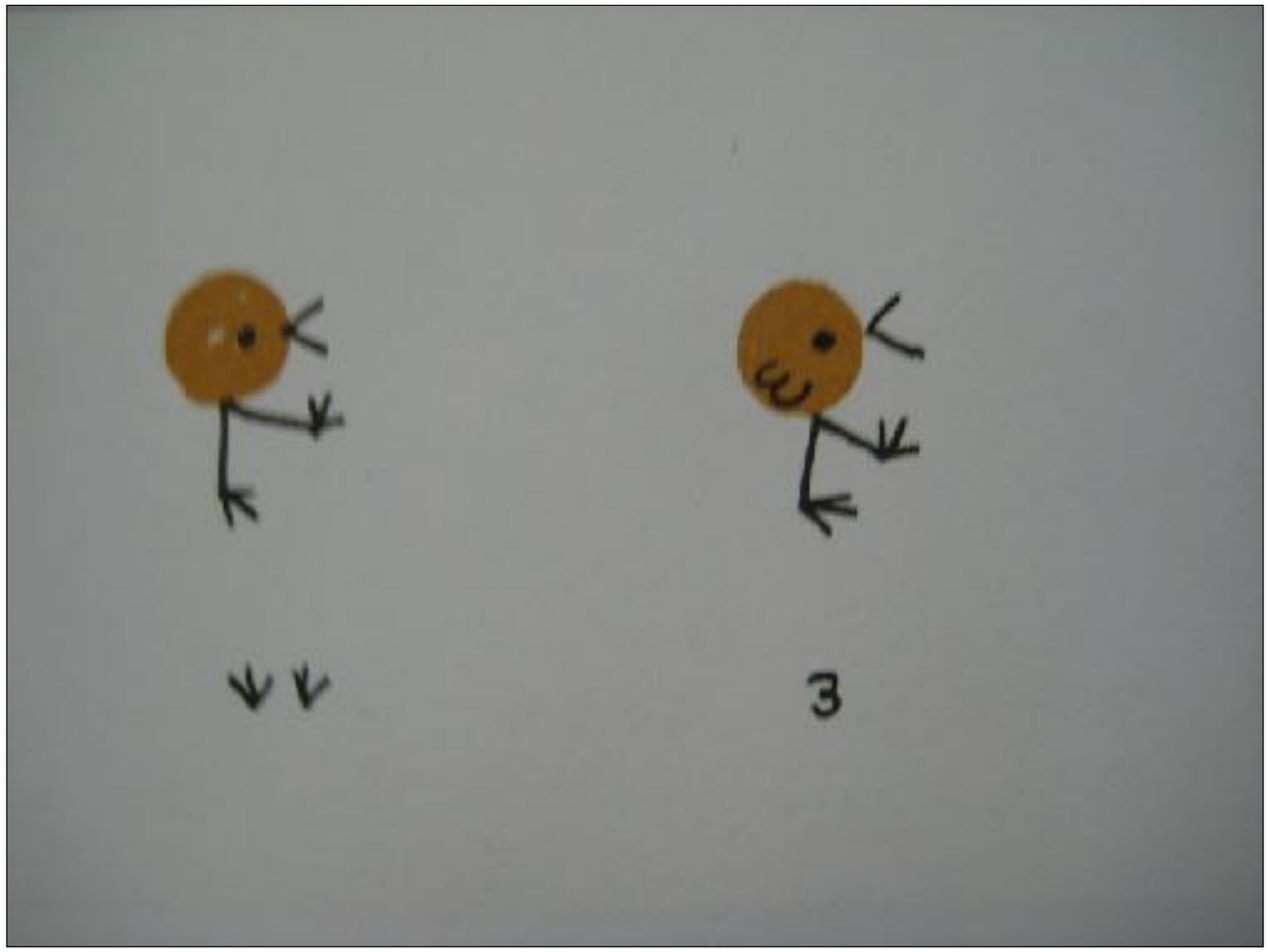

SMALL DOT
LARGE DOT
BIRD TRACK
CURLICUE
SCRATCHY SCRIBBLE
CURLY SCRIBBLE
THINGS



BUG
D
CHICK
LADY BUG


CHICK
AKE




3

Kirk's UI Dev Blog

Showing posts with label **advent2014**. Show all posts

TUESDAY, DECEMBER 23, 2014

modern web tricks

I made my annual virtual advent calendar again, and this year I focused on the work of Ed Emberley. I'll do a "director's commentary" for the 25 puppets once they've all been unlocked, but for now I'd like to talk about some of the little tricks I used in assembling the main launch page for this year's calendar.

(It has been drawn to my attention that some of my work online is older, and doesn't exemplify best web coding practices, and so going into the future I'm determined to always try and put my best face forward with this stuff (and shake some old functional-but-ugly habits I picked up along the way.))

Ed Emberley has a series of books teaching kids and grownups how to draw things one simple shape (a square, a circle, maybe a letter) at a time. I highly recommend

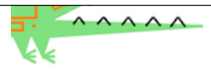
ABOUT THIS SITE

Kirk Israel is a Boston-based UI Engineer. This blog is all about making great UIs, with a heavy emphasis on HTML5 and appealing animations and interactions. See alienbill.com for some of his portfolio.

BLOG ARCHIVE

▼ 2015 (6)
▼ January (6)
note to self: reset intellij font size
text overlay effect with css masking
tags n stuff
semi-meta: blogspot tags and chronological order
animal advent ala ed emberley: director's commenta...
animal advent ala ed emberley: director's commenta...
► 2014 (100)
► 2013 (74)
► 2012 (55)
► 2011 (32)

snap.



Day 24: Dragon

What a beauty this one is! Very tricky making all those funky S-shapes. The fire and smoke was my own creation. I was pleased with the way I got the legs moving.

Day 25: Santa!

The only one that is my own design - and that has its own color palette, distinct from the animal puppets. (It's funny comparing his design to my old Advent santaskate - that the two are so similar reinforces the idea that Emberley was a lasting influence that I reinvoked when I started making processing applets without importing graphics.)

This Santa is actually an homage to a cardboard advent calendar I grew up with, and still makes a seasonal appearance at my mom's house... I made an animated GIF of it a while back, but the short of it is for my folks the pullstring was a slightly naughty joke.

I think I recycled some code from the whale for the presents, but added rotation. (Come to think of it, it's just my old Advent giftplosion.)

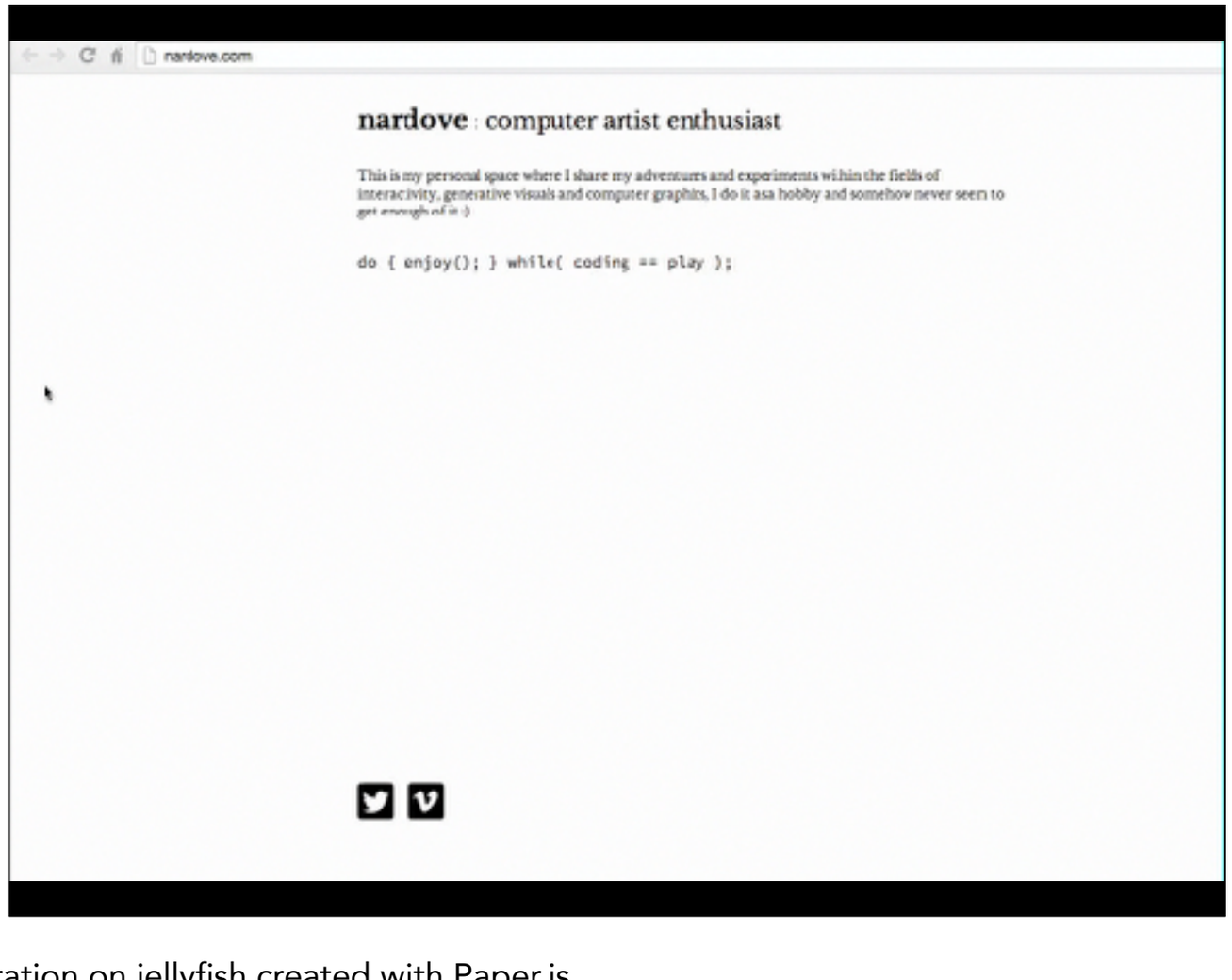
Posted by Kirk Is at 9:04 AM No comments:

+1 Recommend this on Google

Labels: advent2014

Inspiration

Or inspiration can come from revisiting concepts and ideas over and over.



Here is Richardo Sanchez's latest iteration on jellyfish created with Paper.js

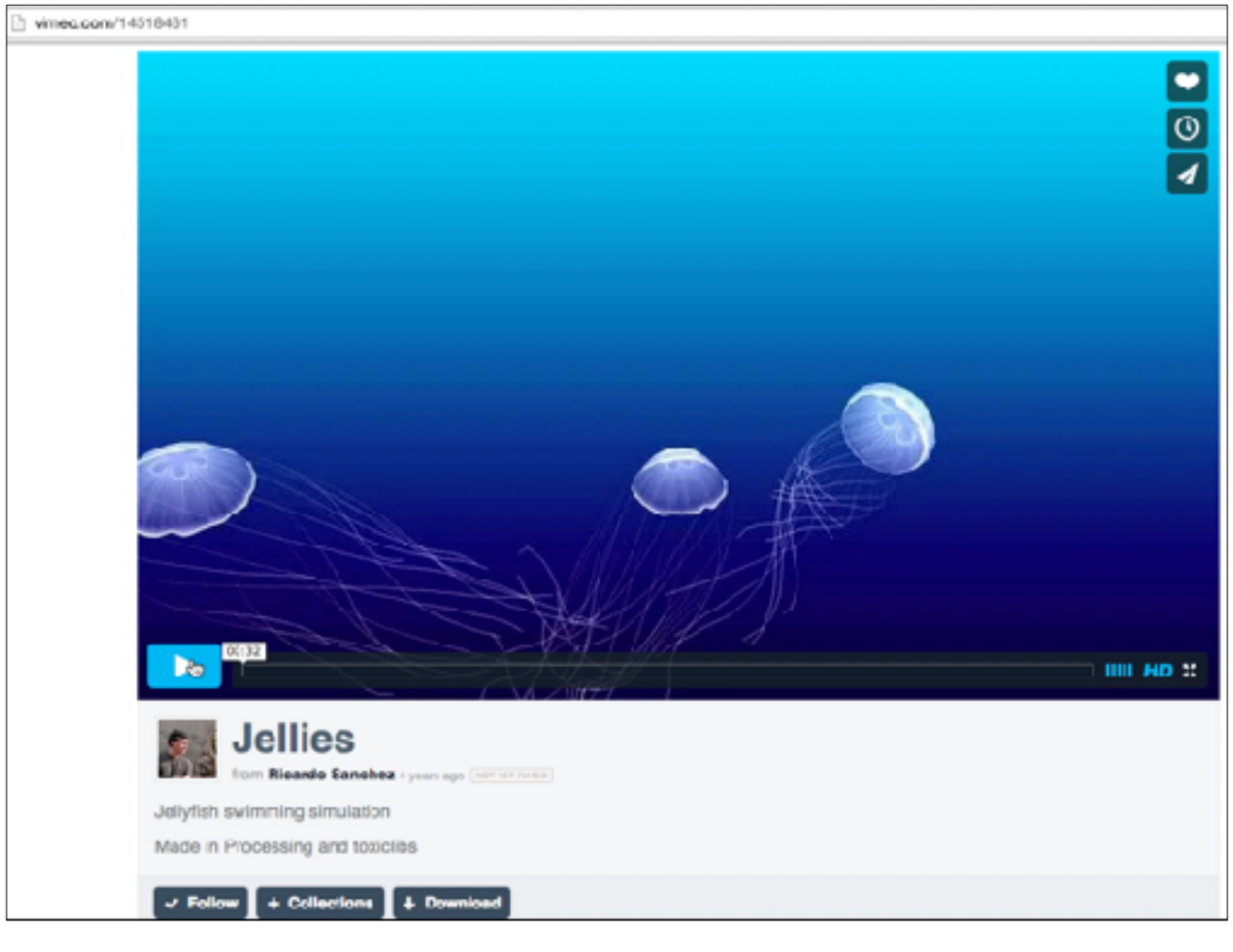
http://nardove.com/

http://www.twitter.com/nardove/

http://www.vimeo.com/nardove/

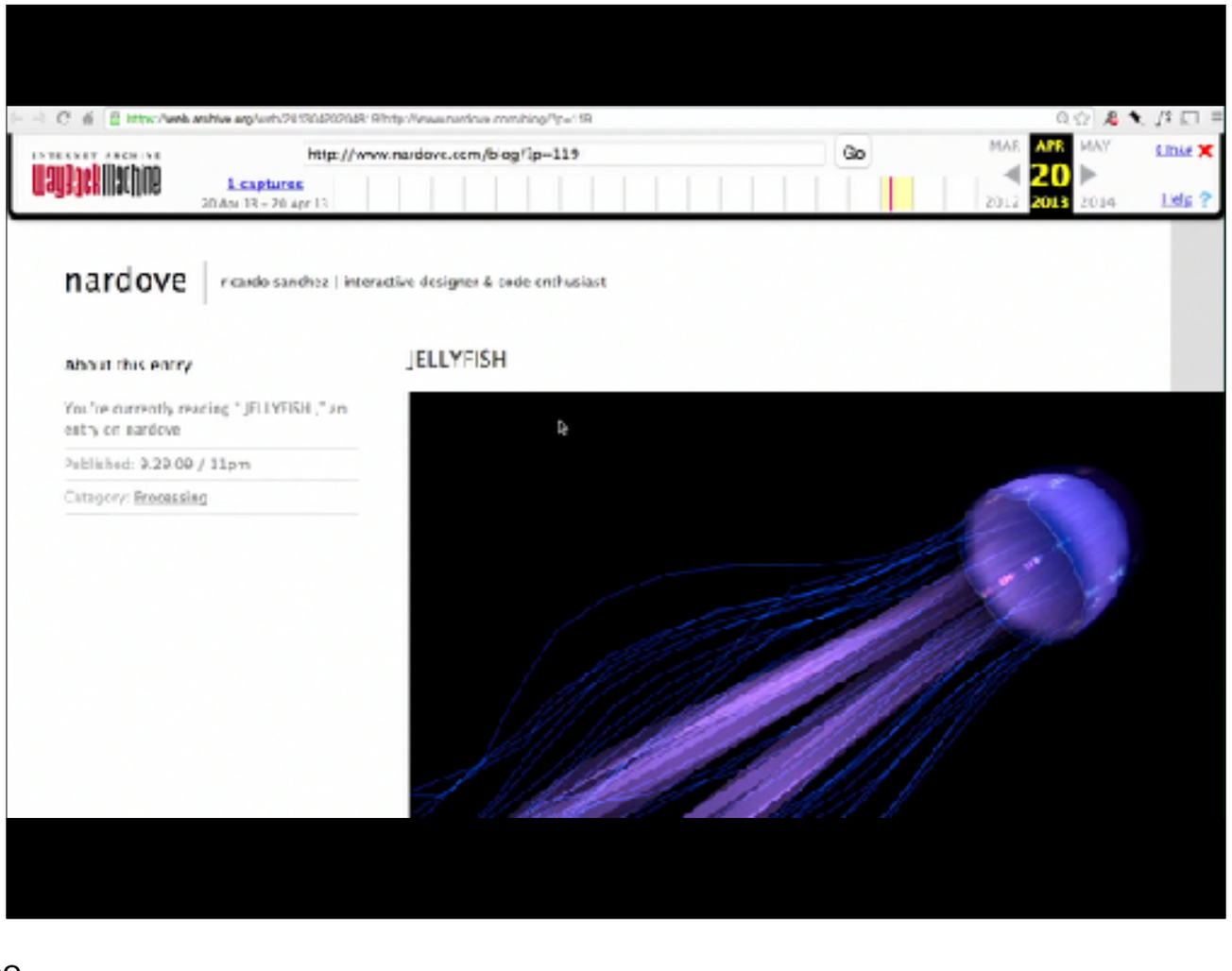
http://ricardodsanchez.com/

https://github.com/nardove



And here jellyfish from 2011 made in Processing

http://vimeo.com/14518461



And here an earlier jellyfish from 2009

Published: 9.29.09 / 11pm
Here is my first 3D programming attempt. Not sure what motivates me to do a jellyfish, but its been a very good learning project, I learn a lot of 3D texture mapping, some basic OpenGL concepts, basic lights and blend modes, organic motion simulation, and general programming stuff.
https://web.archive.org/web/20130420204819/http://www.nardove.com/blog/?p=119

3D jellyfish swimming simulation.
Personal project to learn some basics of 3D programming. The jellies are build using a hemisphere and sin/cos waves for the expand/contract motion, the tendrils are a set of verlet spring chains.
https://web.archive.org/web/20100315071609/http://www.nardove.com/jellies/

And assignments are another good way to start making things.

In the book Mamie Tinkler talks about an assignment from one of her Teaching Assistants.

“You’re supposed to somehow interpret the directions in a new way, let your own special point of view shine through, find the key that unlocks the [b.s] assignment and reveals the art within. The ultimate humiliation is, it’s not really your work. The assignment forces your “art” to act in service of someone else’s art values.

Albeit a somewhat polarized and pessimistic view of assignments, yes there are often more restrictive than one wants to be, but they are a great place to get started, or if you are blocked, or are just looking for another place of inspiration.

Daydreaming does not enjoy tremendous prestige in our culture, which tends to regard it as unproductive thought.... Unlike any other form of thought, daydreaming is its own reward. For regardless of the result (if any), the very process of daydreaming is pleasurable. And, I would guess, is probably a psychological necessity. For isn't it in our daydreams that we acquire some sense of what we are about?... Daydreaming is where we go to cultivate the self, or, more likely, selves, out of the view and earshot of other people. Without its daydreams, the self is apt to shrink down to the size and shape of the estimation of others.

— Michael Pollan, *A Place of My Own, p. 7*

Your assignment: You have been given an object and the above quote by Michael Pollan. Develop a visual response, or responses by our first class on Monday, 9/15.

Here are some assignments I received at MassArt last semester.

As a stimulus, I was given a quote on daydreaming by Michael Pollan from "A Place of My Own" and an actual mini metal slinky.

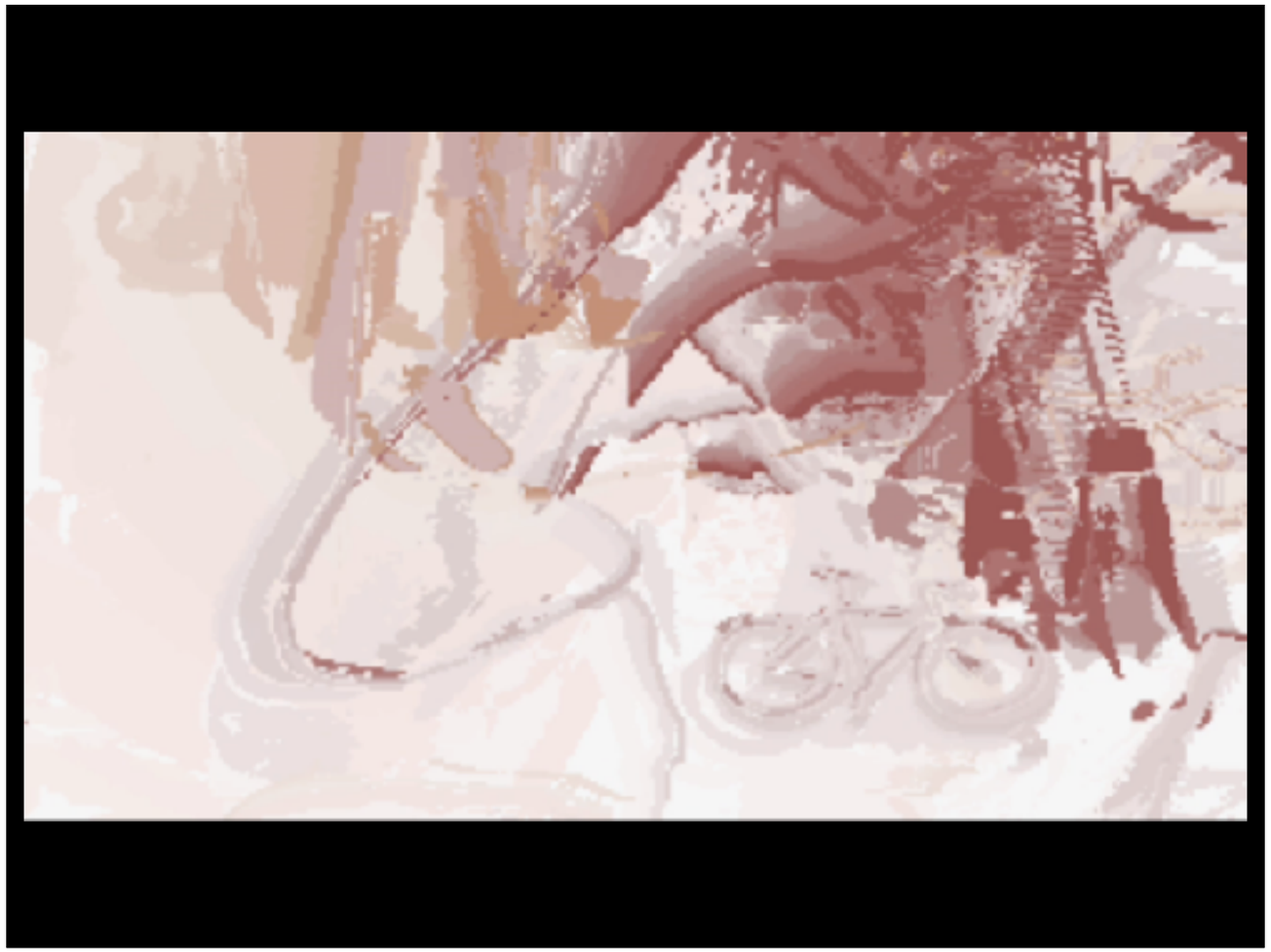
So my first thought was to relate daydreaming to fantasy and I imagined an endless staircase where there slinky could keep going down the stairs forever. This quickly evolved into a building a set of moveable stairs that would allow the slinky to fall forever.

Then when the slinky accidentally became stuck on one of the stairs and started making interesting noises I thought I could make an instrument. Which quickly made me think what if I video-taped the instrument and created visual light effects from the sound and visual and then re-projected that light back on the slinky instrument.

Which led me down a path of creating software that could record the slinky
and translate that into something. So I started a Processing sketch that
brought in video, recording only the motion information, then decided to overlay
2 cameras to make it more dreamlike, and realized that even with just motion information overlaying two videos was too cacophonous so I decided to make each feed monochromatic but monochromatic in the dominant color of the video feed at that moment.

While making the software I realized it could stand on his own and abandoned
the idea of projecting it back on a musical instrument.

Instead I started filming myself with the slinky. I thought that the juxtaposition of myself in front of a projection of the abstracted slinky would be interesting, I took many takes of this. Here is one take.

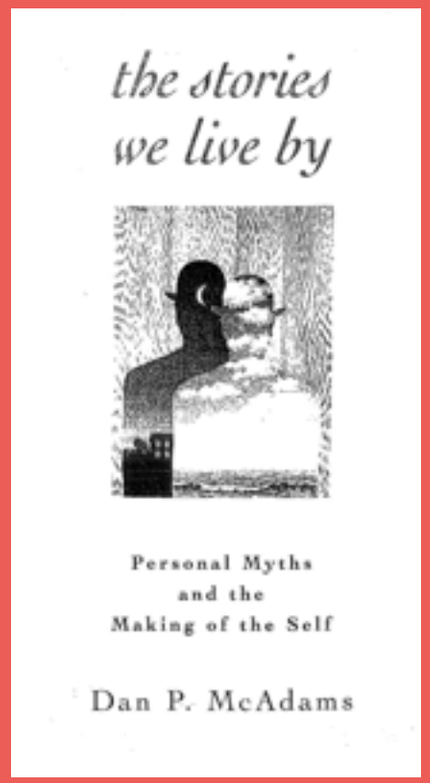


I then realized I didn't need to be in the shot directly, so eventually decided to just keep it abstract. So just started using the computer generated image directly as the final image.

So after several takes, I decided to start reading the quote, then a few more takes later produced a final video.

So here is the final art piece, that came from the original quote and slinky.

Prompts



Locking Yourself Out, Then Trying to Get Back In

—Raymond Carver

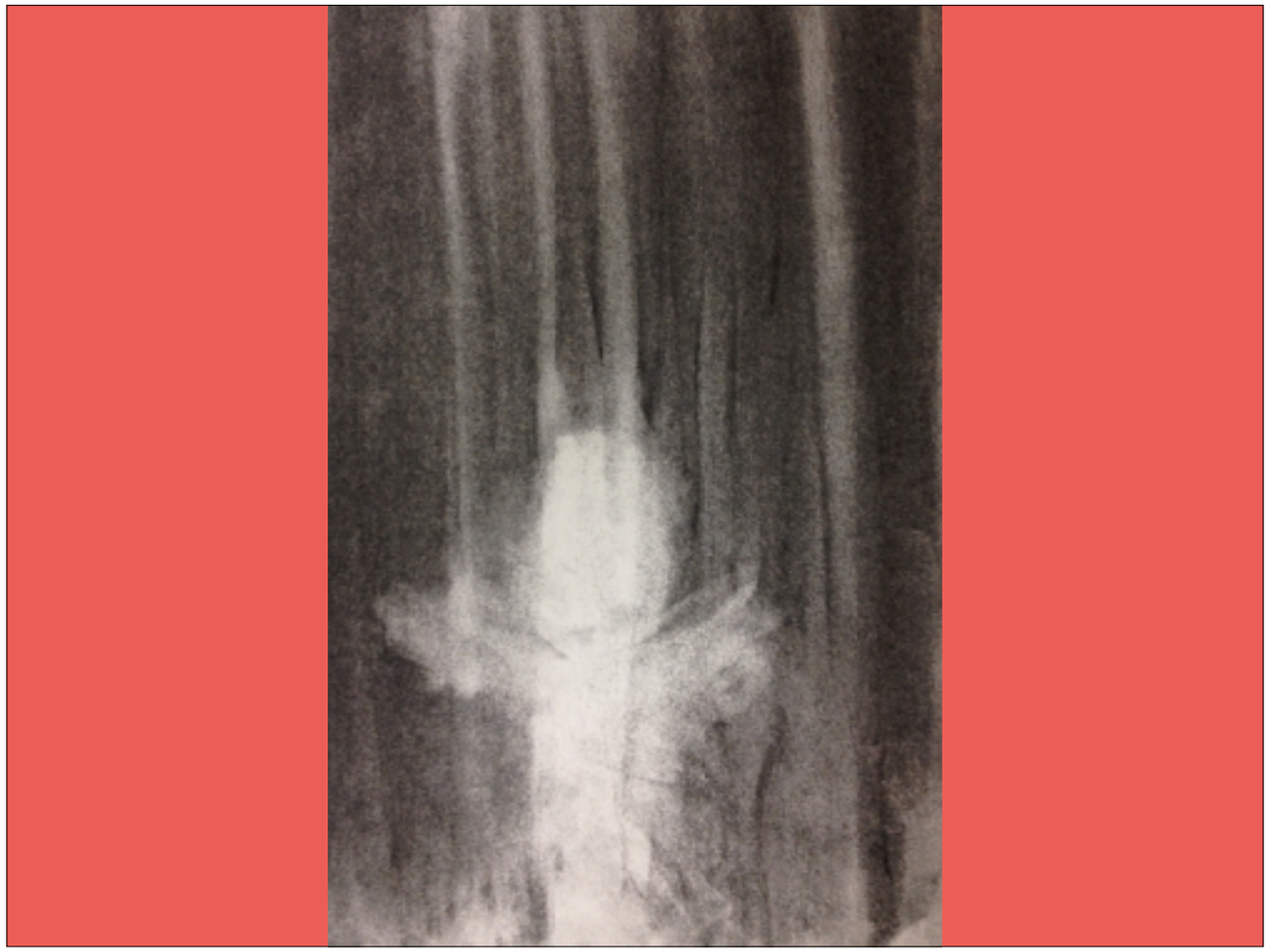
You simply go out and shut the door
without thinking. And when you look back
at what you've done
it's too late. If this sounds
like the story of a life, okay.
It was raining. The neighbors who had
a key were away. I tried and tried
the lower windows. Stared
inside the sofa, plants, the table
and chairs, the stereo set-up.
My coffee cup and ashtrays waited for me
on the glass-topped table, and my heart
went out to them. I said, Hello, friends,
...

Another assignment.

Responding to a short story by Dan McAdams on personal myth and a poem by Raymond Carver on reflection, memory and regret

My first thought was to animate some charcoal drawings, some kind of interactive game perhaps.

i started with charcoal drawings, exploring
locking yourself out, looking in



memories and self

but this didn’t really seem too related to personal myth it just seemed too abstract
so I decided to write things about myself instead.

my favorite sport is mountain biking even though
all my friends and family think that it is skiing

I read Harry Potter 3 times

...

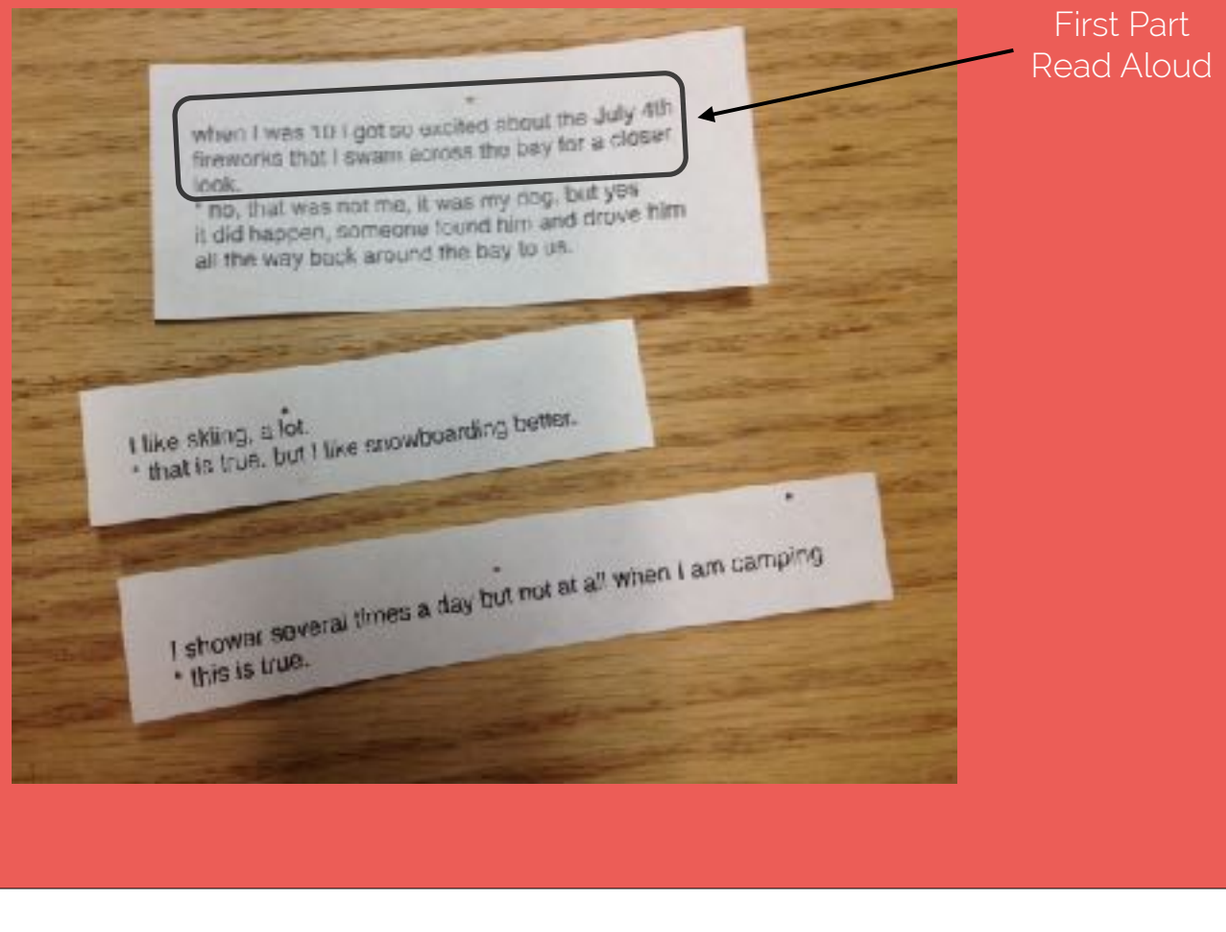
So I wrote true things about myself.

when I was 10 I got so excited about the July 4th fireworks that I swam across the bay for a closer look.

in kindergarten I was the "cool" kid

And then thinking about myth (IE responding to the part about myth in the original stimulus)

I wrote (in additional to the true things), untrue things (that were based in truth)



First Part
Read Aloud

And then after various iterations of this idea

I decided to print out each snippet of information on a small piece of paper, along with the full truth of the statement, 20 different slips of paper, 10 true, 10 mythic.

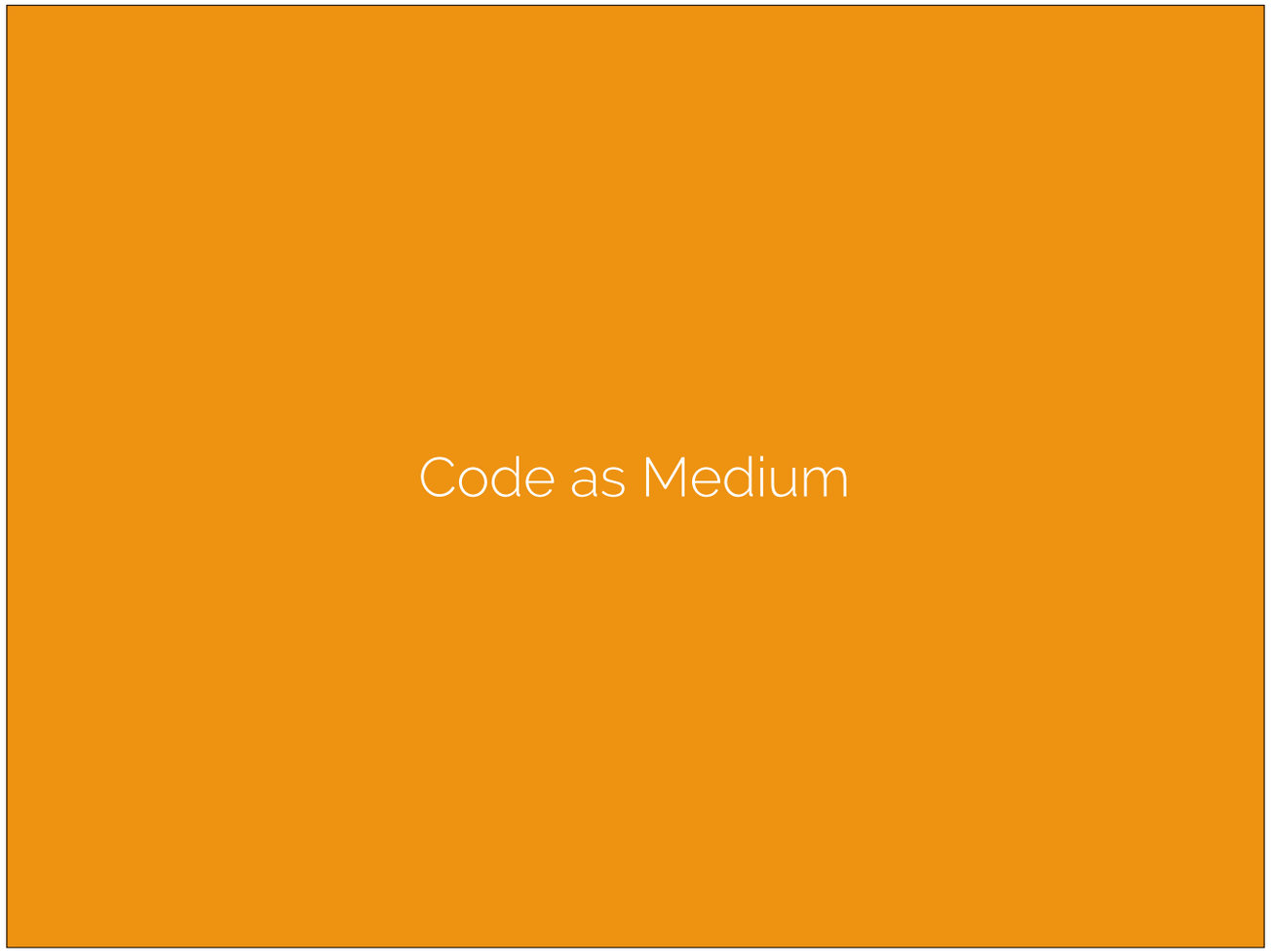
I then passed each person in my class 3 such pieces of paper (so each person received 3 random pieces of paper from the pool of 20 different papers).

I then read just the first part out loud for all 20 statements.

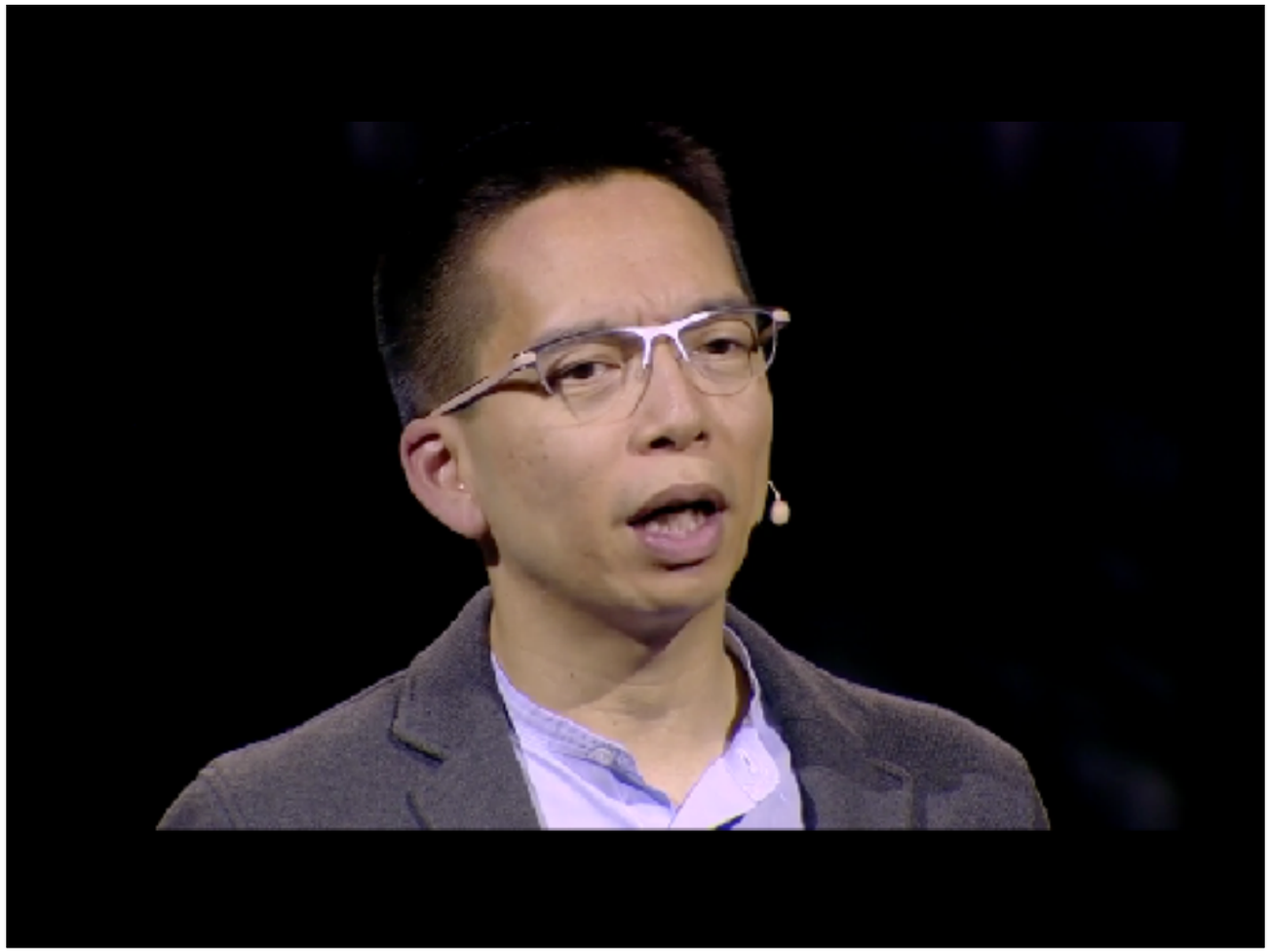
So each person had a different experience by being given different pieces of truth.
so of the 20 things I read, each person had additional insight into 3 of them

but all had the same experience of being withheld something from them.

This allowed me to consider intimately how my participant's experience my art.

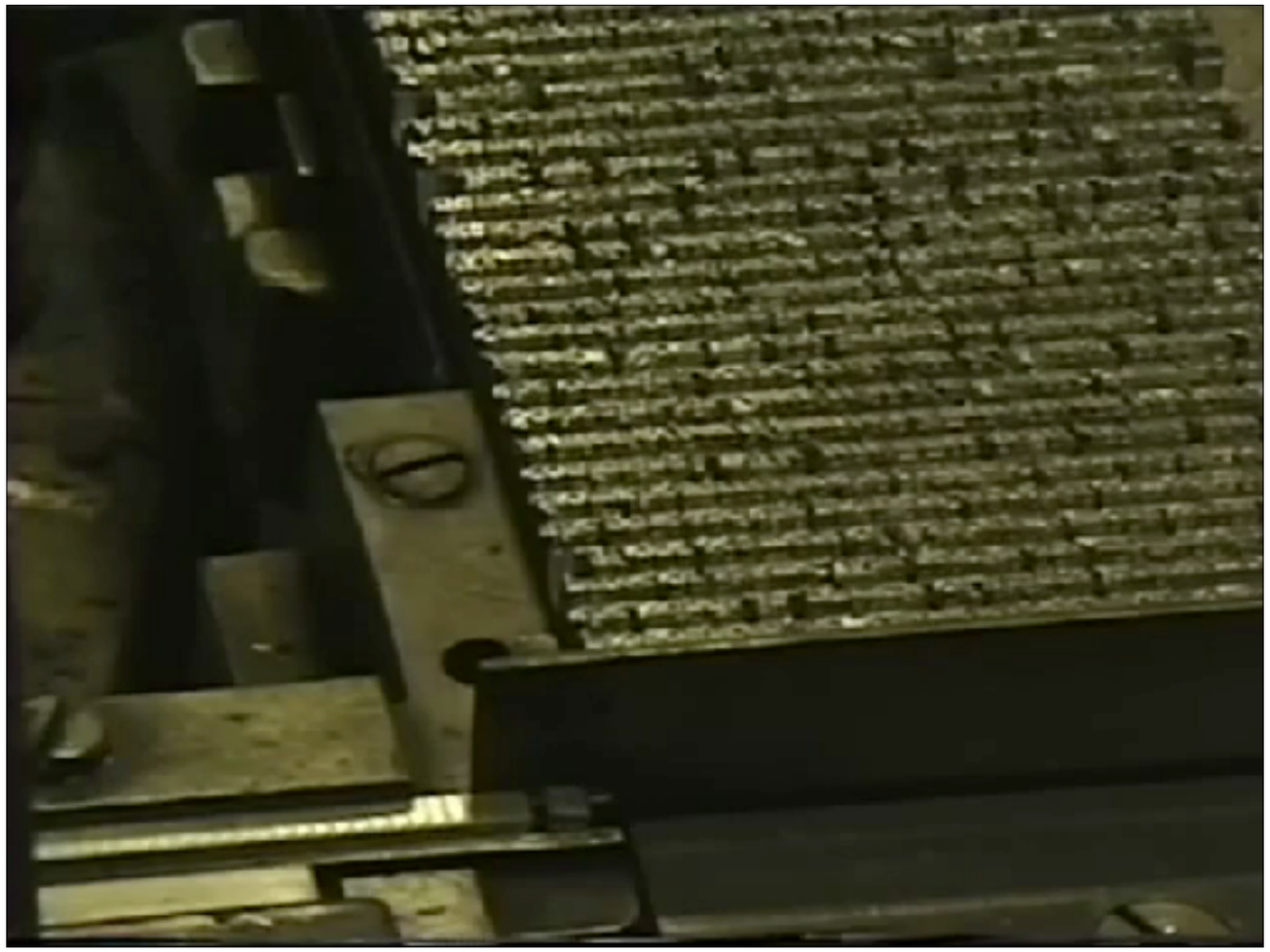


So, some things to keep in mind about software as a medium for art creation.



One thing to keep in mind is where your place of inspiration comes from, technology is not an end of itself. It should come from a place of inspiration. Here John Maeda speaks to that.

http://www.ted.com/talks/john_maeda_how_art_technology_and_design_inform_creative_leaders



“Dan running the caster on a job for others (Orestia, published by Leonard Baskin's Ghenna press, set by us, when the video was being made.)” —Julia Ferrari 7/28/2015

Video by Gahlord Dewald http://thoughtfaucet.com
Golgonooza Letter Foundry http://golgonoozaletterfoundry.com/

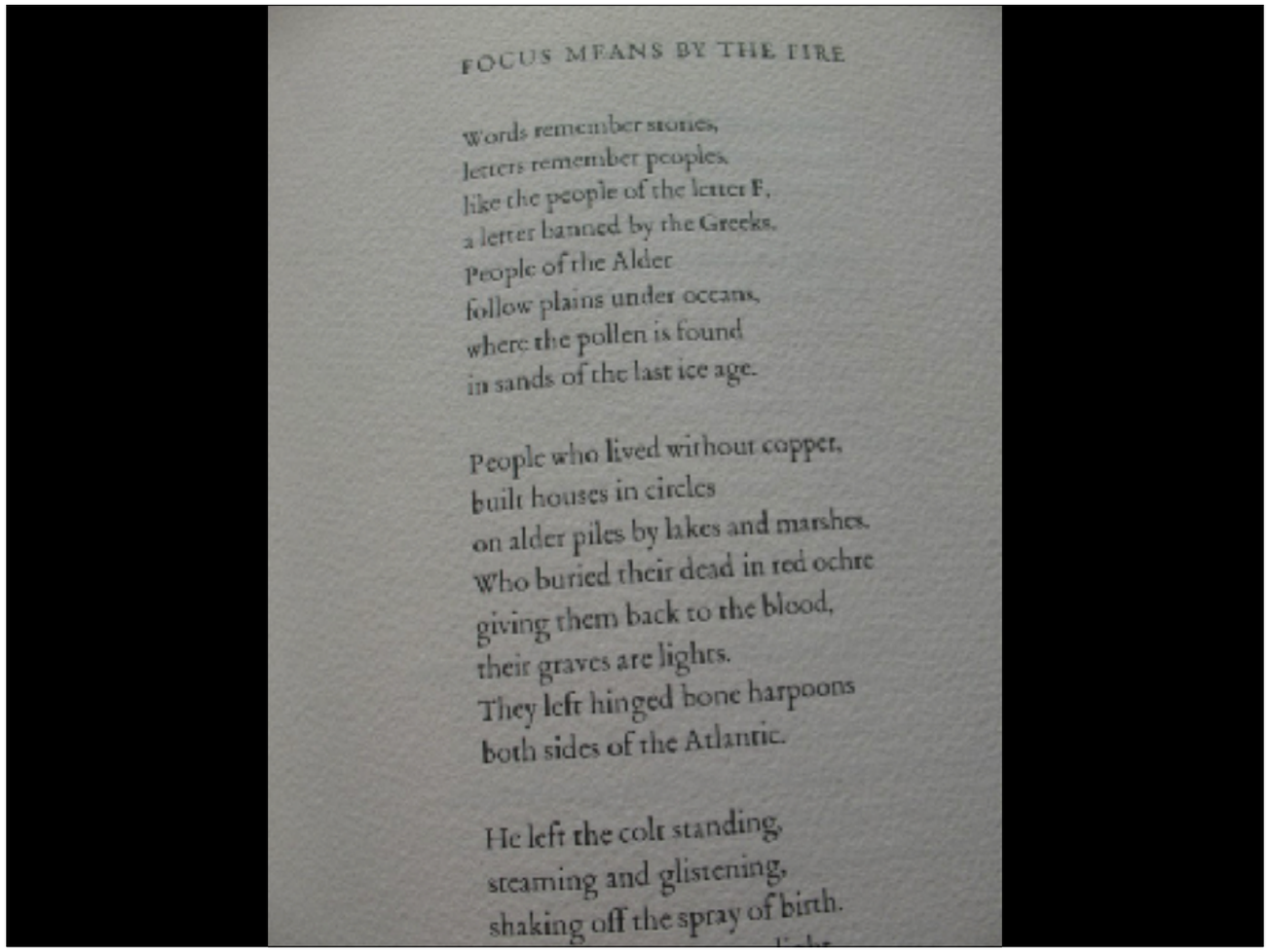
>start video, speak over clacking.
Currently, programming is not a very tactile medium and his mechanisms of feedback to you as a programming artist are often slower and more subtle than those of more physical based mediums.

Here Dan Carr speaks of the letterpress medium and casting type vs. hand cutting metal type.

> after done.
So, how do we get feedback, how do we respond to our own work. That is important to consider as we create.

“The last three slides are of personal creative book projects, making our own letters, etc. ”—Julia Ferrari 7/28/2015

“The last three slides are of personal creative book projects, making our own letters, etc. ”—Julia Ferrari 7/28/2015



“The last three slides are of personal creative book projects, making our own letters, etc. ”—Julia Ferrari 7/28/2015

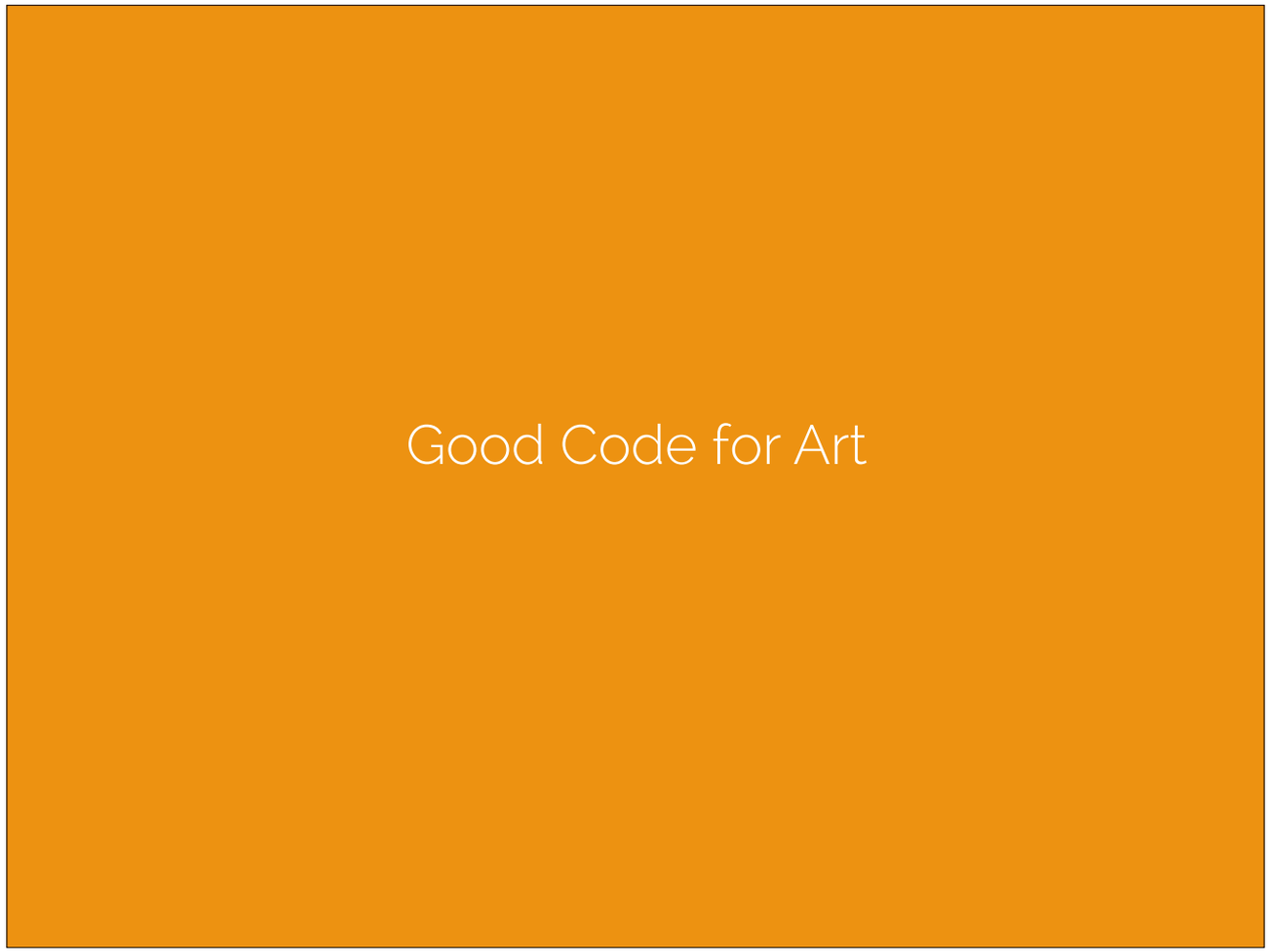
Code for Science/Business

vs

Coding for Art

I think there is often a difference between the process of coding for science or business versus coding for art.

Good code for Business, or production apps should be clean, should be well tested. It should follow best practices like those in the “Clean Code” or “Growing Object Oriented Software Books”, follow SOLID principles and use other design patterns.



But good code for art doesn’t need to be any of those things. The code serves to let you discover and explore. Bugs can be happy accidents or new discoveries or lead us in new and interesting directions

> Both groups are creative, imaginative, intelligent, energetic, industrious, competitive and driven. But programmers, in my [vast world-embracing] experience, tend to be painstaking, logical, inhibited, cautious, restrained, defensive, methodical, and ritualistic. Their exterior actions are separated from their emotions by enough layers of logical defenses that they can always say "why" they did something. Artists, on the other hand, seem to be freer, alogical, intuitive, impulsive, implicit, perceptive, sensitive, and vulnerable. They often do things without being able to say why they do them, and one usually is polite enough not to ask.—Ken Knowlton

Ken Knowlton has an interesting take on the matter from "Collaborations with Artists"

> read quote

So what Ken says about artists, freer, alogical, impulsive. These are useful qualities to consider while you are coding to create art.

OK. Before we start coding. Lets pause for questions.

Questions

What is something you wouldn't want to find under your bed?

What is an issue you care about?

What do you like to do outside?

inspirations / stimulus for project

Live-code Processing

Heather Hansen
The Value of a Line
—Ochi Gallery

Thank you.

Andrew Ringler

I will post slides at http://andrewringler.com/

Thank you.

Resources:

Boston Creative Coders (soon to be Tech Poetics)
Boston Cyberarts Gallery (@Green St. Station)
creativeapplications.net
http://hackingforartists.com/
https://www.coursera.org/course/digitalmedia
Processing by Casey Reas & Ben Fry [book]

Andrew Ringler

I will post slides at http://andrewringler.com/

Thank you.