



Iconolocode @iconolocode

Nov 2, 2021 · 27 tweets · [iconolocode/status/1455665313085894660](https://twitter.com/iconolocode/status/1455665313085894660)



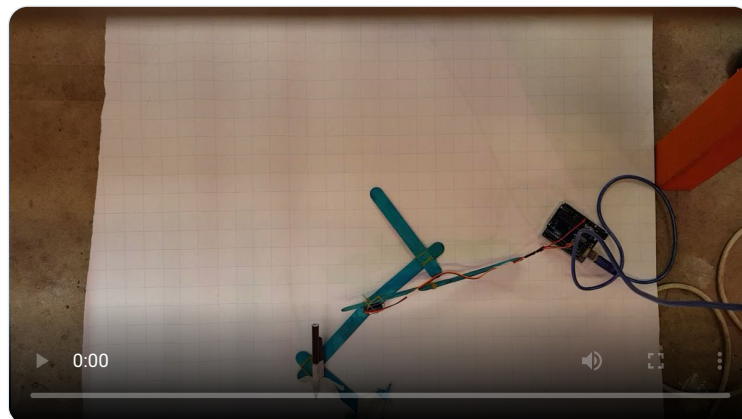
Writing an assignment reflection as a thread (because it's easy to incorporate media in Twitter):

This is a course from the KABK, also available for U. Leiden via the ACPA.

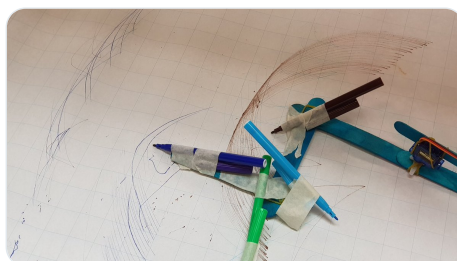
Hacking Tools as Artistic Research given by Anja Groten and Heerko van der Kooij at Hackers and Designers

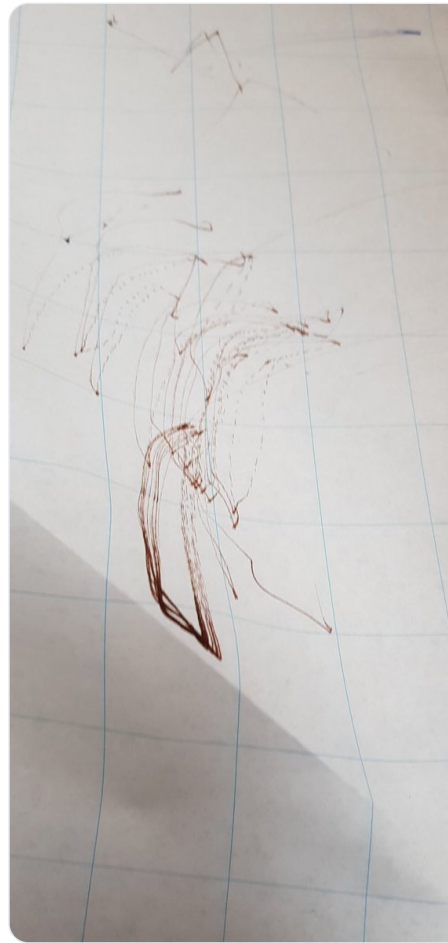
As a literature and art history student, thinking and writing are my main activity
By creating tools about these, I want to reflect on my process when writing and explore new thoughts about text and visuals.

Started first experimenting with
making drawing machines with Arduino.

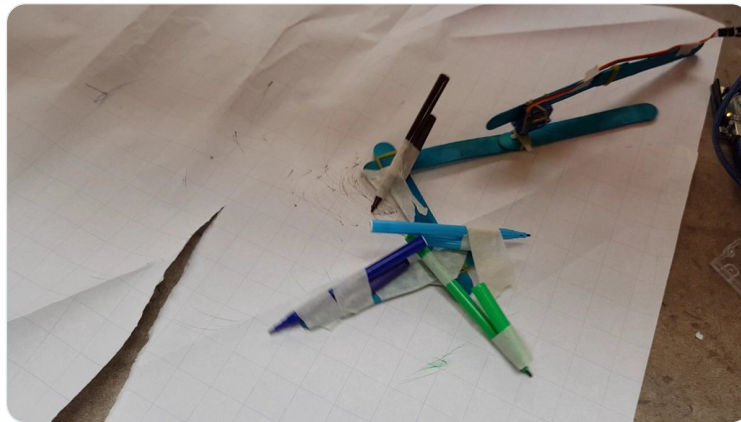


It was just popsicle sticks loosely taped to a simple Servo motor with an arc movement. But there was a lot of variety in the patterns it created.

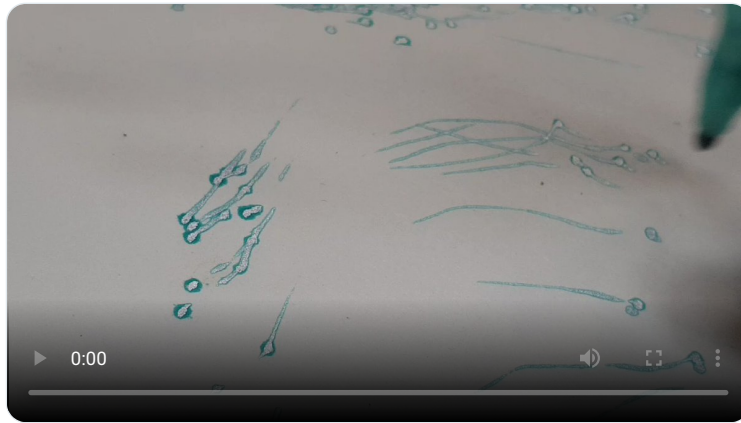




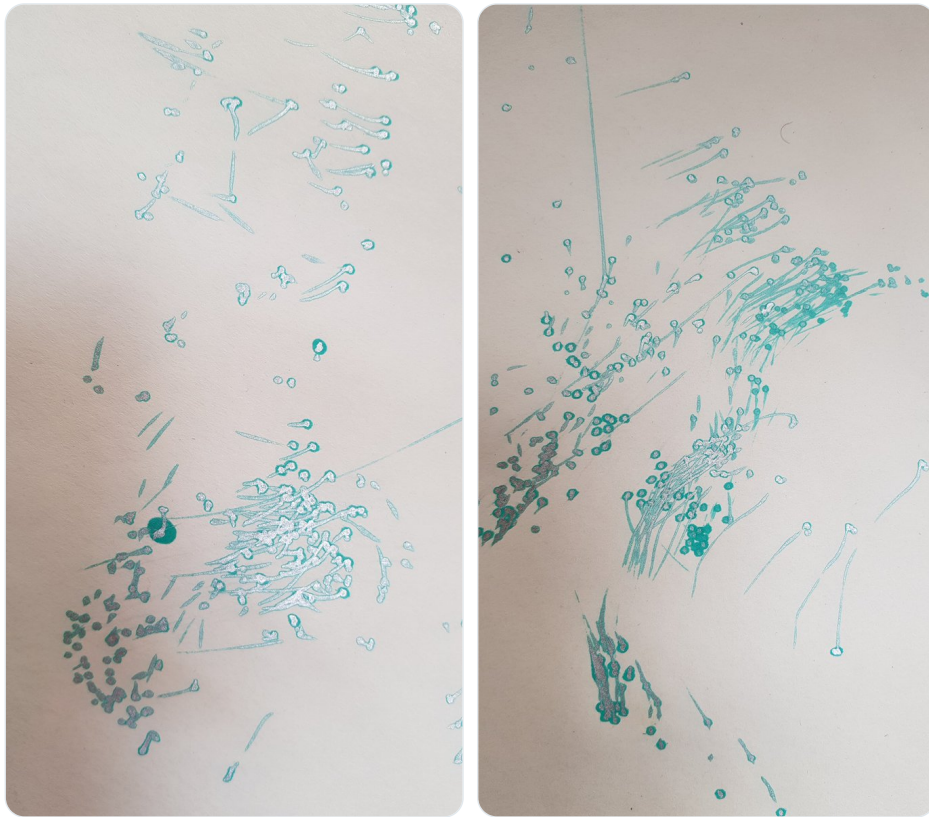
Next I tried to see how it would behave when confronted with "paper obstacles terrain"



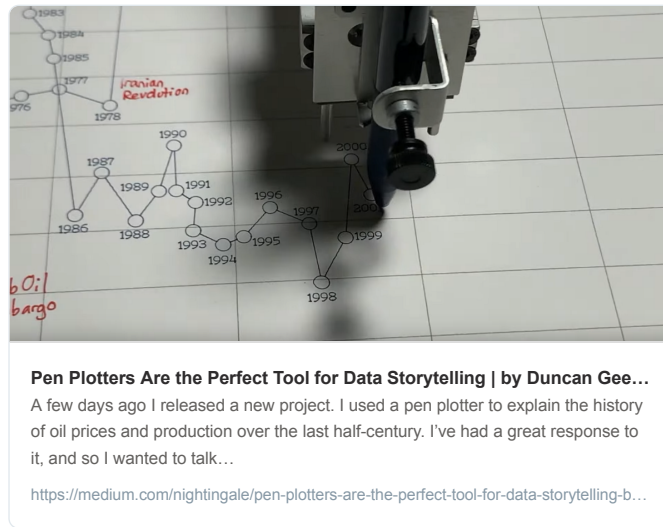
In the last experiment I modified the popsicles arrangement so that instead of lines it would make dots. And decided to sacrifice an outline marker. If the I hope Hema still has them, their effect is really great.



If I had access to a plotter device, I would love to explore this further. Use data visualization to control the placement, size and pressure of the dots, and the final viz would be the data conveyed through the material properties of pencils and ink.



I found that Plottervision made a great article and video about using plotters to do data storytelling.



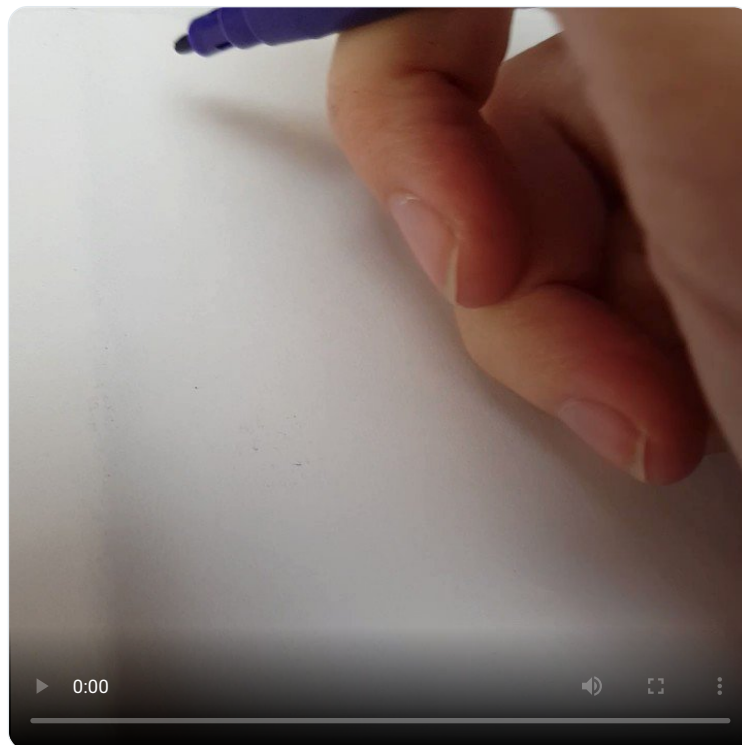
The result is really crisp. The machine makes "perfect" lines. I'm thinking more of a dirty/artsy approach, maybe I should do data viz by hand then?

The project of the day was a pen that would make what's being written unreadable due to small vibrations.

Maybe it's paper encryption?

Watch me try writing "hello world" with different settings.

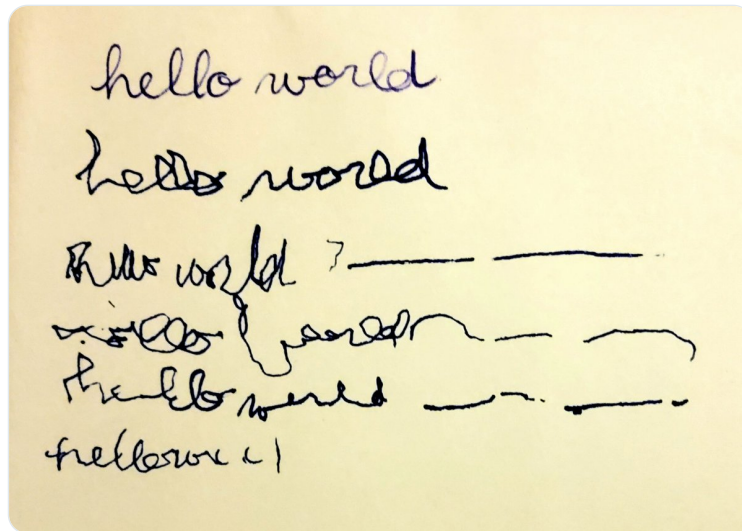
At the end I found an angle where the pen was a bit more stable, but it's still messy



In some way it was playful, me vs the pen, if it's readable I win.

I was "writing" for the sake of writing. It isn't text that's meant to communicate information, but it still has to be text that can be read.

Can this be considered as writing? Or is it a more pure way of writing?



It made me also think about "Hello world", a simple but well loved program to figure out if something works.

Hello world! is the excitement of running one's first program, or later on a day of hope when debugging.

The meaning of this small sentence goes way beyond of what its words say. Reading this sentence on a screen means that the computer is opening itself to you, a small fraction of the invisible computing machinery is shared with you.

It also means your code is machine readable and the instructions in are compilable. The programming language is different than the one that the computer runs on, but like sound waves in the air and then in the ear, through meticulous processes, communication is made possible.

My next explorations combine the realms of code and poetry.

Python is my favourite language, thus the poem should be about a python.



Found one on Project Gutenberg. 《The Python》 by Joseph Belloc Hilaire (1897).

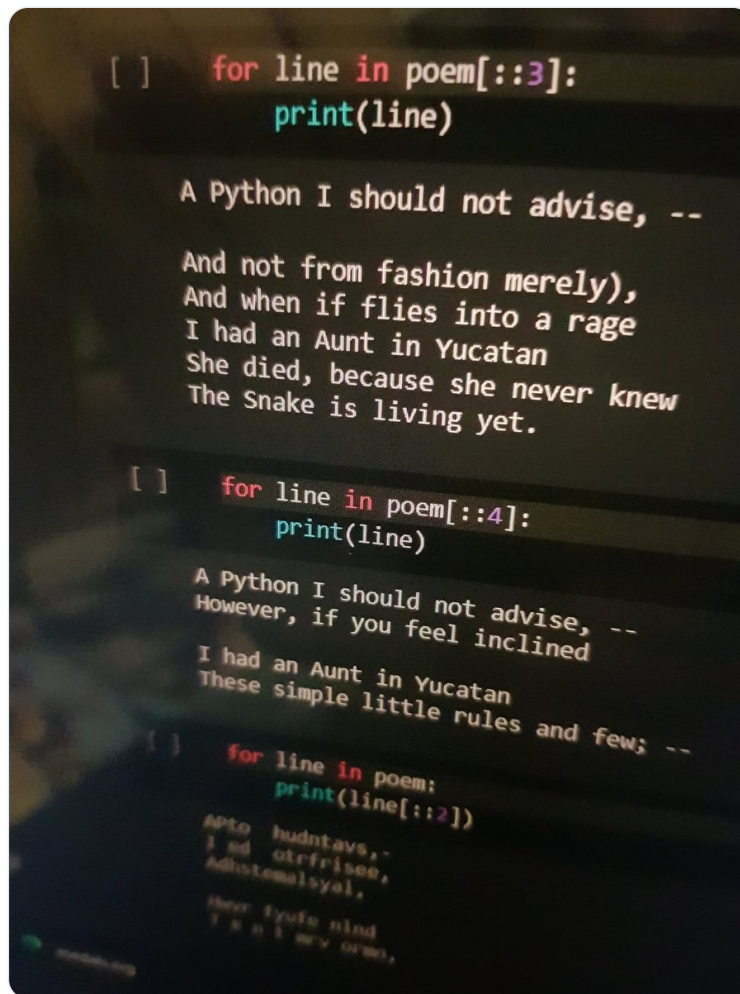
And I became obsessed with it.

Each time one or three lines of code manipulated the text.

Each time change the code slightly, the text becomes different.

Short steps, instant results, this feels quite gratifying each time.

This is how I ended up writing 100 small scripts that generate 100 different poems.



It's a nice number, I reluctantly stopped there. Actually this wasn't what I wanted to do at the start.

There are natural language processing techniques that I can't remember exactly how to code them. I wanted to make a small encyclopedia about them, the poem serving as example.

A Python I should not advise, --
It needs a doctor for its eyes,
And has the measles yearly,

However, if you feel inclined
To get one (to improve your mind,
And not from fashion merely),
Allow no music near its cage;

And when it flies into a rage
Chastise it, most severely.

I had an Aunt in Yucatan
Who bought a Python from a man
And kept it for a pet.
She died, because she never knew
These simple little rules and few; --

The Snake is living yet.



I was thinking of cool NLTK and spaCy functions.

I didn't get to these code libraries.

I didn't even get to touch the Python standard libraries.

I didn't expect going so far with bare Python.

I never thought it was possible was possible do so much with only the basics.

Just with simple python functions, there are way more options than I ever imagined.

Most of the time, when I code I do it towards a specific result that I have in mind.

This time I am looking and playing with the code and see later what results.

With this new perspective, I learned a lot of new things, and my appreciation for Python grew even bigger.

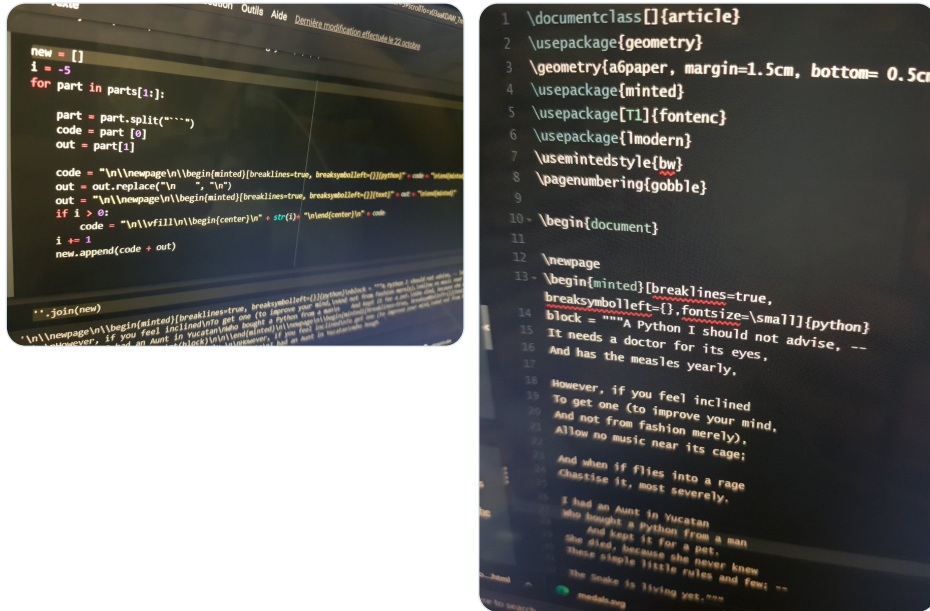
There is something beautiful in its syntax, it's not for nothing the adjective "pythonic" exists.

Is the focus here still the resulting poem? Or is the code the poem?

In any case, I didn't want this to become a forgotten file somewhere on a disk.

Again I started to code, combining messy Python with LaTeX to procedurally make a page for each script and output.

For this zine both the content and the format are result of code.



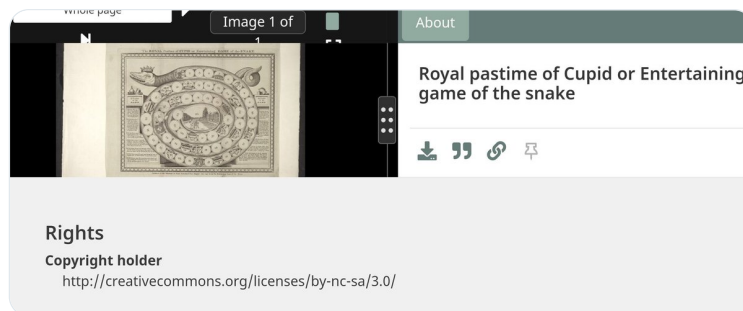
I wanted the zine to be the format of a smartphone, thus portrait A6.

The binding on the short edge, code on above page, output on page below, like an IDE or ipynb cell.

This is not common. Work around was to print in A5 two next to each other, and cut cut cut to separate them.



The cover is an old snake board game. The colors are put in dark theme. I cut out the corner that felt the most interesting to me. The text "entertaining game of the snake" and "rules of the game" and the visuals of numbered steps suit well the content, playful coding.



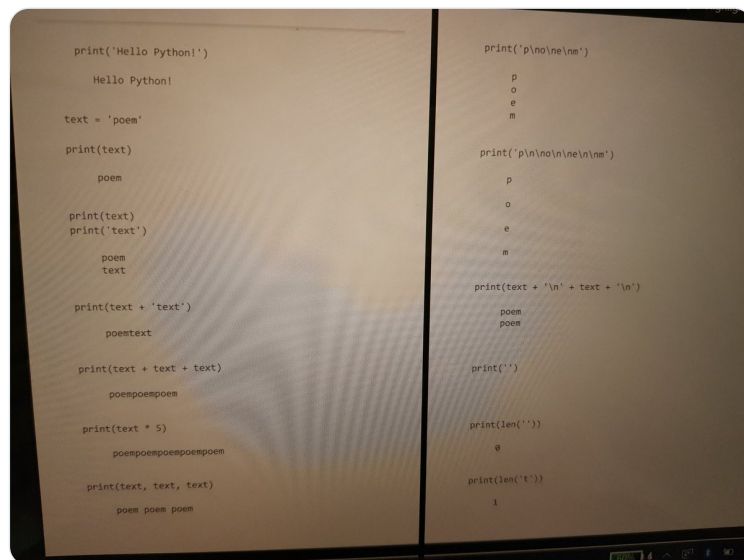
The image comes from <https://digital.bodleian.ox.ac.uk/objects/278f99e8-623d-45a0-93e4-631277631bcf/>

On the back of the book, to describe its content, I put a window with some python code. This was made with Carbon, a tool to display nicely code for tech influencers and educators.



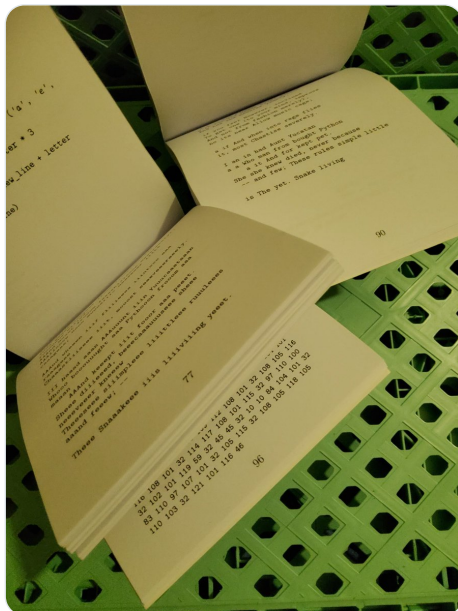
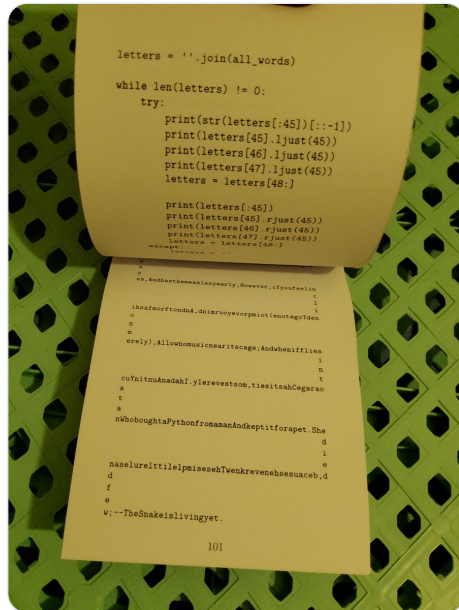
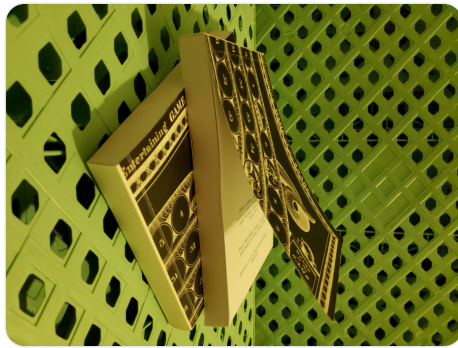
I wanted the zine to be enjoyable by anyone, regardless of programming knowledge. So I added 4 intro pages with simple code lines. Such as `print("Hello Python")`.

There are no explanations. But I tried to illustrate clearly the main concepts and Python is close to spoken language



The reader can choose what they get from the zine. Like reading the meaning between the lines of a poem, the reader can decrypt the functioning of the code.

Here is what the zine looks like:

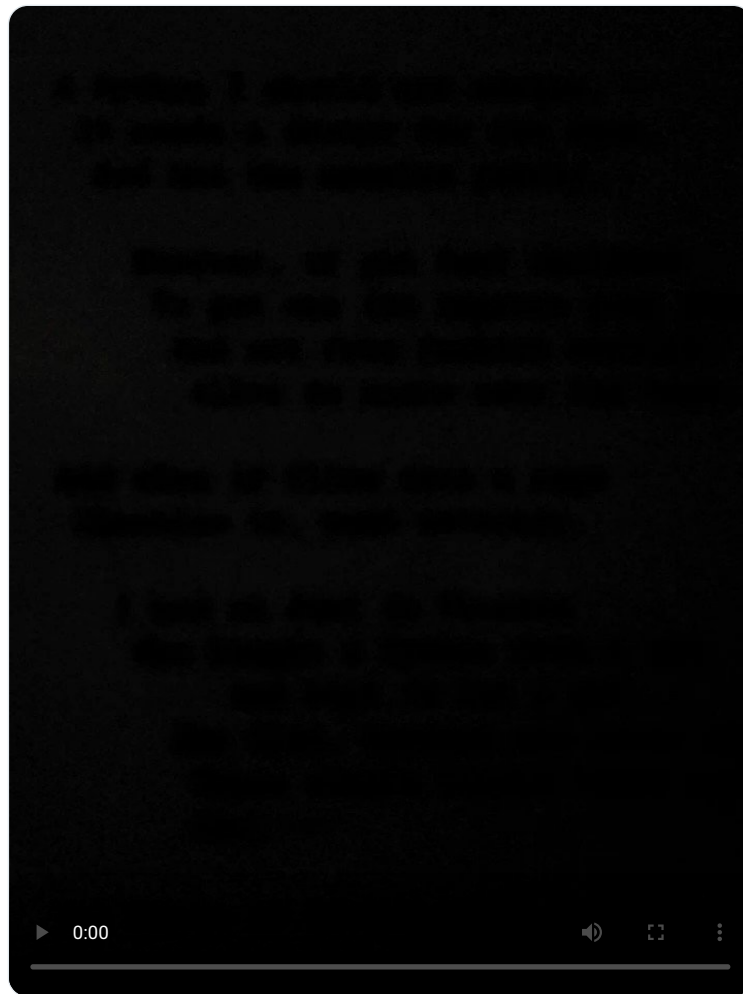


I'm happy with it, but still wanted to do something.

Going back to the digital, I made a gif from the poems and uploaded it to my ereader.

I choose the least optimised display setting for movement.

Even if it's digital, physical e-ink particles move, leaving dancing artefacts



My last project was a small code editor. I write a lot, and this made me reflect on how
how write.

I used Tkinter, which comes from Python. But it was not as flexible and pythonic. I
will try other systems. The writing options were quite limited, I tried to explore new
ones.


```

def __init__(self):
    """creates window"""

    Tk.__init__(self)

    self.title("NotManus++")
    self.scale = 1.5

    self.minsize(
        int(210 * self.scale), int(290 * self.scale)) # in pixels, A4like

    self.createWidgets()

def createWidgets(self):
    """adds text input"""
    self.grid()

    self.editor = Text(self, width=int(29 * self.scale), # in characters
                        height=int(9 * self.scale), # in lines
                        cursor='pencil',
                        spacing1=10, spacing2=10, spacing3=10) # space bet

    self.editor.grid(row=2, columnspan=2, column=0)

    self.editor.insert('0.0', # space fill
        ' ' * int((210 * self.scale) + (29 * self.scale)))

```

During the course I wrote my thoughts in Etherpad. It is possible to watch back the edits. It was quite fun for me to look at my past self writing.

There are a lot ideas in it I still want to explore.

19:37

🔍

🔌 29%

Hacktoberfest

My Drive - Google

Download/

Historique dyna

+

etherpad.hackersanddesigners.nl/p/sophiegr/timeslider#13

☆

⬇

⋮

Version 1396 Enregistré le 7 octobre 2021

Sophie+1 auteur anonyme ()

⏮

⏪

⏩

⏭

I used my tool to read, but most of its time of use was for writing. I connected a keyboard to it, and had like this a "modern typemachine" that I could use under the nouce weather whereas a smartphone screen wouldn't let show much use to the sun. Typing was something that got difficult for me as I had to do it more and more. My studies required longer papers each year, essays would be the replacement for presentations and exams during times of the pandemic. It was something I could only do inside, and the experience started to be dreadful.

First, when I could, I tried to write with pen on paper outside. But I often needed to consult documents or search information online, and thus needed to go in and out. Also I needed to type everything after that as handwritten submission aren't accepted, in the end it wasn't really a solution.

Session 2

6 October

Some notes on the readings: [write in phrases]

Text 2:

Buste vaporwave

NFT

Ubiquitous Computing

Text 3:

Confusion elements

Download folder search functions

Not showing file extension

Overlay lazynezumi

Reading mode

Color picker

Infinite remote / macro keyboard

Blue light filter

New Media objects _ tools

Paths:

Writing

Reading

Type machine editor

Pageless book

GUI manuscript

eTV

Computer book

Page turning mechanism

Artefacts

Plain text

Magnetic paper

Toy Book 0:00

🔊

⏏

⋮

0:00 / 0:00

...