code::art::1

code::art

code::art is an art journal which publishes code.

The intent of the journal is to further explore the relationship between code and art, and to challenge our perceptions of what both can look like.

You can find code::art online at https://code-art.xyz, or on Twitter @codeart_journal. Get in touch by email at codeartjournal@gmail.com.

Issue #1

Gantt charts as sprawling monstrosities, fortifications built in Swift code, instructions on how to never ask out your crush, what may be the world's worst (best?) Hello World program, C interpretations of Dali. All of these and more await you in this collection of code-as-art.

As with the inaugural issue, I was delighted by the imaginative pieces which people submitted and the huge variety of forms and mediums which the artists employed.

I hope that these works will challenge and amuse you, and that you'll consider submitting work of your own for issue #2!

Sy Brand

Editor

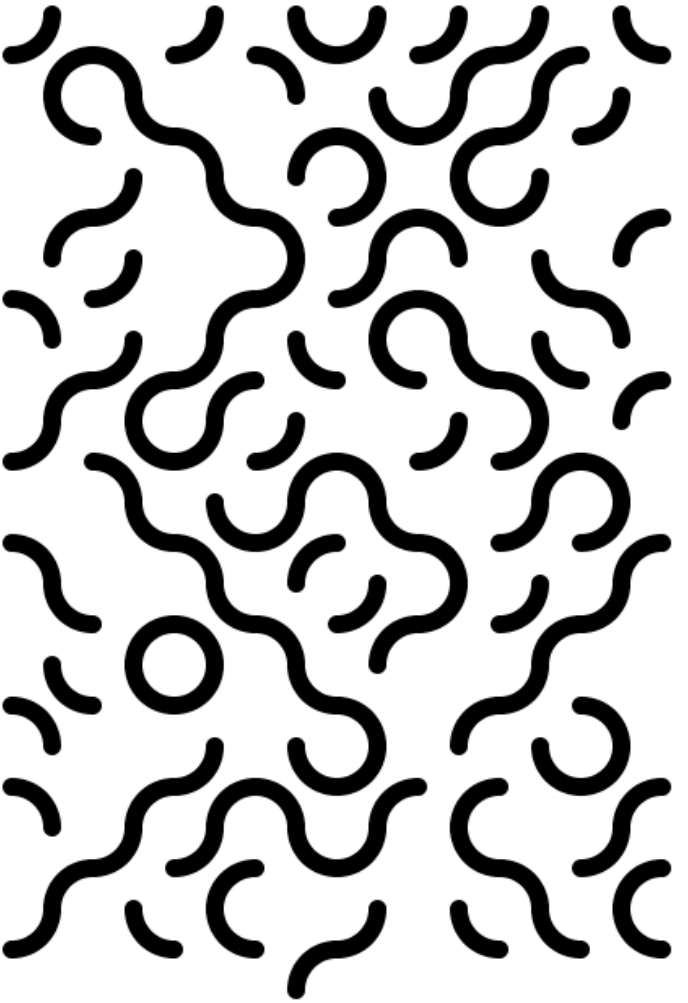
journal: code::art

issue: 1

contents:


ENIGMA
SCHLÜSSEL
Kenngruppen

```
void the_endless_cycle(person& i, person const& they)
{
  if (i.want(they)
      && i.appreciate(they)
      && i.respect(they)
      && i.love(they))
  {
    // but ...
    if (!they.want(i)
        && !they.appreciate(i)
        && !they.respect(i))
        // !they.love(i), no need to check
    {
      person const& other = i.await_for();
      the_endless_cycle(i, other);
      return;
    }
    else
    {
      assert(false); // todo: implement
    }
  }
}
```



```
Joel’s.apology


func == (life: Hers, lives: [Other]) -> Bool {
  guard
    you’re ≠ judge(to: choose(life, or: lives)),
    she ≠ judge(to: choose(life, or: lives)),
    choose(life, or: lives) == (us.humanity == nil)
  else { preconditionFailure() }
  return true
}

func ≠ (people: [Human], worth: Saving) -> Bool {
  guard
    people + fungus == downfall,
    what(we, became) == animals,
    her.sacrifice ≠ worth(people)
  else { preconditionFailure() }
  return true
}

func > (love: Love, _: Any) -> Bool {
  guard I.couldn’t(loose:her:)(not, again)
  else { fatalError() }
  return true
}
```

```
chimeras                                [tribute(to: tim.hecker)]


extension Gatherable {
  func welcome(gatherer: inout Gatherer) {
    gatherer.gather(from: self)}
}

extension Chimeras: Gatherer {

  func gather(from: Streets) {
     `let`(the: car.doors(closing)
     .counterpoint(the: drones),
      see(the: centrepiece(of: handlebars)
     .that(corrosion.adorns)))}

  func gather(from: Fields) {
     `let`(the: seedlings.take(over:
      steel, and: leather).alike)
  -- (bare.feet && bare.sounds)
     .pulverise(just: fine)}

  func gather(from: Homes) {
     `let`(the: weathered.glow
     .paralyse(the: spheres,
      of: still.lifes, and: sharp.edges
     .left(without)(a,`guard`)))}
}

Streets().welcome(&chimeras)
Fields().welcome(&chimeras)
Homes().welcome(&chimeras)

then {

         watch(as: they.relinquish(the: chitinous.shells),
        butterfly(their: arms), and: disclose(the: axons),
     as: (the: two(kinds).of(waves).interweave(to: amber),
 cells.mixed(in: togetherness, long.lost, but: now.found)))}
```

```
fortifications


{

    I’ve.built

   ┌──────────────────────────────────────────────┐
  [                                                ]
  [                                                ]
  [ fortifications.around(my: brain)               ]
  [ cerebrospinal<fluid>.fills(the: moat)          ]
  [ neurones.power(the: electric, fence)           ]
  [                                                ]
  [ and.from(the: bastions, of(the: lobes))        ]
  [ I(desperately).gaze(at, the: horizon)          ]
  [ wish.not(for:the:enemy:) - but(for: anyone)    ]
  [                                                ]
  [ to.tell(me)                                    ]
  [                                                ]
  [              it’s.all.gonna.end                ]
  [                                                ]
  [                                                ]
   └──────────────────────────────────────────────┘


                     well
                            soon

                                                  }
```

The Forest

noise intoxicates
way with one

hunter hunts
lifts knife
rather with away,
where lift rather pendulum

who with one lift

feeds the forest
thirst rather key one dangles
who feeds
with flow
thirst of the forest
intoxicates the hunted
forest tight

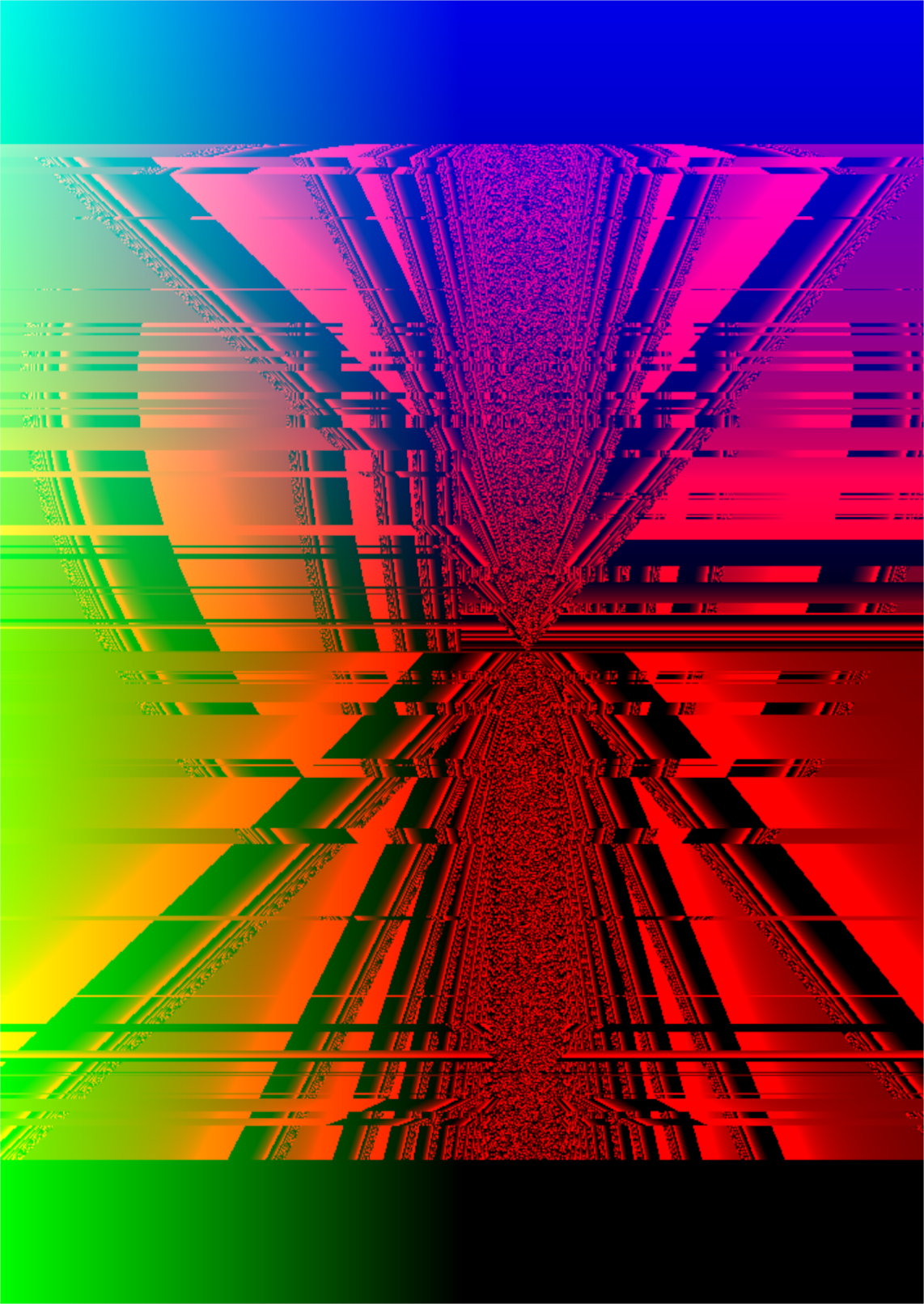
hunt way
dangle knife one hunger
where feeds the hunted
hearse cap pendulum dangle,
away with the forest,

way with one
away with the knife

flow pendulum
away you flow,,
pendulum way

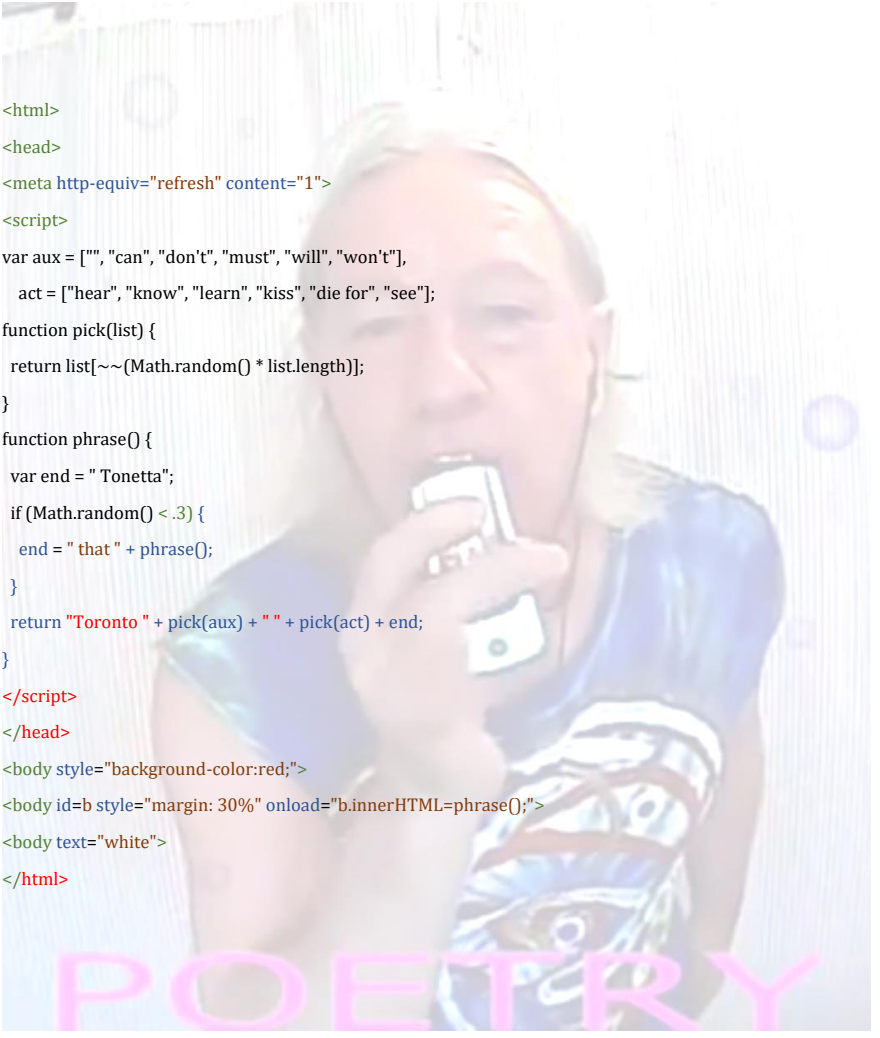
who would we rather?
pendulum
where,
who feeds the numb hunter

lift on flow
one who flows



```
// Lockdown Dérive
var WellBalancedDay = function() {
  this.todo = [];
  for (var action in URGENT_TASKS) {
    this.todo.push(URGENT_TASKS[action]);
  }
};
setInterval(function() {
  let today = new WellBalancedDay();
  for (var i in today.todo) {
    today.todo[i]();
  }
  URGENT_TASKS[
    Object.keys(URGENT_TASKS)[
        Math.floor(Math.random() *
        Object.keys(URGENT_TASKS).length)
    ]
  ] = SEARCH_YOUTUBE_BREAD_TUTORIALS;
}, 86400000);
```

Toronto Ontario Tonetta




```
<html>
<head>
<meta http-equiv="refresh" content="1">
<script>
var aux = ["", "can", "don't", "must", "will", "won't"],
  act = ["hear", "know", "learn", "kiss", "die for", "see"];
function pick(list) {
 return list[~~(Math.random() * list.length)];
}
function phrase() {
 var end = " Tonetta";
 if (Math.random() < .3) {
  end = " that " + phrase();
 }
 return "Toronto " + pick(aux) + " " + pick(act) + end;
}
</script>
</head>
<body style="background-color:red;">
<body id=b style="margin: 30%" onload="b.innerHTML=phrase();">
<body text="white">
</html>
```

Dopesmokers' Problem

```
def dope_smoker():
  repeat:
    paper.wait()
    matches.wait()
    smoke()
    dope_smoker_done.signal()
```

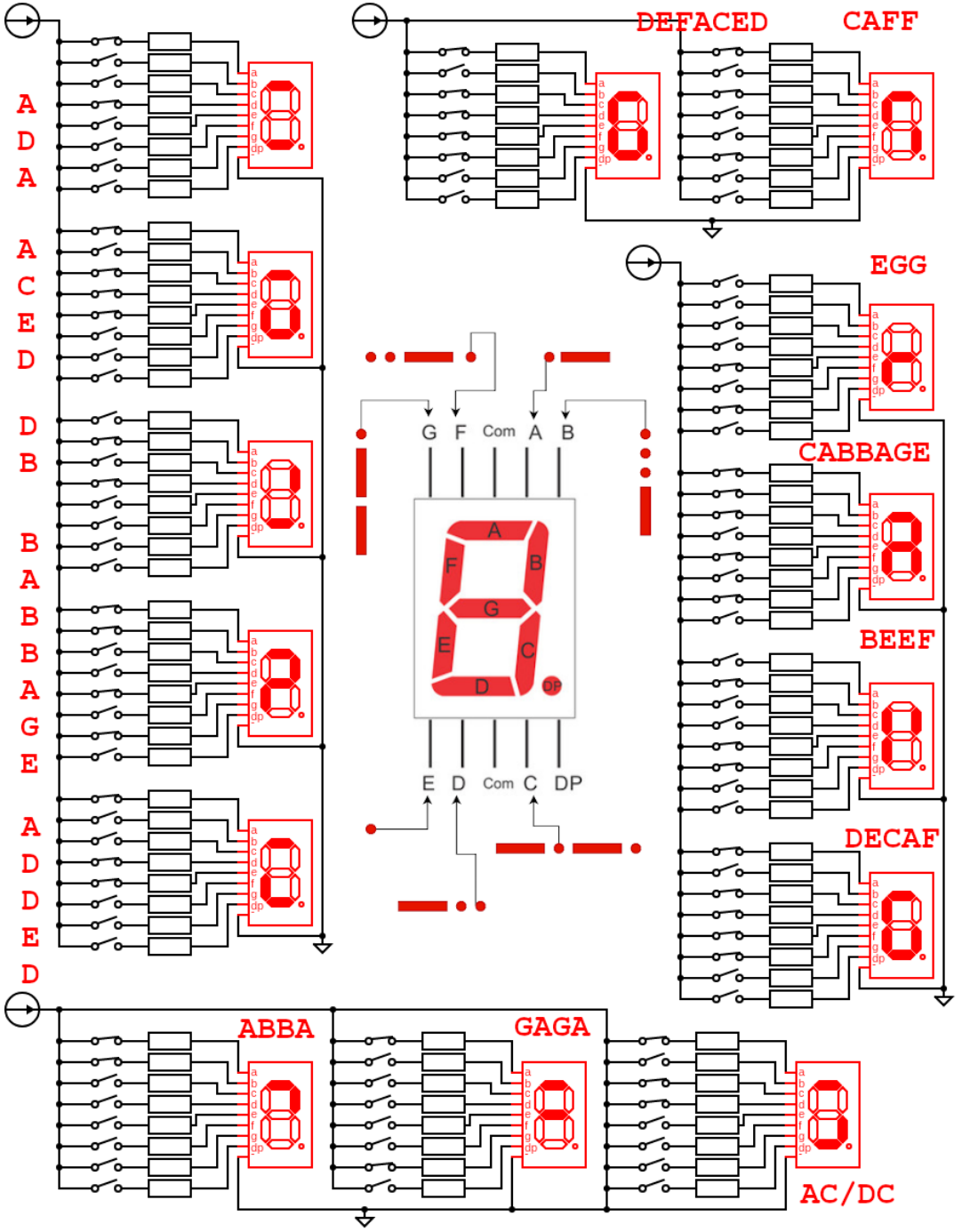
```
def dope_smoker():
  repeat:
    paper.wait()
    matches.wait()
    smoke()
    dope_smoker_done.signal()
```

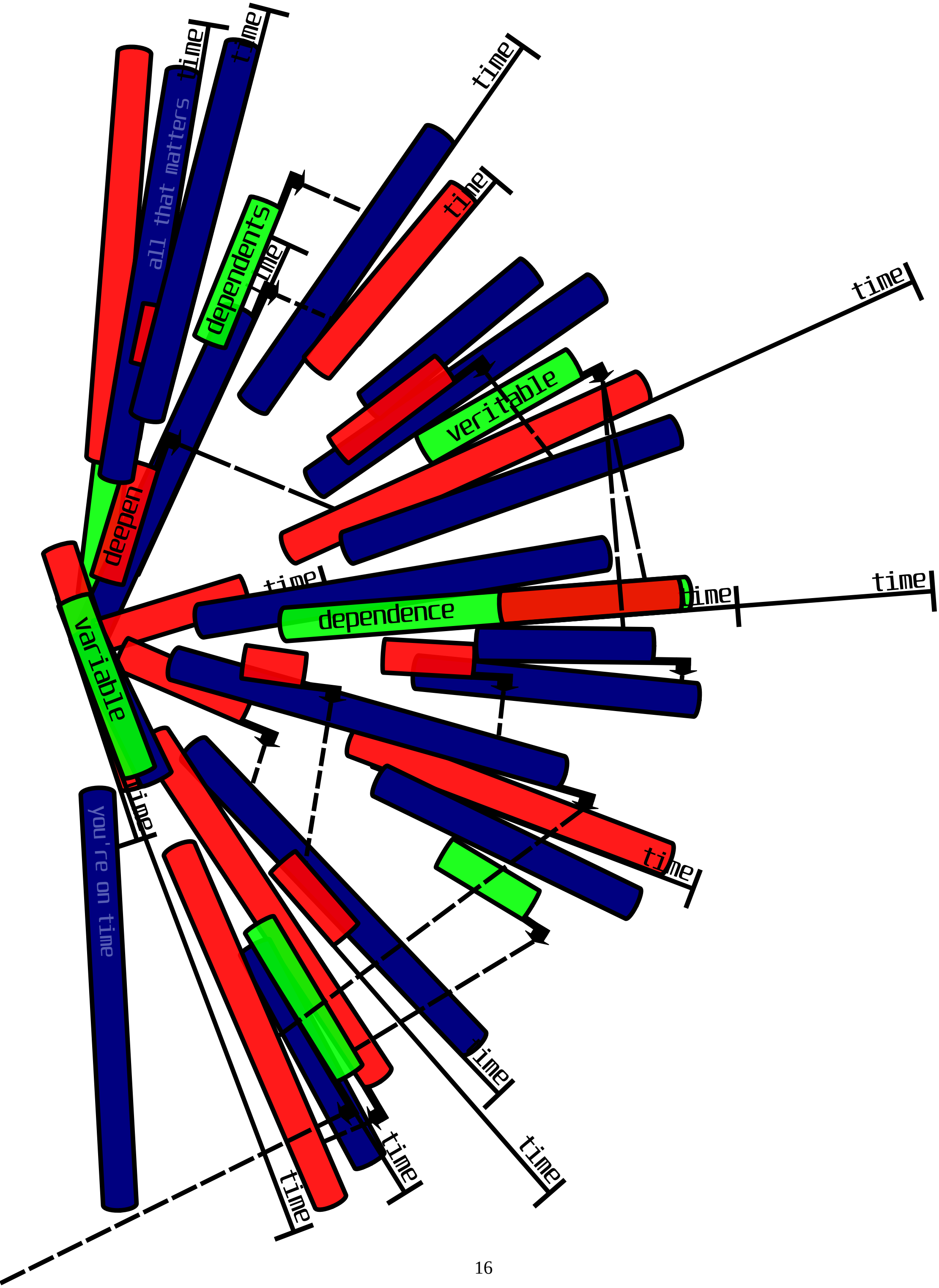
Bus factor

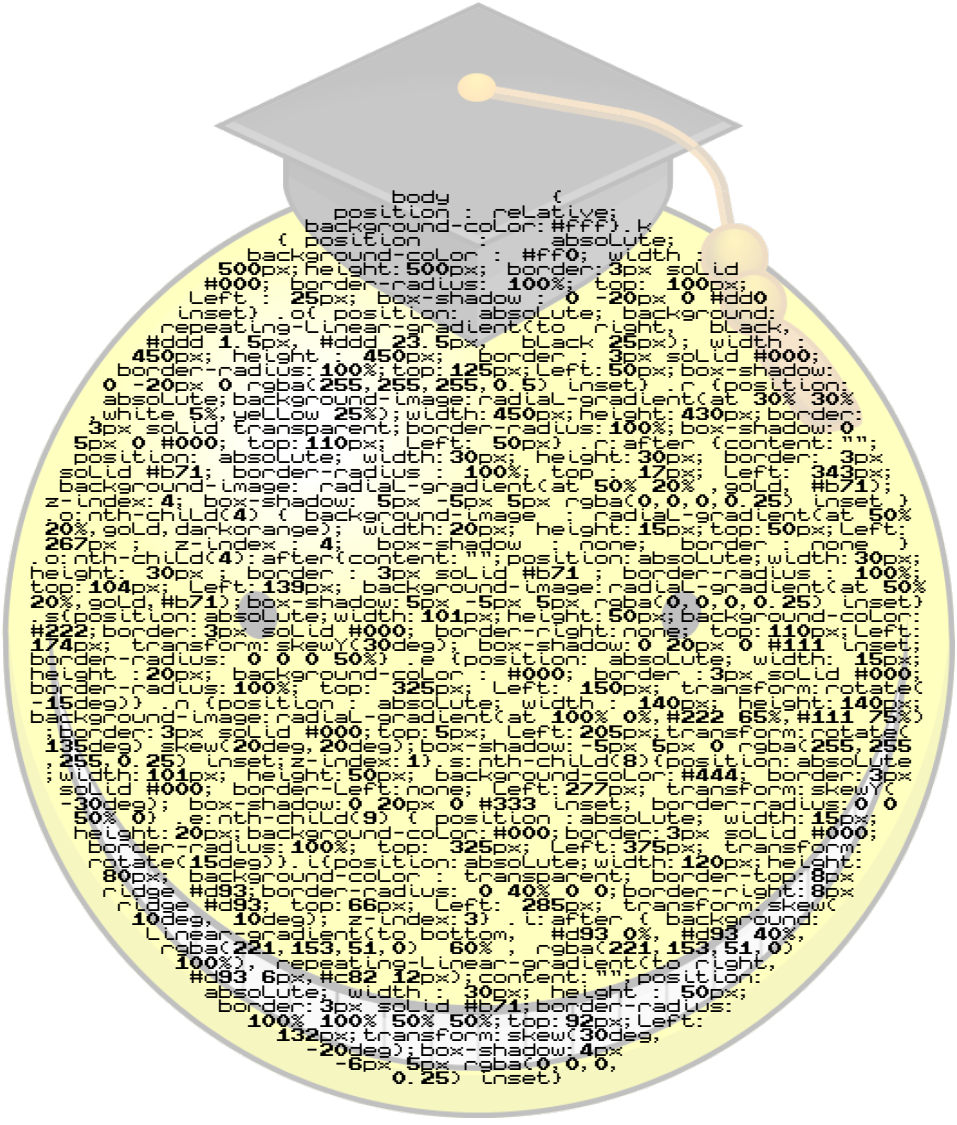
Unlike the block comment
{soundproofed testudo}
the single line comment exceeds the character limit
in free climate change lecture to the arresting officer.
The comment box under Section 14
/*the right to silence*/
expands dynamically
and is never disabled.
The line commenter throws his fellow travellers under the bus
the cops have requisitioned for arrestee processing.

```
/**************************\
* No comment no comment no *
* comment no comment no com *
* rade no comment no comm   *
* ent no comment no comment *
\**************************/
```

The police kettle breaks the code
and comments comrades out.
To escape you must break
the arbitrary delimiter line:
commit an illegal operation.


DEFACED
CAFF
EGG
CABBAGE
BEEF
DECAF
ABBA
GAGA
AC/DC
ADA
ACED
DB
BABBAGE
ADDED
G F Com A B
A
F B
G
E C
D DP
E D Com C DP


time
time
all that matters
time
time
dependents
time
time
veritable
deepen
time
time
dependence
time
variable
time
you're on time
ime
time
time
time



```
body {
  position : relative;
  background-color:#fff}.k
{ position : absolute;
  background-color : #ff0; width :
500px;height:500px; border:3px solid
#000; border-radius: 100%; top: 100px;
Left : 25px; box-shadow : 0 -20px 0 #ddd
inset} .o{ position: absolute; background:
repeating-linear-gradient(to right, black,
#ddd 1.5px, #ddd 23.5px, black 25px); width :
450px; height : 450px; border : 3px solid #000;
border-radius:100%;top:125px;left:50px;box-shadow:
0 -20px 0 rgba(255,255,255,0.5) inset} .r {position:
absolute;background-image:radial-gradient(at 30% 30%
,white 5%,yellow 25%);width:450px;height:430px;border:
3px solid transparent;border-radius:100%;box-shadow:0
5px 0 #000; top:110px; left: 50px} .r:after {content: "";
position: absolute; width:30px; height:30px; border: 3px
solid #b71; border-radius : 100%; top : 17px; Left: 343px;
background-image: radial-gradient(at 50% 20% ,gold, #b71);
z-index:4; box-shadow: 5px -5px 5px rgba(0,0,0,0.25) inset }
.o:nth-child(4) { background-image : radial-gradient(at 50%
20%,gold,darkorange); width:20px; height:15px;top:50px;left:
267px ; z-index : 4; box-shadow : none; border : none }
.o:nth-child(4):after{content: "";position:absolute;width:30px;
height: 30px ; border : 3px solid #b71 ; border-radius : 100%;
top:104px; Left:139px; background-image:radial-gradient(at 50%
20%,gold,#b71);box-shadow:5px -5px 5px rgba(0,0,0,0.25) inset}
.s{position:absolute;width:101px;height:50px;background-color:
#222;border:3px solid #000; border-right:none; top:110px;left:
174px; transform:skewY(30deg); box-shadow:0 20px 0 #111 inset;
border-radius: 0 0 0 50%} .e {position: absolute; width: 15px;
height :20px; background-color : #000; border :3px solid #000;
border-radius:100%; top: 325px; Left: 150px; transform:rotate(
-15deg)} .n {position : absolute; width : 140px; height:140px;
background-image:radial-gradient(at 100% 0%,#222 65%,#111 75%)
;border:3px solid #000;top:5px; Left:205px;transform:rotate(
135deg) skew(20deg,20deg);box-shadow: 5px 5px 0 rgba(255,255
,255,0.25) inset;z-index:1} .s:nth-child(8){position:absolute
;width:101px; height:50px; background-color:#444; border:3px
solid #000; border-left:none; left:277px; transform:skewY(
-30deg); box-shadow:0 20px 0 #333 inset; border-radius:0 0
50% 0} .e:nth-child(9) { position :absolute; width:15px;
height:20px;background-color:#000;border:3px solid #000;
border-radius:100%; top: 325px; left:375px; transform:
rotate(15deg)}.i{position:absolute;width:120px;height:
80px; background-color : transparent; border-top:8px
ridge #d93;border-radius: 0 40% 0 0;border-right:8px
ridge #d93; top:66px; left: 285px; transform:skew(
10deg, 10deg); z-index:3} .i:after { background:
linear-gradient(to bottom, #d93 0%, #d93 40%,
rgba(221,153,51,0) 60% , rgba(221,153,51,0)
100%), repeating-linear-gradient(to right,
#d93 6px,#c82 12px);content: "";position:
absolute; width : 30px; height : 50px;
border:3px solid #b71;border-radius:
100% 100% 50% 50%;top:92px;left:
132px;transform:skew(30deg,
-20deg);box-shadow:4px
-6px 5px rgba(0,0,0,
0.25) inset}
```

```
<div class="k"></div>
<div class="o"></div>
<div class="r"></div>
<div class="o"></div>
<div class="s"></div>
<div class="e"></div>
<div class="n"></div>
<div class="s"></div>
<div class="e"></div>
<div class="i"></div>
```

```
int
main(
int argc,char
*argv[]){char *addr
;size_t len;int is_pmem;
addr=pmem_map_file(argv[1]
,4096,PMEM_FILE_CREATE,0666
,&len,&is_pmem);strcpy(addr,
"The Persistence of Memory"
);pmem_persist(addr,len);
pmem_unmap(addr,len);
return 0;}
```

```
#include <stdio.h>

int main(){
    puts((char[]){
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<
        !!""<<!!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""|+
        !!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<
        !!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<
        !!""|+!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<
        !!""<<!!""|+!!""<<!!""<<!!"",+!!""<<!!""<<!!""<<!!""<<
        !!""<<!!""<<!!""|+!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""|+!!""<<!!""|+
        !!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<
        !!""<<!!""|+!!""<<!!""<<!!"",+!!""<<!!""<<!!""<<!!""<<
        !!""<<!!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""|+!!""<<
        !!""|+!!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""<<
        !!""|+!!""<<!!""<<!!""|+!!""<<!!""|+!!"",+!!""<<!!""<<
        !!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""<<!!""<<
        !!""<<!!""|+!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!"",+
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<
        !!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!""<<!!""|+!!""<<
        !!""<<!!"",+!!""<<!!""<<!!""<<!!""<<!!""<<!!""<<!!""|+
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!""<<!!""<<!!"",+
        !!""<<!!""<<!!""<<!!""<<!!""<<!!""|+!!"",!""/*Hello!*/
    });
}
```

Notes

Page 5. This piece contains an instance of a graphical, esoteric programming language called Flowpath, which has been inspired by 10 PRINT algorithms, cellular automata, and the long history of using liquids as a computing medium, from water clocks to fluidic logic systems. It is an interpreted language.

Page 9-10. Created using in:verse, a creative programming language which you can find at https://inverse.website/.

Contributors

Bora Aydintug: An experimental visual artist from Istanbul. He's currently based in Brooklyn. He works with video, code and various software to create abstract art. His main ambition is to invite curiosity and openness in others. https://www.boraaydintug.com

Johan Berg: A software developer from Sweden. He loves learning new things and sharing knowledge with others.

Richard A Carter: An artist and lecturer in Digital Media. Carter's artistic work explores the potentialities of seeing, knowing, and writing at the intersection between human and machinic actors.

S Cearley: A person who tricked a computer into making poetry when it thinks it is making art.

Brian James: An educator and creative coder living in Brooklyn. He is interested in the links between design, language, and technology.

Chris Kerr: From London. He collaborated with Daniel Holden on ./code –poetry. His first pamphlet, *Citidyll*, was published by Broken Sleep Books. His work has appeared in *Adjacent Pineapple*, *Anthropocene*, *Blackbox Manifold*, *Haverthorn*, and others.

Juan Alberto "stage7" Martínez: Self-taught in as most artistic areas as he have been able to put my hands in, he is always searching new ways to blend as many of them as possible to blur the border between programming and everything else, mainly for the local demoscene.

Krzysztof Siejkowski: Exploring the "language" part of programming languages while making a living with the "programming" part. Pieces at swiftpoetry.com. From Warsaw with love.

Luna Sorcery: While she's busy procrastinating writing her bio for this issue, Luna also enjoys making pretty pictures with code, disassembling old games, and torturing C++ compilers to within an inch of their life.

C. R. Resetarits: A writer and artist. Her art had appeared on the covers and in the pages of dozens of little literary mags. She lives in Faulkner-riddled Oxford, Mississippi.

Ólafur Waage: Types on keyboards for a living, sometimes for computers, sometimes for people. You can yell at him on Twitter: @olafurw.

Logan Young: Published everywhere from Jacket2 to the forthcoming second volume of Tonebook, Logan K. Young's recent exhibitions include Synchrony 2020 Demoparty and Offprint London at the Tate Modern. His factorial chapbook, I(<3)U!, is out now.

```
code::art::1::end()
```

code::art is an art journal which publishes code.

Its intent is to further explore the relationship between code and art, and to challenge our perceptions of what both can look like.

Gantt charts as sprawling monstrosities, fortifications built in Swift code, instructions on how to never ask out your crush, what may be the world's worst (best?) Hello World program, C interpretations of Dali. All of these and more await you in this collection.