Making an Asemic Art Book

Keith McKay

21st August 2024

- Lives in Glasgow, Scotland, UK.
- Now retired. It's the best job I've ever had.
- Formerly was an Environmental Radiochemist.
- I have been using ConTEXt on and off for almost 20 years. Nowhere near an expert.

What I will be talking about.

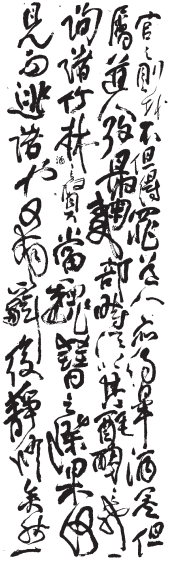
- What Does asemic mean?
- How would one code it?
- How I made the book Asemic Diagrams.
- How I could improve the code.
- Some examples of asemic writing, both hand written and code generated.

Meaning of asemic, and asemic writing

- Derived from the Ancient Greek *sema*, ‘sign’.
- Preceded by *a-* it is negated, *asema*, or without meaning.
- Definition of asemic - something as having ‘*no specific semantic content*’.
- When applied to writing, asemic writing is therefore ‘*writing without meaning*’.
- Given that name by two visual poets, Jim Gaze and Jim Leftwich in 1998.
- Asemic writing has existed long before there was a name for it.

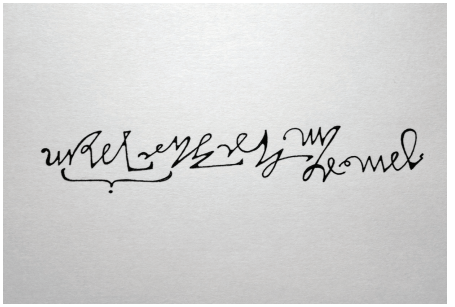
From *An Anthology of Asemic Writing*.

FÙ SHĀN



75

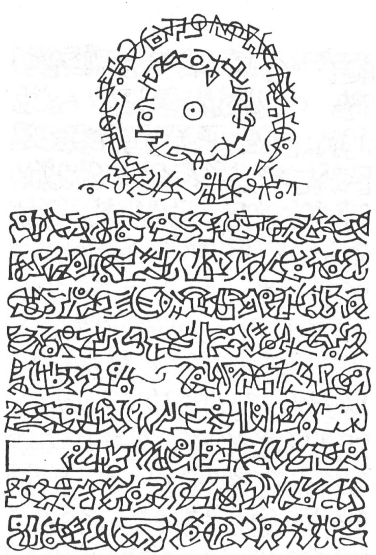
GEOF HUTH



93

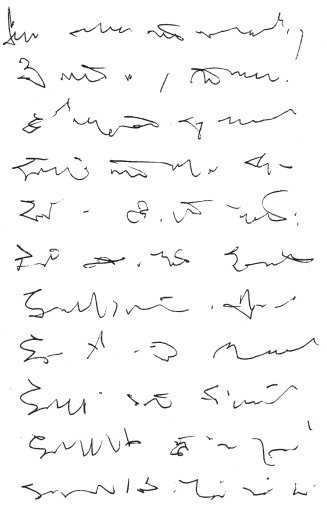
From *An Anthology of Asemic Writing*.

MICHAEL JACOBSON



97

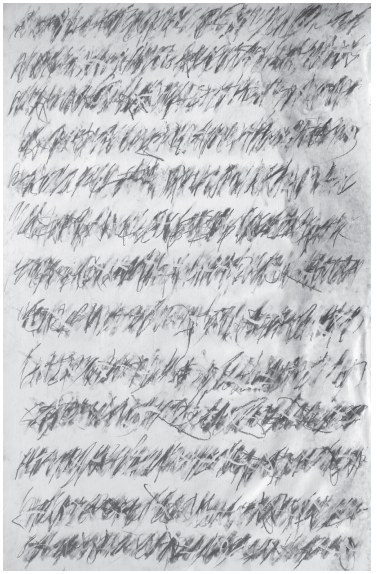
JIM LEFTWICH



113

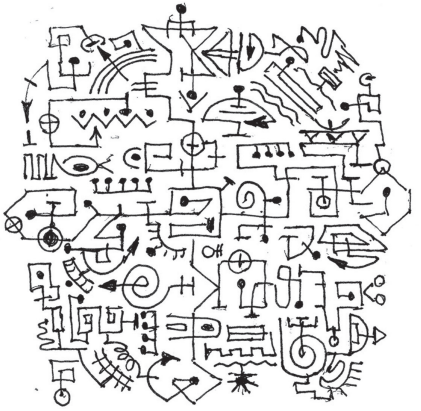
From *An Anthology of Asemic Writing*.

CECIL TOUCHON



175

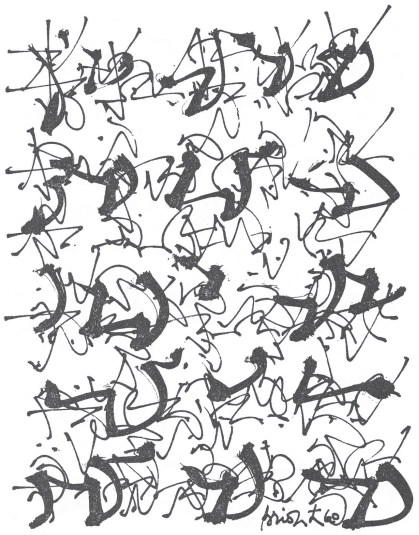
EDWARD KULEMIN



107

From *An Anthology of Asemic Writing*.

BRION GYSIN

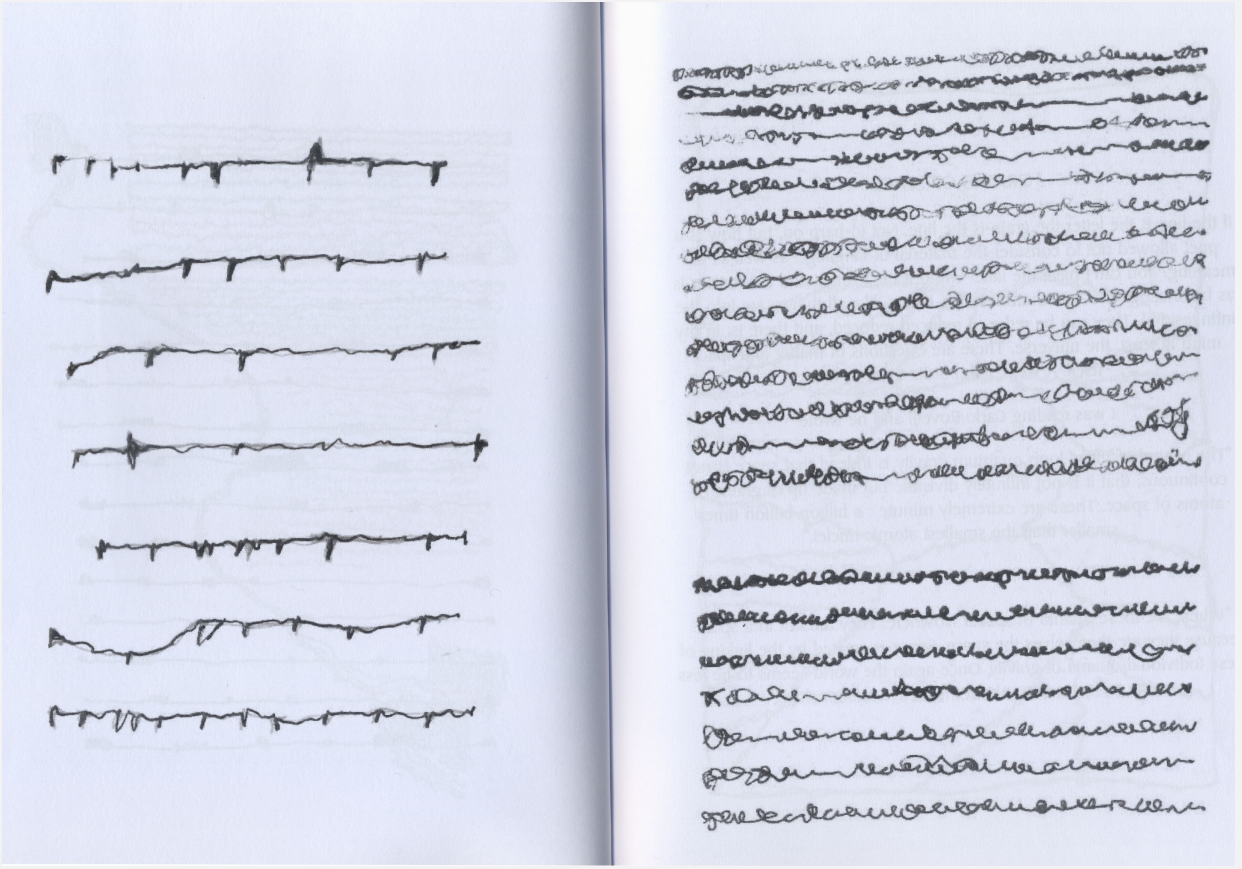


87

LARS PX

143

From *The Selected Scribbling and Scrawling of SJ Fowler*



By Amy Rodriguez and Kerri Pullow.

See:

- Asemica Poetica by Amy Rodrigues
- Kerri Pullow in WAAVE Global Gallery: Women Asemic Artists and Visual Poets 2021

The Book Structure

- I have had some experience in typsetting two novels, which my wife had written, however, '*Asemic Diagrams*' would be completely different.
- There were a number of challenges and hurdles to clear, but, some parts were easy to solve.

Let's look at the *Pros* and *Cons*

Pros

- Not much text to worry about. A short introduction, just over 2 pages. Two Appendicies, one page each
- Setups fairly simple.
- About 50 pages of diagrams.

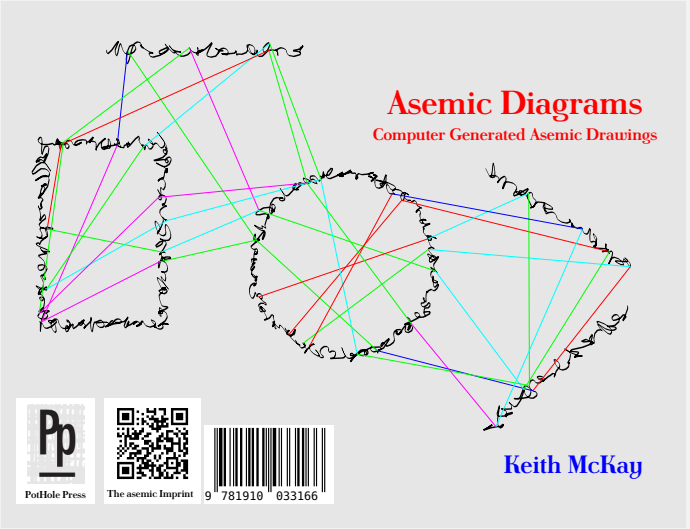
Cons

- About 50 pages of diagrams
- Had to learn some MetaPost, MetaFun, (Lua)MetaFun.
- Each diagram had to fill the the text area of a page.
- Had to get pagination right for the diagrams, since I wanted the recto page to mirror the verso page.

I was up for the challenge!

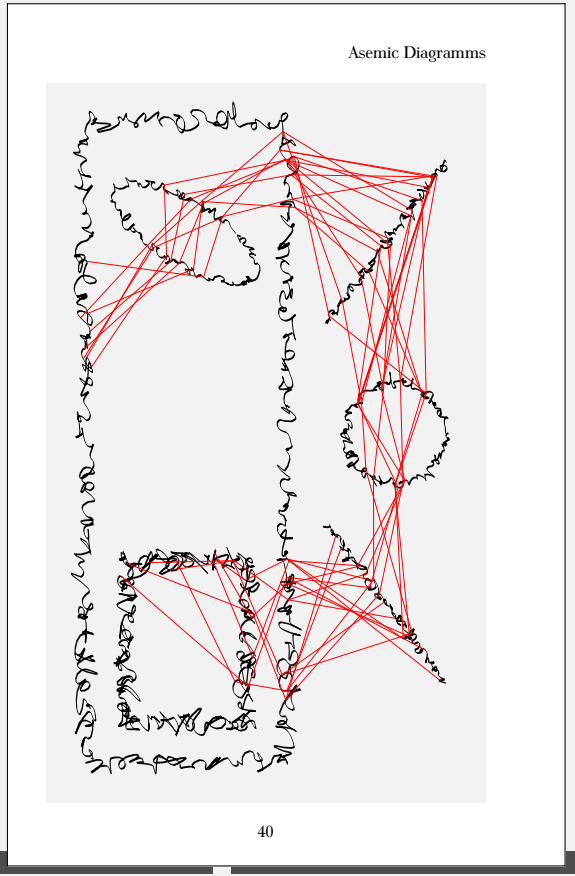
The book structure

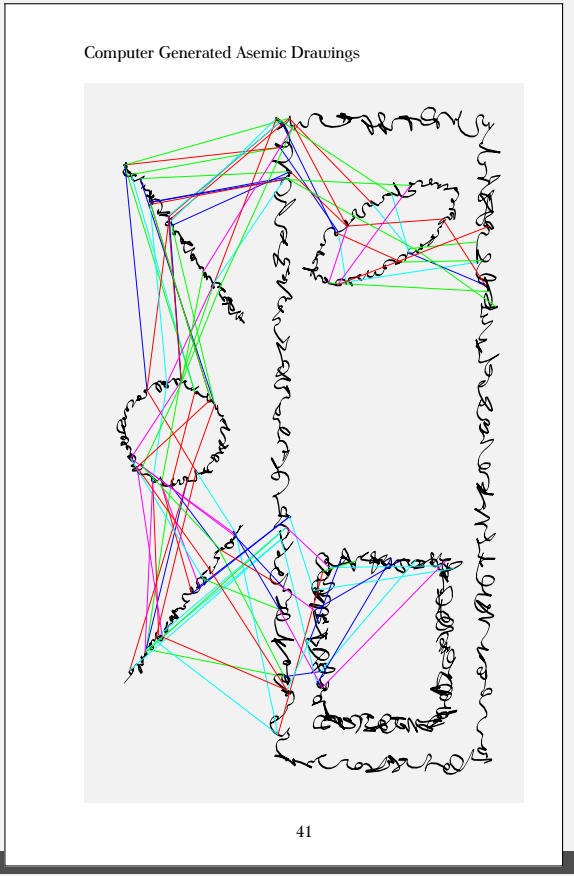
Cover
Title page
Publisher's page
Also by page
Blank page
Contents page
Blank page
Introduction
Title page before the diagrams
Fifty pages of diagrams
Appendix 1
Appendix 2



- Front and back pages of the cover.
- This was done separately from the book because I was going to self-publish through Amazon.
- Amazon provide cover templates of all the book sizes for KDP Amazon print-on-demand
- The cover has about 160 lines of code

Example pages from the book





The book setups

- Used the zint module to create the isbn barcode.
- Used Amazon book size 5.5 inches by 8.5 inches.
- Setup layout - maximised the text area whilst ensuring there was sufficient backspace.
- Setup page numbering - double sided.
- Setup body font - antykwa-poltawskiego default font size and font features.
- Setup the par builder - used for paragraph alignments.

Now the important ones

- Setup MPinclusions.
  - vardef randnumRange(expr mini,maxi)= generates a randum number within the range mini, maxi.
  - vardef lineAsemic(expr aPath,nPoints,minii,maxii)= generates an asemic path from an open path of nPoints and randnumRange.
  - vardef nonLineAsemic(expr aPath,nPoints,minii,maxii)= generates an asemic path from a closed path of nPoints and randnumRange.
  - def coloredPath(expr s) text c= a pen circled scaled s with colour c

. . . now the hard part.

- Setup MPgraphics.
  - Each MPgraphic was positioned in the Text area of the page with a gray background.
  - Every asemic path had to be drawn within the MPgraphic area - all were positioned relative to the llcorner of the text area.
  - Start and stop positions of the lines between the asemic paths were calculated using and stored in an array.
  - Lines were drawn between the asemic paths in red on the verso page, and a random colour on the recto page.
  - The recto page image was reflected about the vertical center of the Text area

Back to another easy part

- Setups for the text, indenting, headers and header texts, pagination etc.
- Create Title, Publisher's, and Also by pages.
- Create Contents page.
- Write an ‘arty’ Introduction.
- Add filler blank pages to ensure the asemic diagrams start on the verso page.
- Write Appendicies one and two.
- Design Cover using the template from Amazon.

Code has meaning

- Randomness is key to coding anything asemic!
- MetaFun has a number of macros available to make paths random.
  - randomized (the main one).
  - ramdomshifted.
  - randomizedcontrols.
  - randomrotatedcontrols.
- Not what I was looking for so I decided to code my own random number generator, and to learm more about MetaFun.

Code has meaning (Continued)

- My first attempt was so bad, but it worked.
  - Calculated each point on the page, added a bit of randomness and joined to make a path.
  - Created vardefs for horizontal, vertical, circular asemic lines . . . it was a mess, and not very efficient.
  - Similarly for asemic words but worse. Took ages to get it to work.
  - Used this method for a few years whilst tinkering with it from time to time, but with not much in the way of improvement.
- Then I found . . .

Code has meaning (still continued)

on and *along*

- Using *along* fitted in well with what I wanted to do. Here's why.
  - *on*: gives you the point at the supplied distance from 0 on the path.
  - *along*: gives you the point at the fraction of the length of the path.
- *along* was the best option for me since I define the number of characters i.e. points on the path and can then calculate the position of the individual as a fraction of the total number of points.

Code has meaning (it goes on)

Here is the code that produces the asemic paths in the '*Asemic Diagrams*' book.

```
 %generating a random number between a maximum and minimum value
vardef randnumRange(expr mini,maxi) =
randnum := ((maxi - mini) * uniformdeviate(1)) + mini;
randnum
enddef;
```

Code has meaning (it goes on)

```
 %generates an asemic path from an open path.
vardef lineAsemic(expr aPath,nPoints,minii,maxii)=
path asemicPath;
asemicPath := ((((xpart point 0cm on aPath)/72)*2.54)*cm, (((ypart point 0cm
on aPath)/72)*2.54)*cm);
mini := minii; maxi := maxii; nbPoints := nPoints;
for x = 1 upto nbPoints:
    ycoord := ((ypart point(x/nbPoints) along aPath)/72)*2.54
              + randnumRange(mini,maxi);
    xcoord := ((xpart point(x/nbPoints) along aPath)/72)*2.54
              + randnumRange(mini,maxi);
    asemicPath := asemicPath ... {curl 100}(xcoord*cm, ycoord*cm);
endfor;
asemicPath
enddef;
```

Code has meaning (on and on)

```
%generates an asemic path from a closed path.
 vardef nonLineAsemic(expr aPath,nPoints,minii,maxii)=
path aasemicPath;
aasemicPath := ((((xpart point 0cm on aPath)/72)*2.54)*cm, (((ypart point 0cm
on aPath)/72)*2.54)*cm);
mini := minii; maxi := maxii; nbPoints := nPoints;
for x = 1 upto nbPoints:
    ycoord := ((ypart point(x/nbPoints) along aPath)/72)*2.54
              + randnumRange(mini,maxi);
    xcoord := ((xpart point(x/nbPoints) along aPath)/72)*2.54
              + randnumRange(mini,maxi);
    aasemicPath := aasemicPath ... {curl 100}(xcoord*cm, ycoord*cm);
endfor;
aasemicPath := aasemicPath ... cycle;
aasemicPath
enddef;
```

How to improve the code?

- No need for for two macros to calculate the asemic path. We check using an if statement to see if the begining of the original path is the same as the end of the original path. If it is, we cycle the asemic path and return it, otherwise we just return the asemic path.

```
if ((point 0 along aPath) = (point nbPoints along aPath)):
    aasemicPath := aasemicPath ... cycle;
fi;
    aasemicPath
```

- Change the def ColoredPath to scale the pen circle in both the x and y direction.

- Create a background colour for the text areas of the images only.

How to improve the code? Cont.

- Each MPgraphic page has essentialy four groups of code.
  - Creating and drawing the asemic paths.
  - Calculating where points on the asemic paths are to be connected.
  - Drawing the interconnecting lines.
  - Reflect the MPgraphic on the recto page.
- There is a lot of repetion of code which could probably converted to vardefs or defs, however it is important maintain readability and understanding of code.

How to improve the code? Cont.

- Create arrays for the various kinds of paths e.g. lPath[i] for linear or open paths, cPath[i] for circular paths, sPath[i] for square paths and so on. Adding an a as in asPath[i] means it's the asemic path of the square path. Cuts down on variables.
- It would be nice to just draw the returned asemic path however we need this be given a variable name since it's required when calculating points along the asemic path for the interconnecting lines later.
- If paths are named using arrays could the calcuted points be named similarly?
  alongasa[i] = point randnumRange(0.75,0.68) along asaPath; become
  alongas1[i] = point randnumRange(0.75,0.68) along asPath[1];
- Similarly, with drawing the paths between the calculated points above?

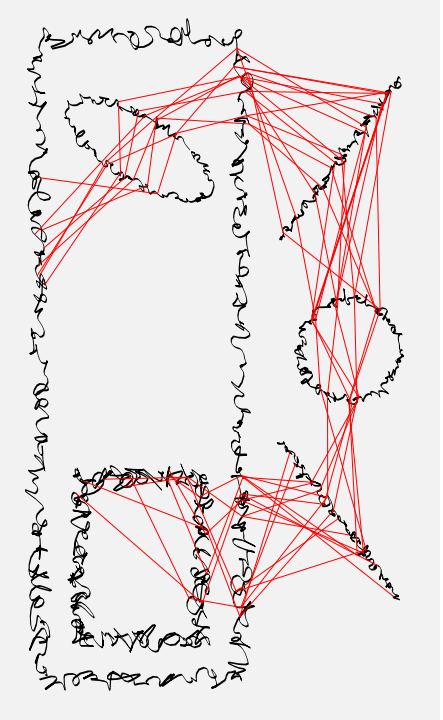
How to improve the code? Cont.

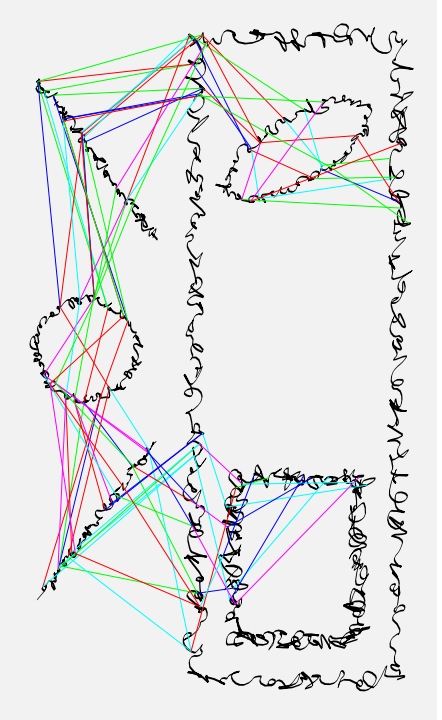
- I should learn the about the variables available in MetaFun
  - Between the StartPage and StopPage command you have access to a wide range of variables, which are usefull in positioning calculations. Here are some examples:

page	PaperHeight	PaperWidth
	PrintPaperHeight	PrintPaperWidth
	PageOffset	PageDepth
margins	TopSpace	BackSpace
text	MakeupHeight	MakeupWidth

The MetaFun manual has more, see chapter 6, particularly Section 6.4.

Example pages from the book





How to improve the code? Cont.

- Each MPgraphic page has essentialy four groups of code.
  - Creating and drawing the asemic paths.
  - Calculating where points on the asemic paths are to be connected.
  - Drawing the interconnecting lines.
  - Reflect the MPgraphic on the recto page.
- There is a lot of repetion of code which could probably converted to vardefs or defs, however it is important maintain readability and understanding of code.

Drawing the pages.

```
\starttext
%clipped
\dorecurse{2}{
\useMPgraphic{page1}
}
\dorecurse{2}{
\useMPgraphic{page2}
}
%clipped
\dorecurse{2}{
\useMPgraphic{page24}
}
\dorecurse{2}{
\useMPgraphic{page25}
}
%clipped
\stoptext
```

Creating and Drawing the asemic paths

```
\startuseMPgraphic{page15}
fill Field[Text][Text] withcolor 0.95white;
xstart := ((xpart llcorner Field[Text][Text])/72)*2.54;
ystart := ((ypart llcorner Field[Text][Text])/72)*2.54;
path saPath, asaPath,sbPath, asbPath, caPath, acaPath, cbPath, acbPath,laPath,
alaPath, lbPath, albPath;
pair alongasa[], alongasb[],alongaca[], alongacb[], alongala[], alongalb[];
saPath := unitsquare xscaled(xstart + 5cm) yscaled(ystart + 16cm) shifted((xstart
+ 1)*cm,(ystart + 1)*cm);
asaPath := Asemic(saPath,300,-0.2,0.2);
draw asaPath withpen pencircle xscaled 0.15mm yscaled 0.5mm rotated 30;
%
caPath := unitcircle scaled 2.5cm slanted -1 shifted((xstart + 3.5)*cm,(ystart
+ 13)*cm);
acaPath := Asemic(caPath,100,-0.1,0.1);
draw acaPath withpen pencircle xscaled 0.15mm yscaled 0.5mm rotated 30;
%
laPath := (((xstart + 7)*cm,(ystart + 12)*cm) -- ((xstart + 10)*cm,(ystart +
16)*cm)) rotatedaround(((xstart + 7)*cm,(ystart + 12)*cm),0);
alaPath := Asemic(laPath,75,-0.1,0.1);
draw alaPath withpen pencircle xscaled 0.15mm yscaled 0.5mm rotated 30;
%
```

Calculating where points on the asemic paths are to be connected.

```
for i = 0 step 1 until 7:
       alongasa[i] = point randnumRange(0.75,0.68) along asaPath;
       alongasa[(i + 8)] = point randnumRange(0.5,0.45) along asaPath;
       alongasa[(i + 16)] = point randnumRange(0.275,0.125) along asaPath;
       alongasb[i] = point randnumRange(0.75,0.5) along asbPath;
       alongasb[(i + 8)] = point randnumRange(0.5,0.25) along asbPath;
       alongasb[(i + 16)] = point uniformdeviate(0.3) along asbPath;
       alongaca[i] = point uniformdeviate(0.25) along acaPath;
       alongaca[(i + 8)] = point randnumRange(0.8,0.4) along acaPath;
       alongacb[i] = point randnumRange(0.375,0.125) along acbPath;
       alongacb[(i + 8)] = point randnumRange(0.875,0.625) along acbPath;
       alongala[(i)] = point uniformdeviate(1) along alaPath;
       alongala[(i + 8)] = point uniformdeviate(1) along alaPath;
       alongalb[(i)] = point uniformdeviate(1) along albPath;
       alongalb[(i + 8)] = point uniformdeviate(1) along albPath;
```

Drawing the interconnecting lines.

```
for j= 0 step 1 until 7:
if \recurselevel = 1:
    draw alongasa[(j)] -- alongaca[(j + 8)] coloredPath(0.25mm)red;
    draw alongaca[(j)] -- alongaca[(j + 8)] coloredPath(0.25mm)red;
    draw alongaca[(j)] -- alongasa[(j + 8)] coloredPath(0.25mm)red;
    draw alongasa[(j + 8)] -- alongala[(j)] coloredPath(0.25mm)red;
    draw alongasa[(j + 8)] -- alongala[(j + 8)] coloredPath(0.25mm)red;
    draw alongala[j] -- alongacb[j] coloredPath(0.25mm)red;
    draw alongala[(j + 8)] -- alongacb[j] coloredPath(0.25mm)red;
    draw alongacb[j] -- alongacb[(j + 8)]coloredPath(0.25mm)red;
    draw alongacb[(j + 8)] -- alongalb[(j)]coloredPath(0.25mm)red;
    draw alongalb[(j + 8)] -- alongasa[(j + 16)]coloredPath(0.25mm)red;
    draw alongalb[(j)] -- alongasa[(j + 16)]coloredPath(0.25mm)red;
    draw alongasa[(j + 16)] -- alongasb[(j + 8)] coloredPath(0.25mm)red;
    draw alongasb[(j + 8)] -- alongasb[(j)] coloredPath(0.25mm)red;
```

Drawing the interconnecting lines. (cont)

```
else:
draw alongasa[(j)] -- alongaca[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongaca[(j)] -- alongaca[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongaca[(j)] -- alongasa[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongasa[(j + 8)] -- alongala[(j)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongasa[(j + 8)] -- alongala[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongala[j] -- alongacb[j]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongala[(j + 8)] -- alongacb[j]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongacb[j] -- alongacb[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongacb[(j + 8)] -- alongalb[(j)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongacb[(j + 8)] -- alongalb[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongalb[(j + 8)] -- alongasa[(j + 16)
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
```

```
draw alongalb[(j)] -- alongasa[(j + 16)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongasa[(j + 16)] -- alongasb[(j + 8)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
draw alongasb[(j + 8)] -- alongasb[(j)]
    coloredPath(0.25mm)c[floor(uniformdeviate(5))];
fi;
endfor;
```

Reflect the MPgraphic on the recto page.

```
th := Vsize[Text]/72/2.54;
tw := Hsize[Text]/72/2.54;
if \recurselevel = 2:
       picture pic;
       pic := currentpicture;
       currentpicture := nullpicture;
       draw pic
        reflectedabout(((tw/2)*cm,(ystart + 1)*cm), ((tw/2)*cm,(th - 1)*cm));
fi;
\stopuseMPgraphic
```

Part Two

- There is more than one way to skin a cat! (Randomness revisited.)
- We have asemic lines, what about words?
- (Lua)MetaFun is really fun! Envelopes and paths.
- Some examples of my asemic stuff.
- Finally! Lunch.

Randomness revisited

- The for loop I've coded for creating asemic paths is (relatively) slow. Here is another way suggested by Hans.

```
vardef randomiser(expr p, mini, maxi) =
    (xpart p + randnumRange(mini, maxi)*cm,
     ypart p + randnumRange(mini, maxi)*cm)
enddef;
vardef Asemic(expr aPath, nPoints, mini, maxi)=
       path aasemicPath; numeric nbPoints;
    nbPoints := nPoints;
    aasemicPath := aPath;
    aasemicPath := arcpointlist nbPoints of aasemicPath;
    aasemicPath :=
        for i within aasemicPath:
          randomiser(pathpoint, mini, maxi)
          .. tension 2.5 ..
        endfor
        nocycle
        ;
    aasemicPath
enddef;
```

Randomness revisited

- I had a rethink about the *randomized* command and created another asemic path vardef.

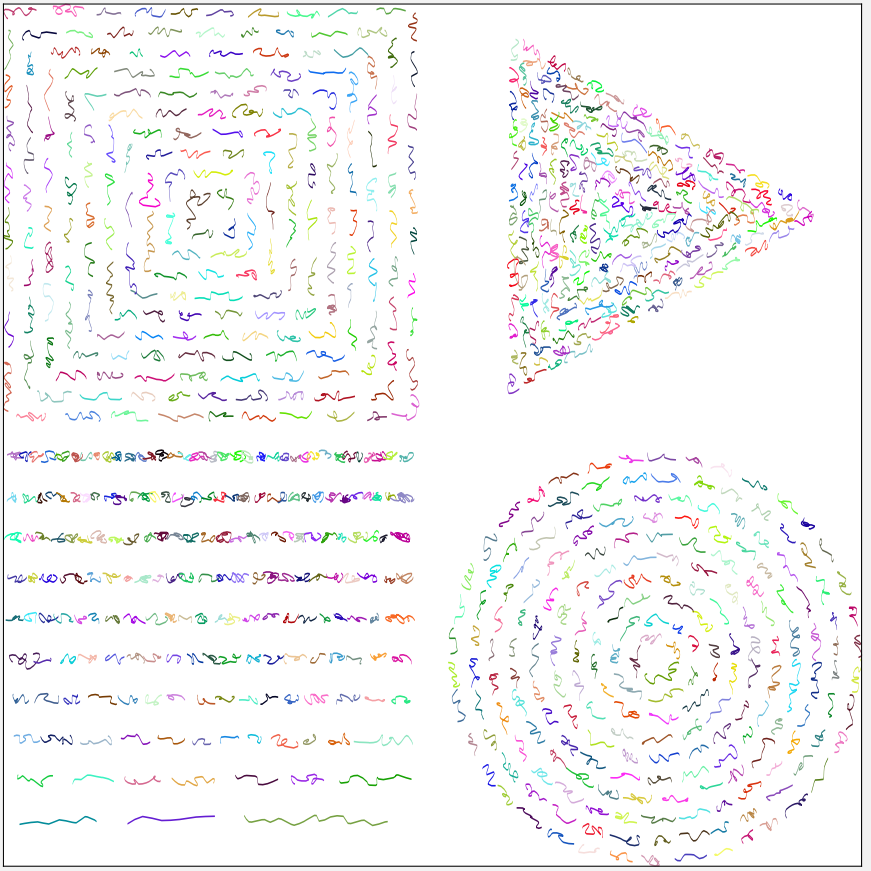
```
vardef Aasemic(expr aPath, nPoints, mini, maxi)=
       path aasemicPath; numeric nbPoints;
    nbPoints := nPoints;
    aasemicPath := aPath;
    aasemicPath := arcpointlist nbPoints of aasemicPath;
    aasemicPath :=
        for i within aasemicPath:
          pathpoint randomized randnumRange(mini*4cm, maxi*4cm)
          .. tension 2.5 ..
        endfor
        nocycle
        ;
    aasemicPath
enddef;
```

There are other ‘randomizing’ commands in metafun which could also be used within the vardef above e.g. *randomizedcontrols*. See the source code.

Asemic words

- Asemic words are a sequence short asemic paths on a line separated by spaces. It's just like text on a page.
- We need to decide on the line length, the number of characters in a line, the maximum and minimum number of characters in a word, the number of spaces between the words and ensure the last word does not end outside the line (justified).
- Number of characters in a line = nbPoints. ca. 100
  Number of character in a word = wordlength := round(randnumRange(minword,maxword))
  Number of charaters in a space = spaces := 3;

Asemic words



Asemic words code

```
for i = 10 downto 1:
    %Four Paths
    for j = myPatha, myPathb, myPathc, myPathd:
    ending := false;%boolean
    wordlength := round(randnumRange(minword,maxword));
    beginPath := 1;
    endPath := beginPath + wordlength;
    forever:
        if (nPoints - beginPath) < 18:
            ending := true;
            endPath := nPoints - 1;
            t := (beginPath/nPoints) * length j;
            u := (endPath/nPoints) * length j;
            asubPath := subpath(t,u) of j;
            wordlength := (nPoints - 1) - beginPath;
            draw Asemic(asubPath,wordlength,mini,maxi) coloredPath(.1mm,.4mm);
        else:
            ending := false;
            t := (beginPath/nPoints) * length j;
            u := (endPath/nPoints) * length j;
            asubPath := subpath(t,u)  of j;
            draw Asemic(asubPath,wordlength,mini,maxi) coloredPath(.1mm,.4mm);
            beginPath := endPath + spaces;
```

```
            endPath := beginPath + wordlength;
            wordlength := round(randnumRange(minword,maxword));
        fi;
        exitif ending = true;
    endfor;
    endfor;
endfor;
```

Fun with Paths and Envelops!

- What are Envelops? It's complicated, see Chapter 20 *Envelops*, of the (Lua)MetaFun Manual.
- '*An envelop is the outline that we get when we run a pen over a path. An envelop is (of course) a closed path.*' from Chapter 20.3 of the (Lua)MetaFun manual.

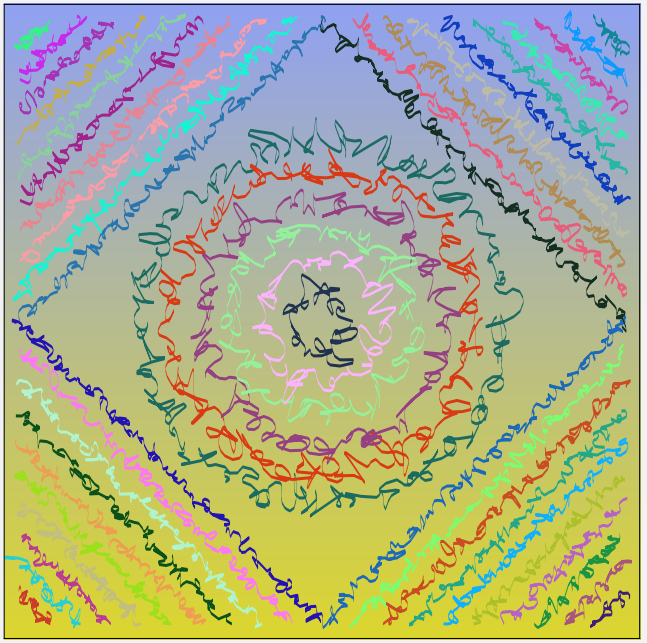
Fun with Paths and Envelops! Continued.

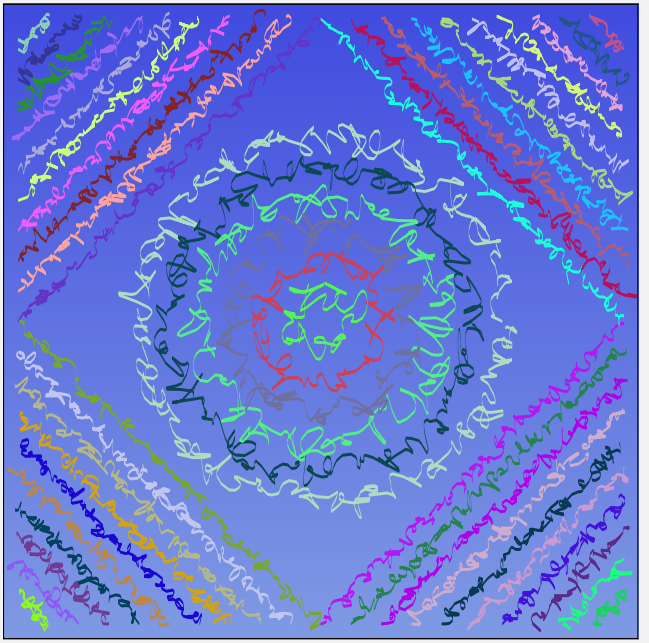
Example

```
asubPath:= Asemic(asubPath,wordlength,-0.25,0.25);
asPathEnvelope := envelope pensquare scaled .5mm rotated 45 of asubPath;
definecolor [name = "Mycolor1", r = uniformdeviate(1), g = uniformdeviate(1),
b = uniformdeviate(1)];
definecolor [name = "Mycolor2", r = uniformdeviate(1), g = uniformdeviate(1),
b = uniformdeviate(1)];
fill asPathEnvelope yscaled 1 withshademethod "linear"
                              withshadecolors ("Mycolor1","Mycolor2");
```

I have used them in asemic witing to create some nice effects.

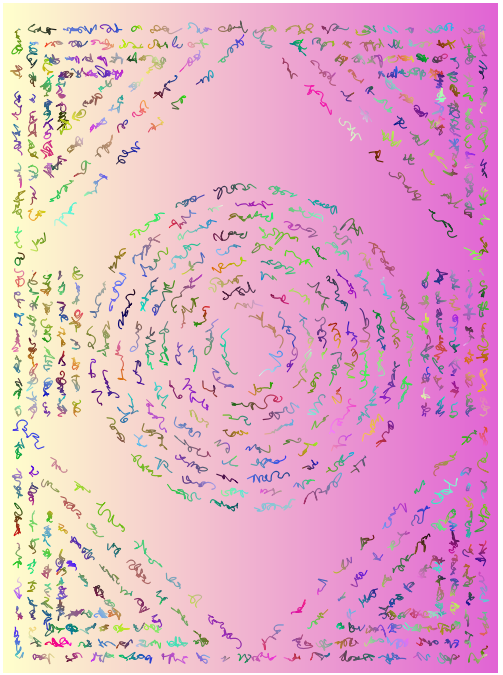
Computer generated asemic writing examples

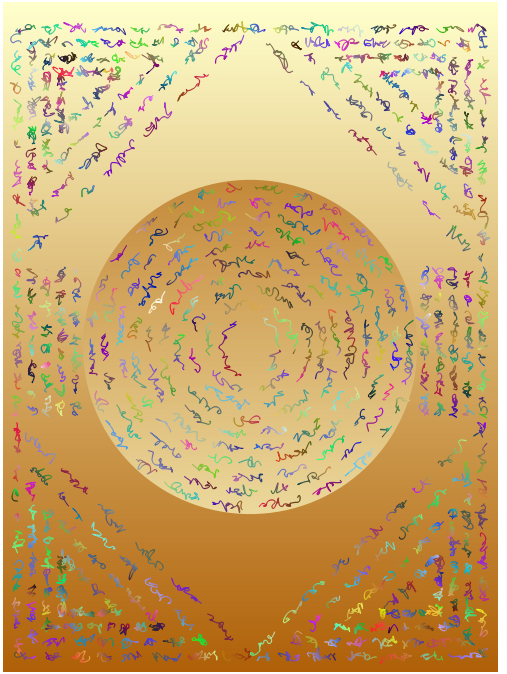




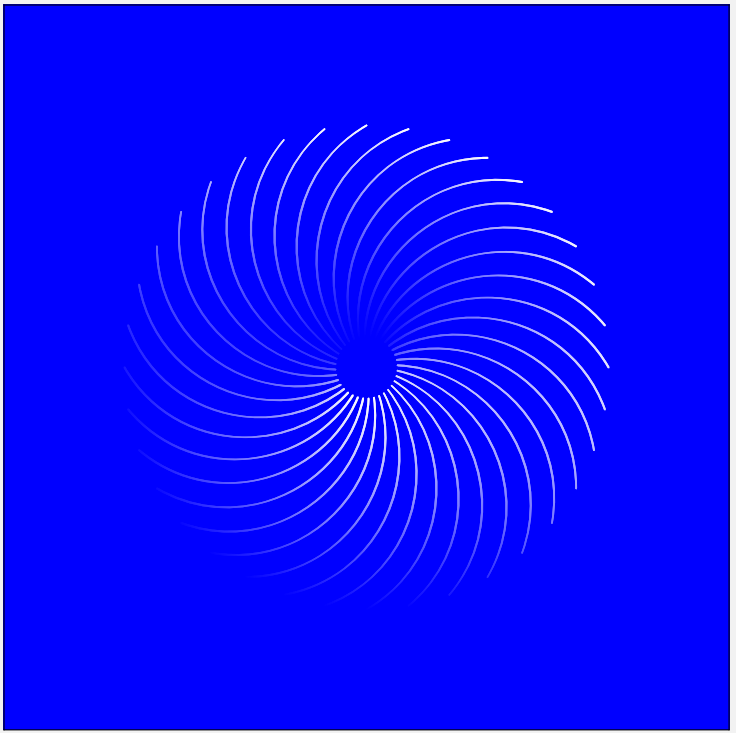
Computer generated asemic writing examples

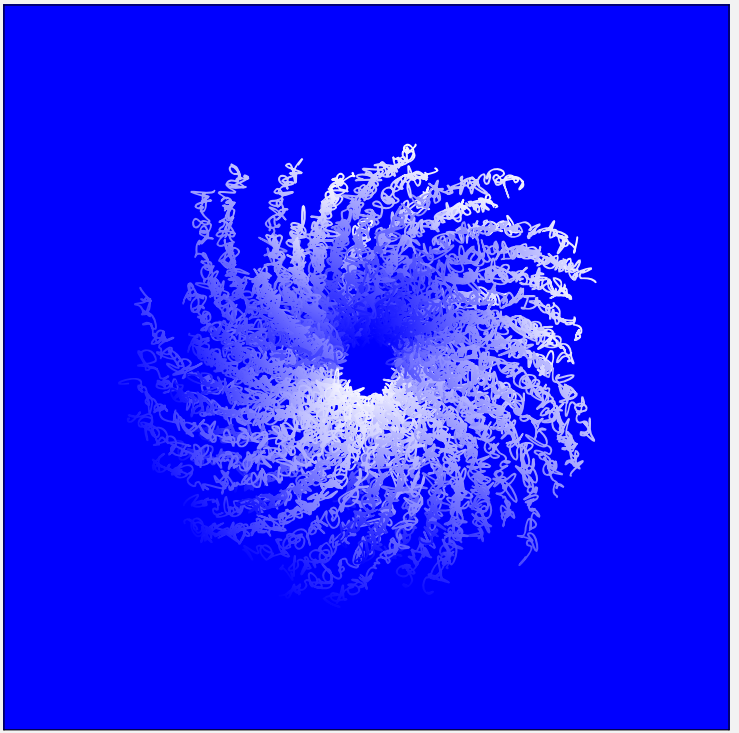
Computer generated asemic writing examples



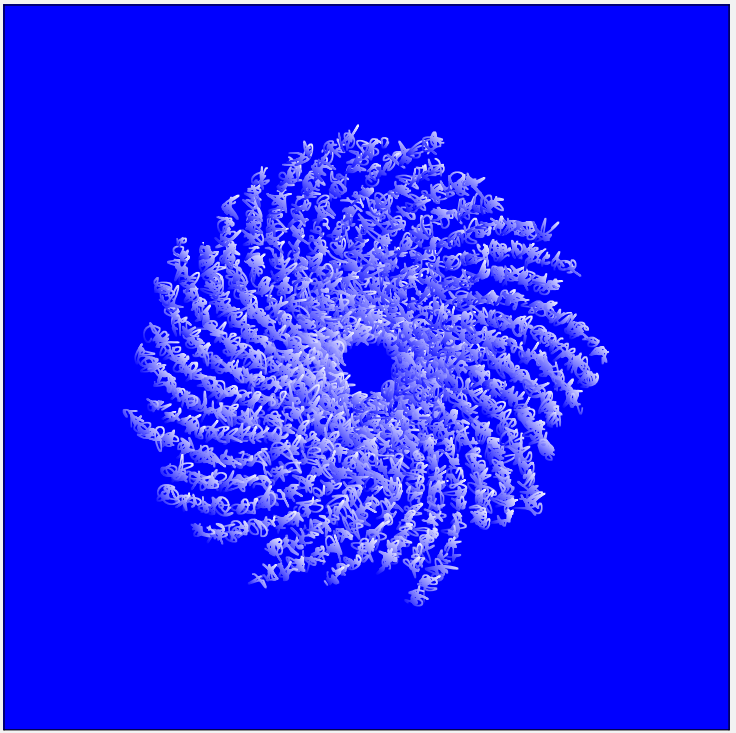


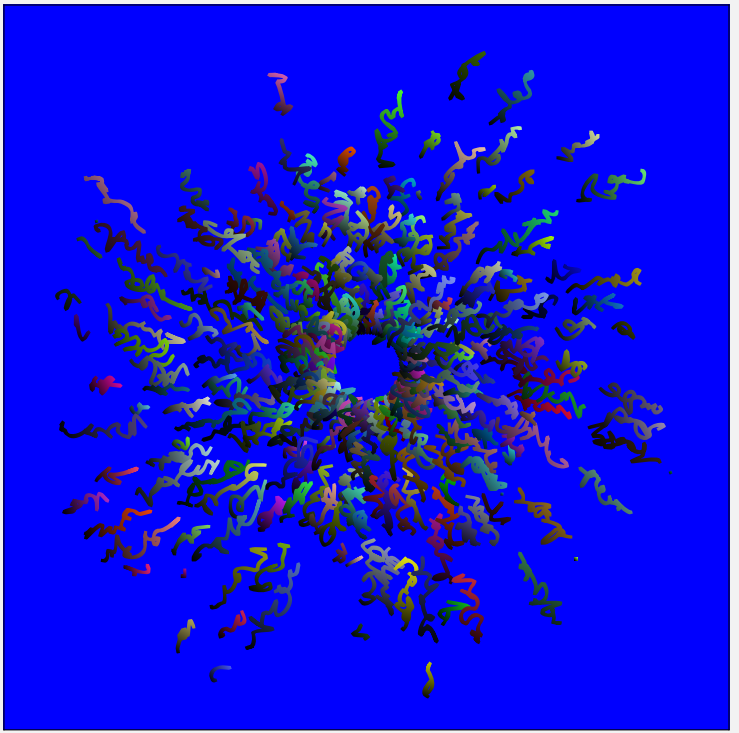
Computer generated asemic writing examples



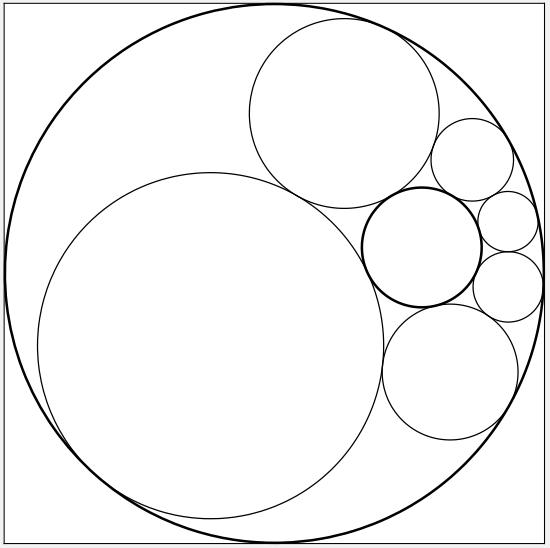


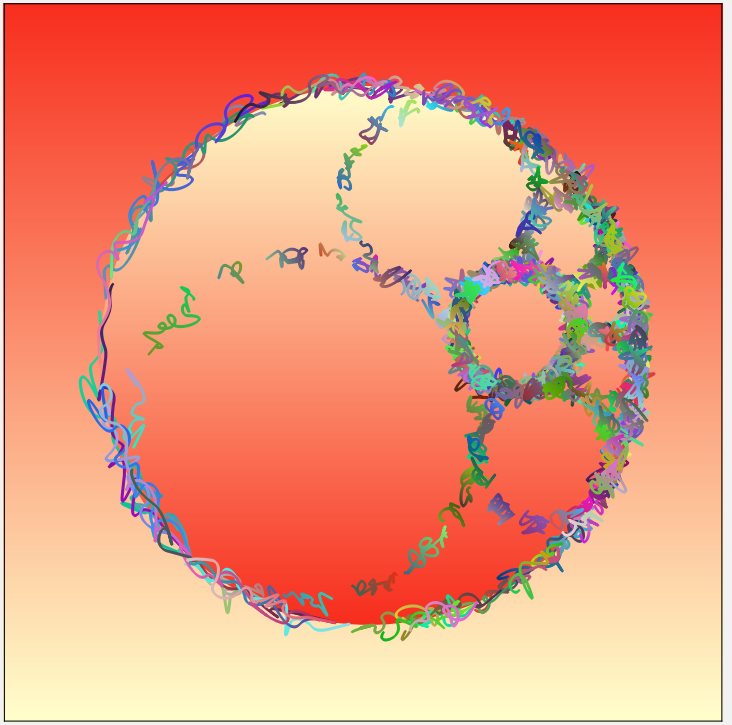
Computer generated asemic writing examples



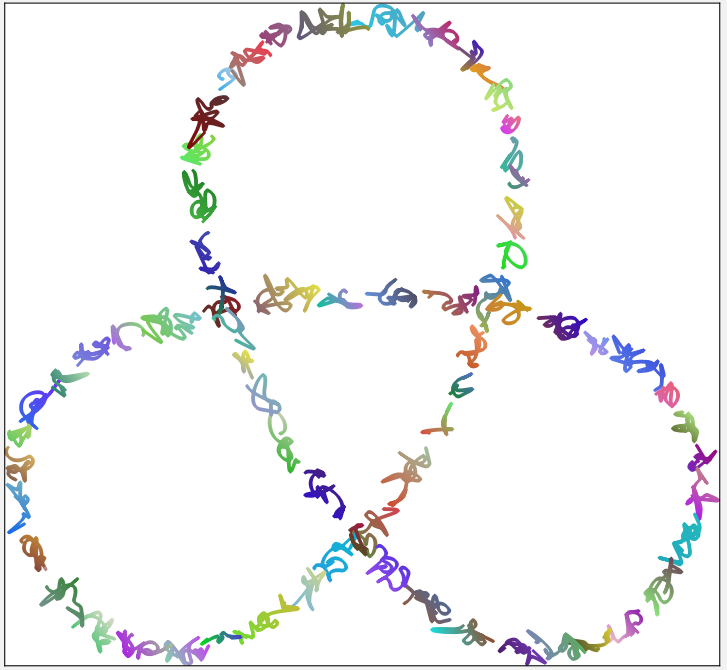


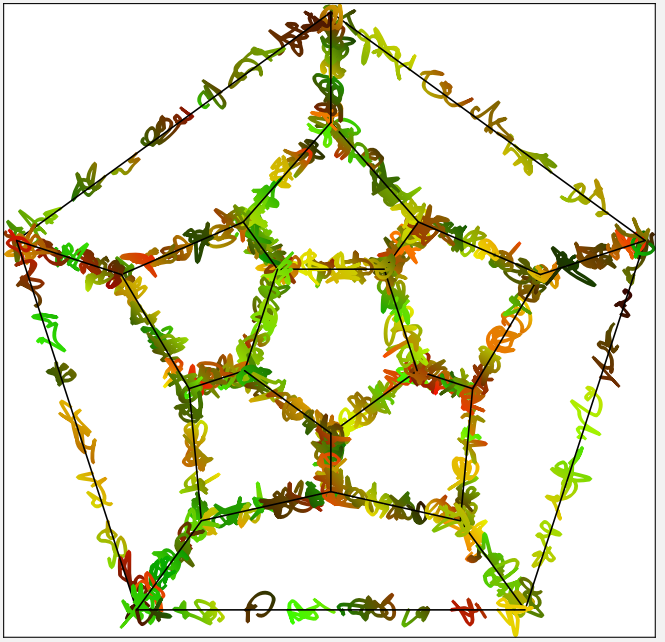
Computer generated asemic writing examples



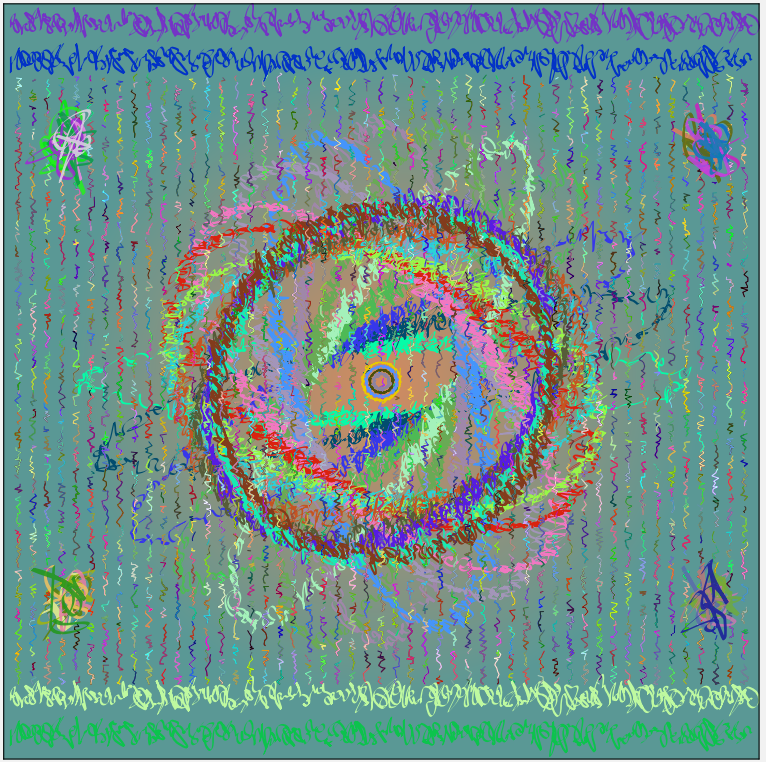


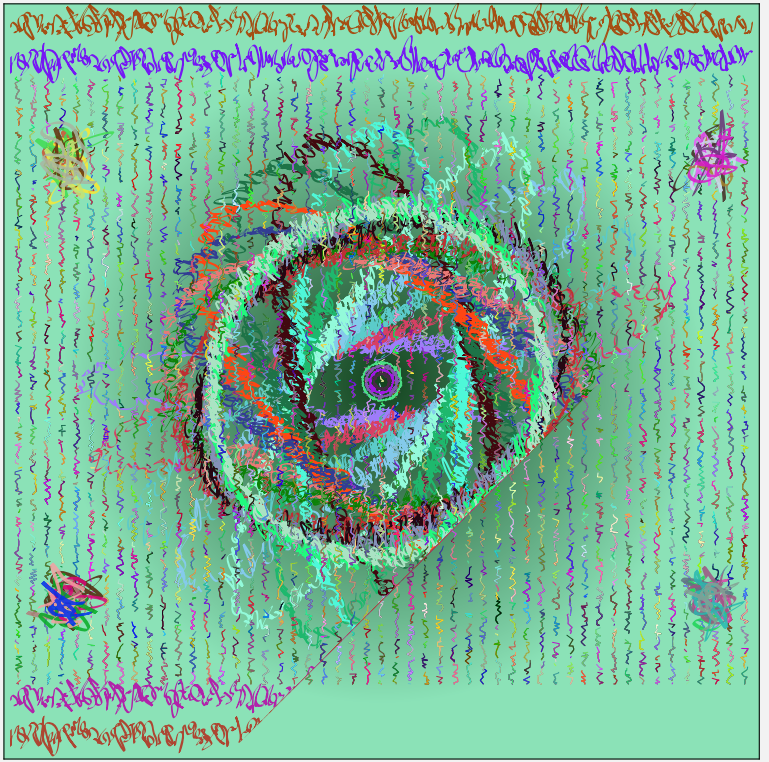
Computer generated asemic writing examples



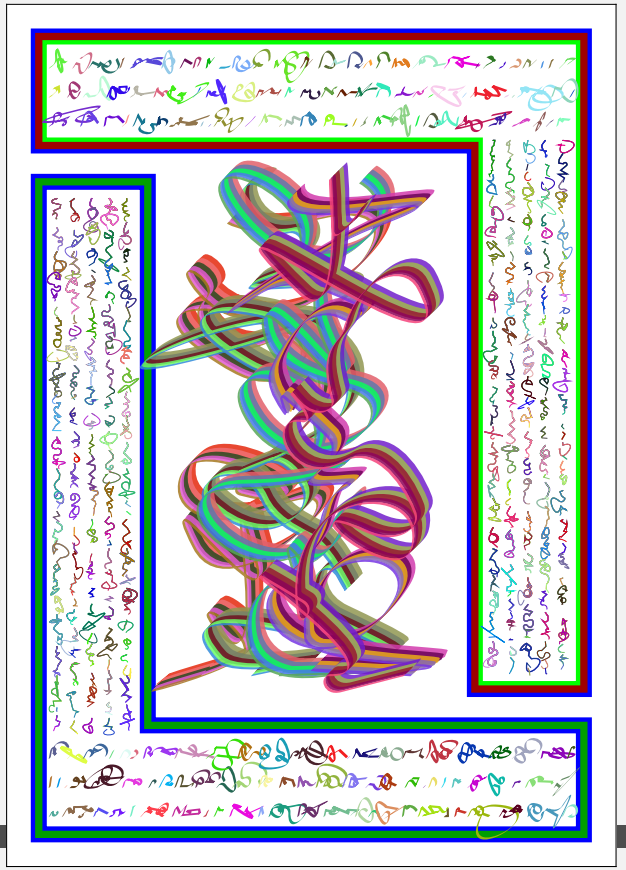


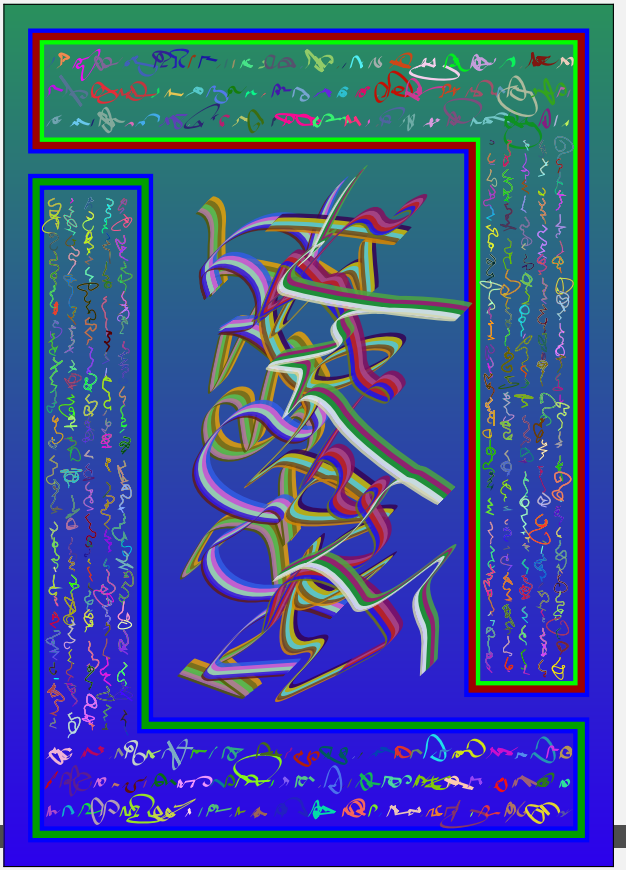
Computer generated asemic writing examples



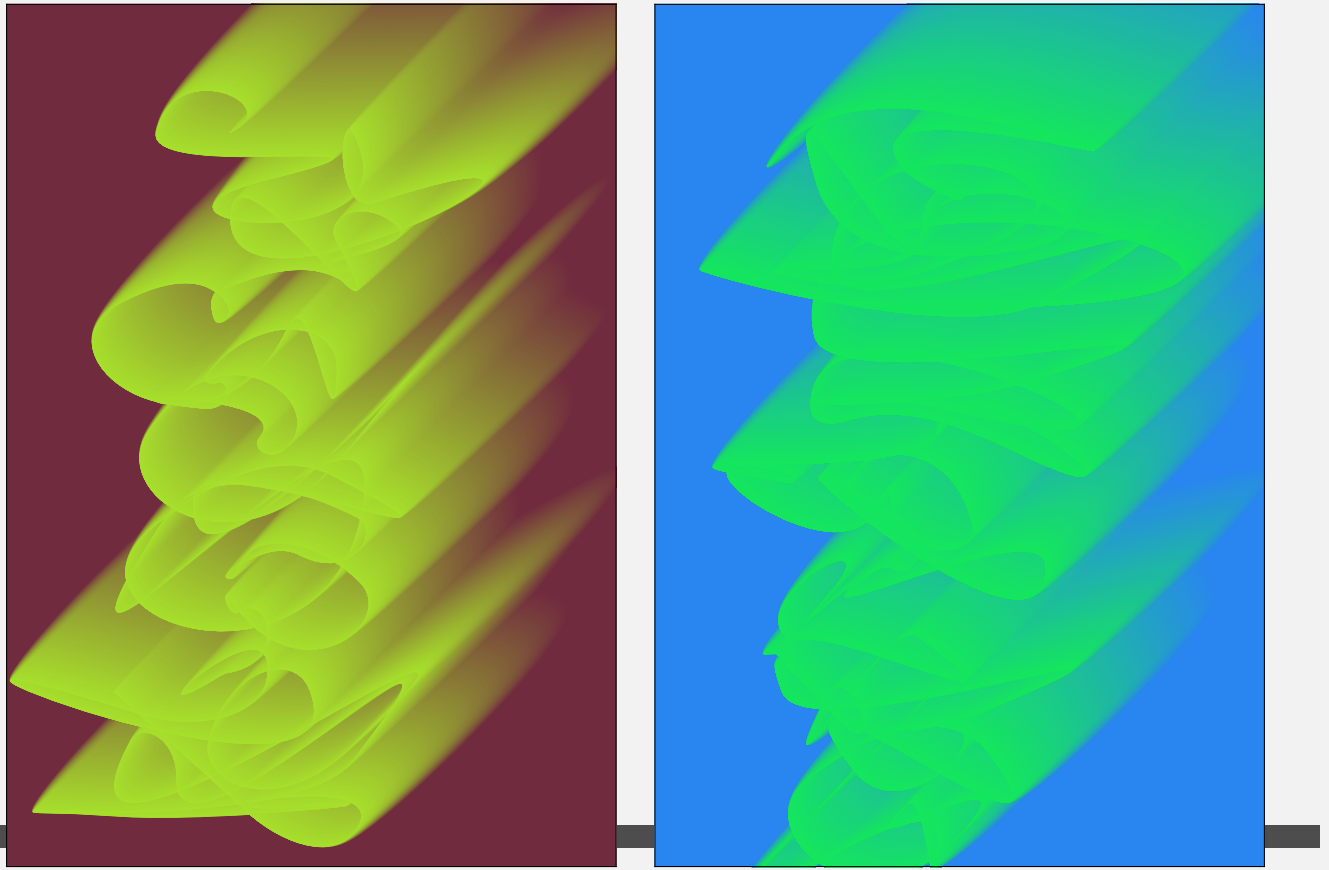


Computer generated asemic writing examples

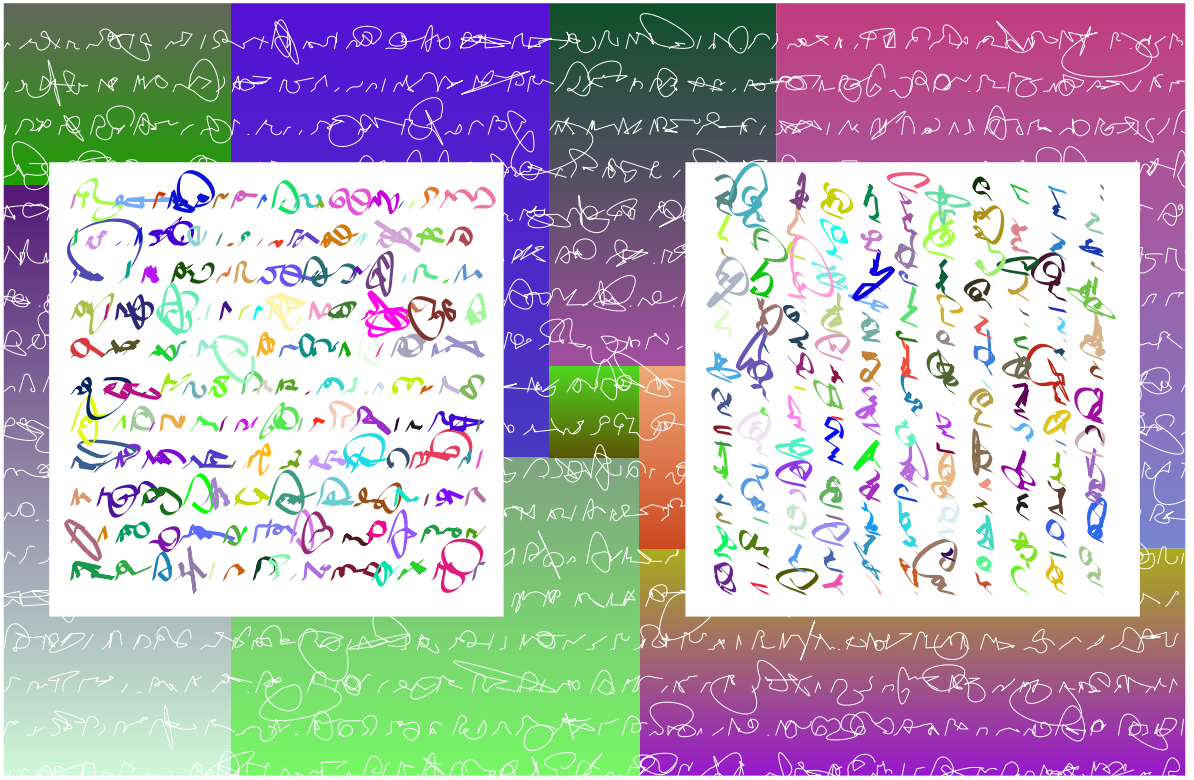




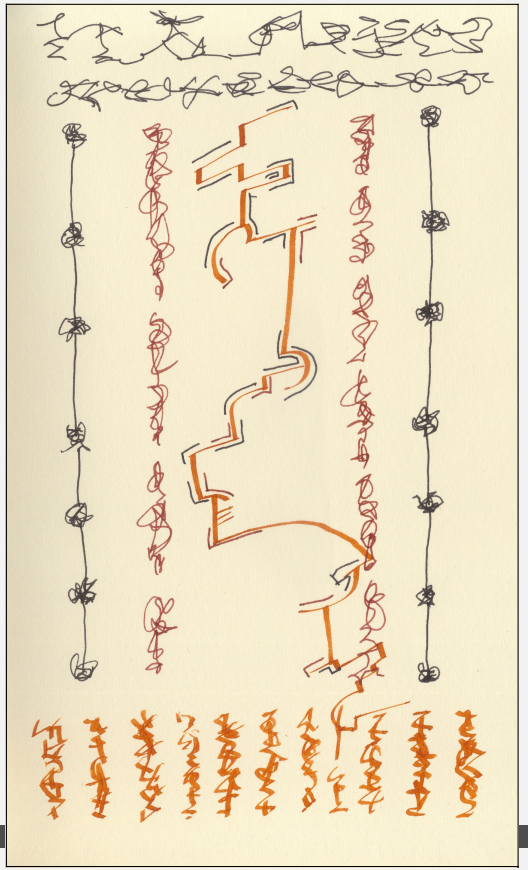
Computer generated asemic writing examples

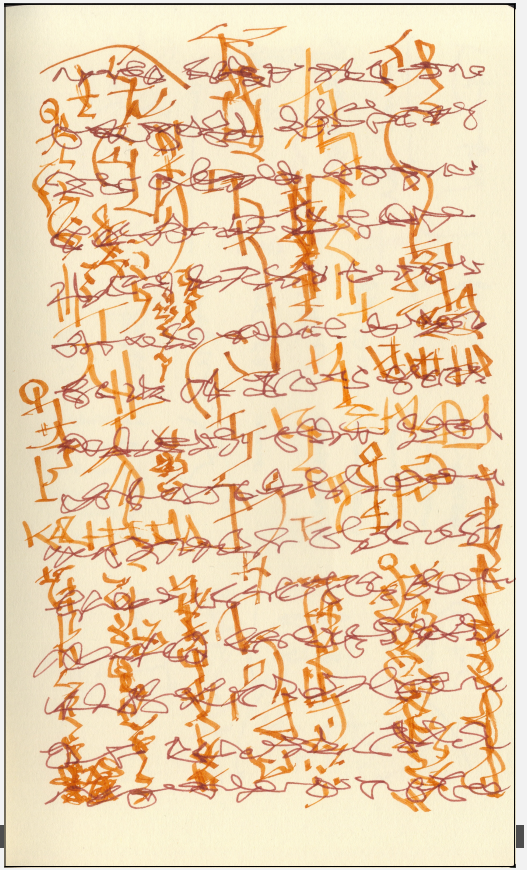


Computer generated asemic writing examples



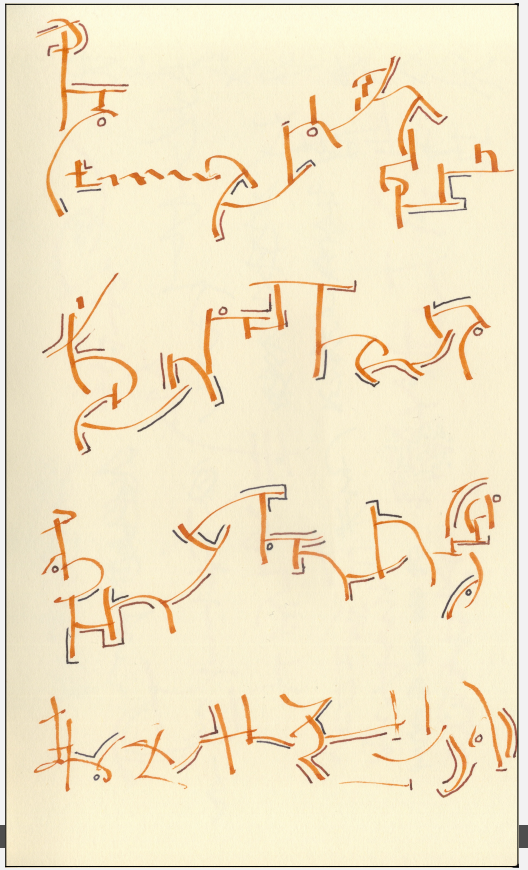
Hand written asemic writing examples

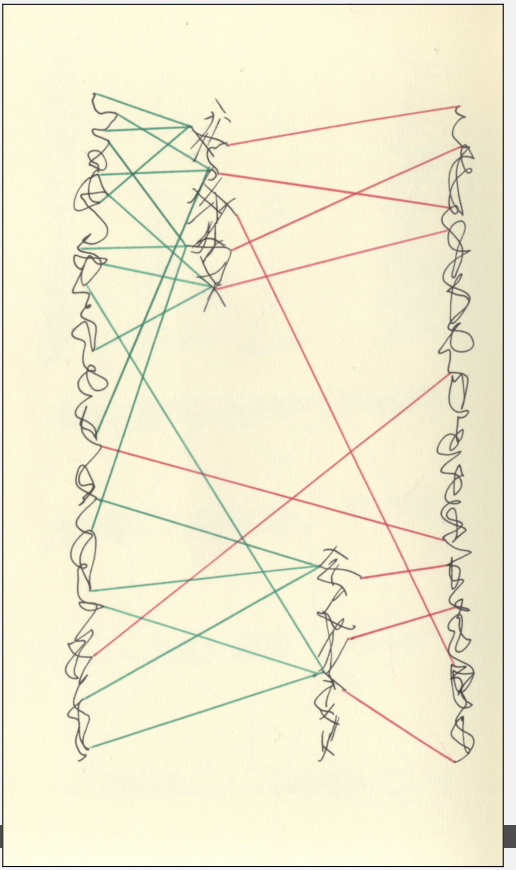


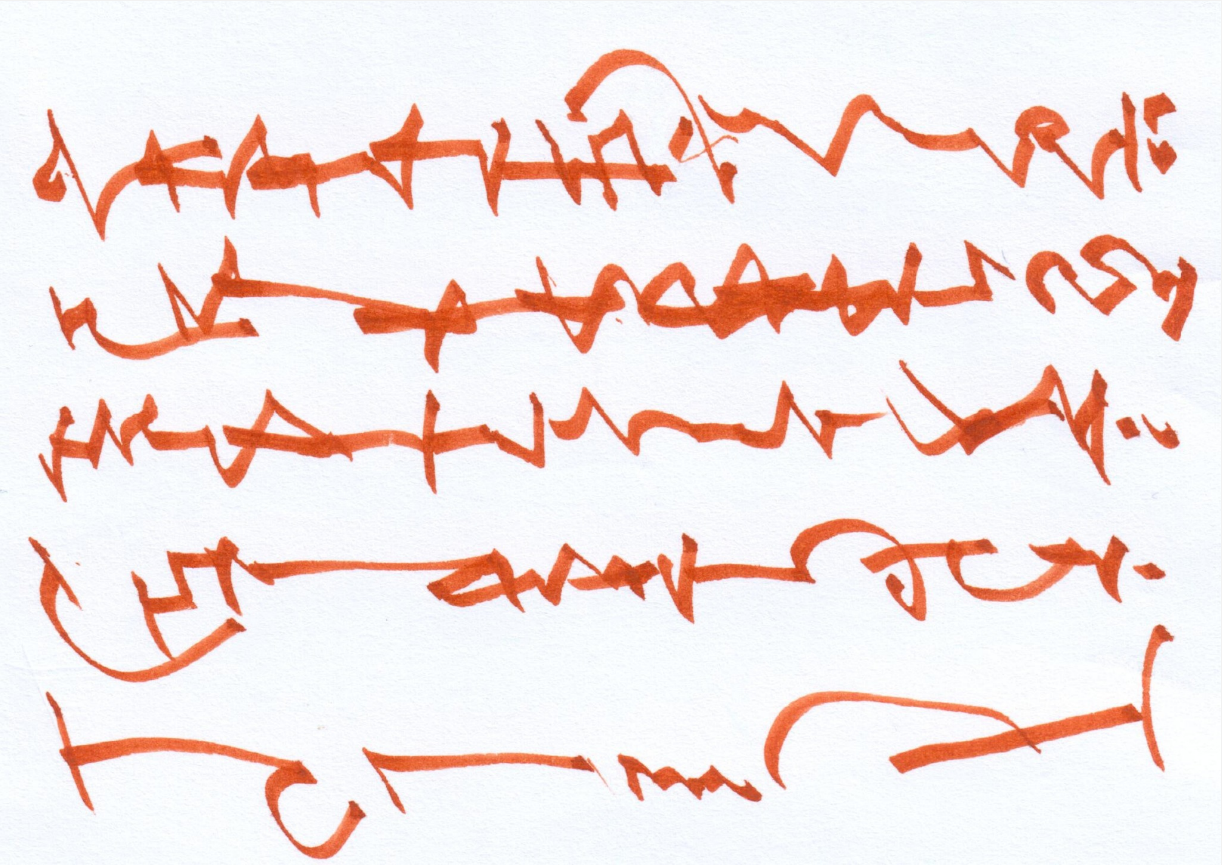


Hand written asemic writing examples

Hand written asemic writing examples







Further Reading

- AsemicDiagrams: computer Generated Asemic Drawings by Keith McKay
- Circles and Squares: One page Asemic Poems by Keith McKay
- An Anthology of Asemic Writing edited by Time Gaze and Michael Jacobson
- Asemic The Art of Writing by Peter Schwenge
- The Selected Scribbling and Scrawling of SJ Fowler by SJ Fowler
- MetaFun for Generative Art by Fabrice Larribe, MAPS 52
- Chapter 7 in the On Target manual (in the distribution)
- MetaPost, MetaFun, and LuaMetaFun Manuals