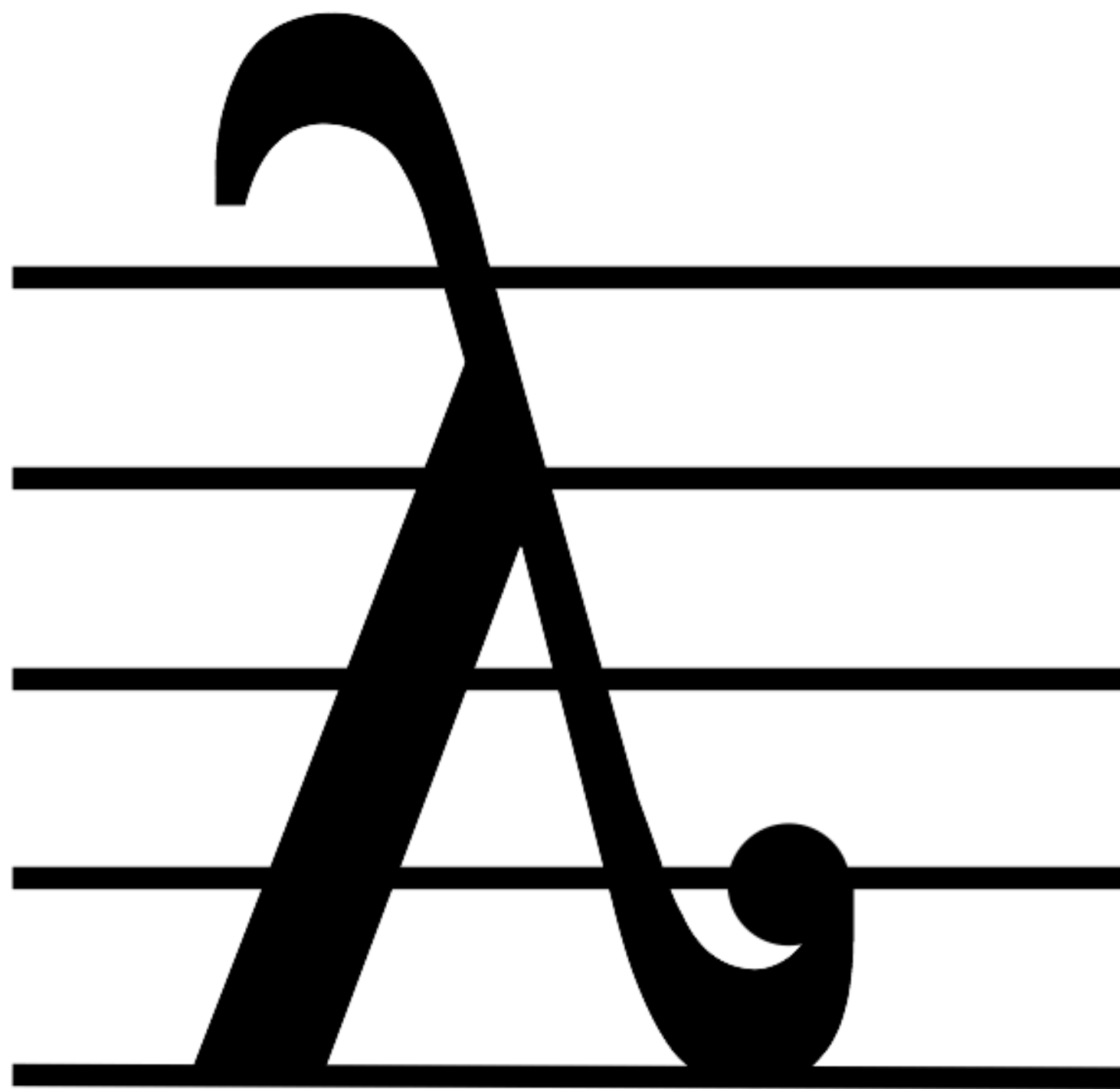


LEARN AS YOU
PLAY

SAM AARON





improcess





The Timeless Way of Building



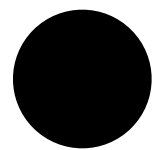
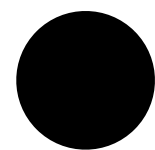
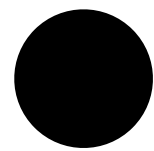
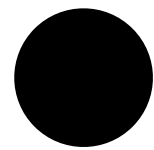
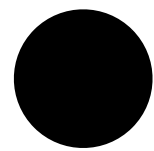
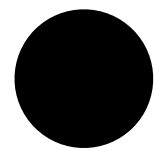
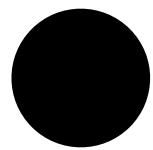
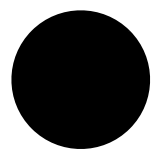
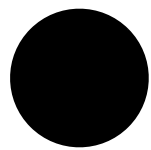
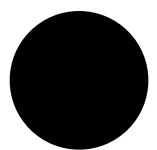
Christopher Alexander

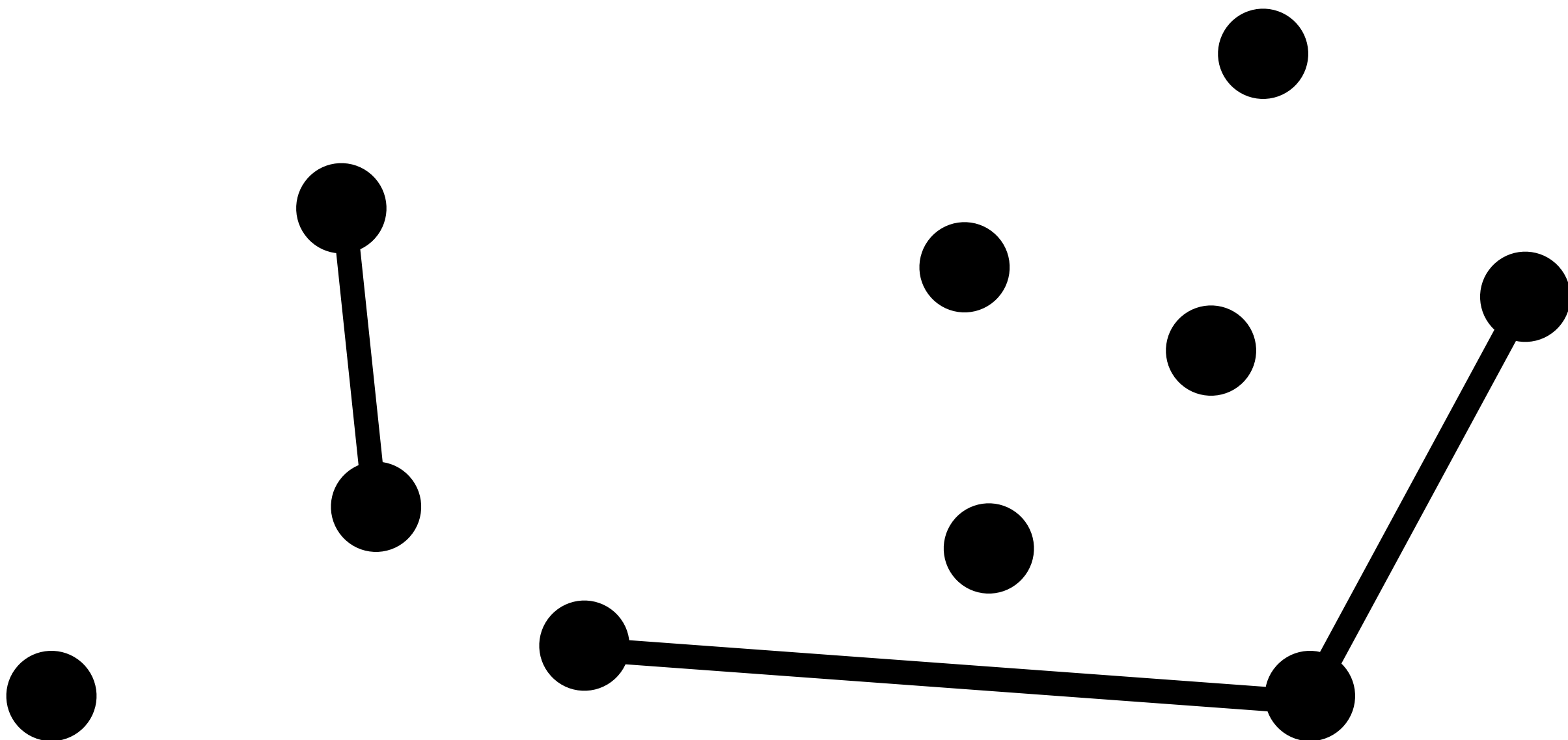
NOTES ON THE
SYNTHESIS
OF FORM
CHRISTOPHER ALEXANDER

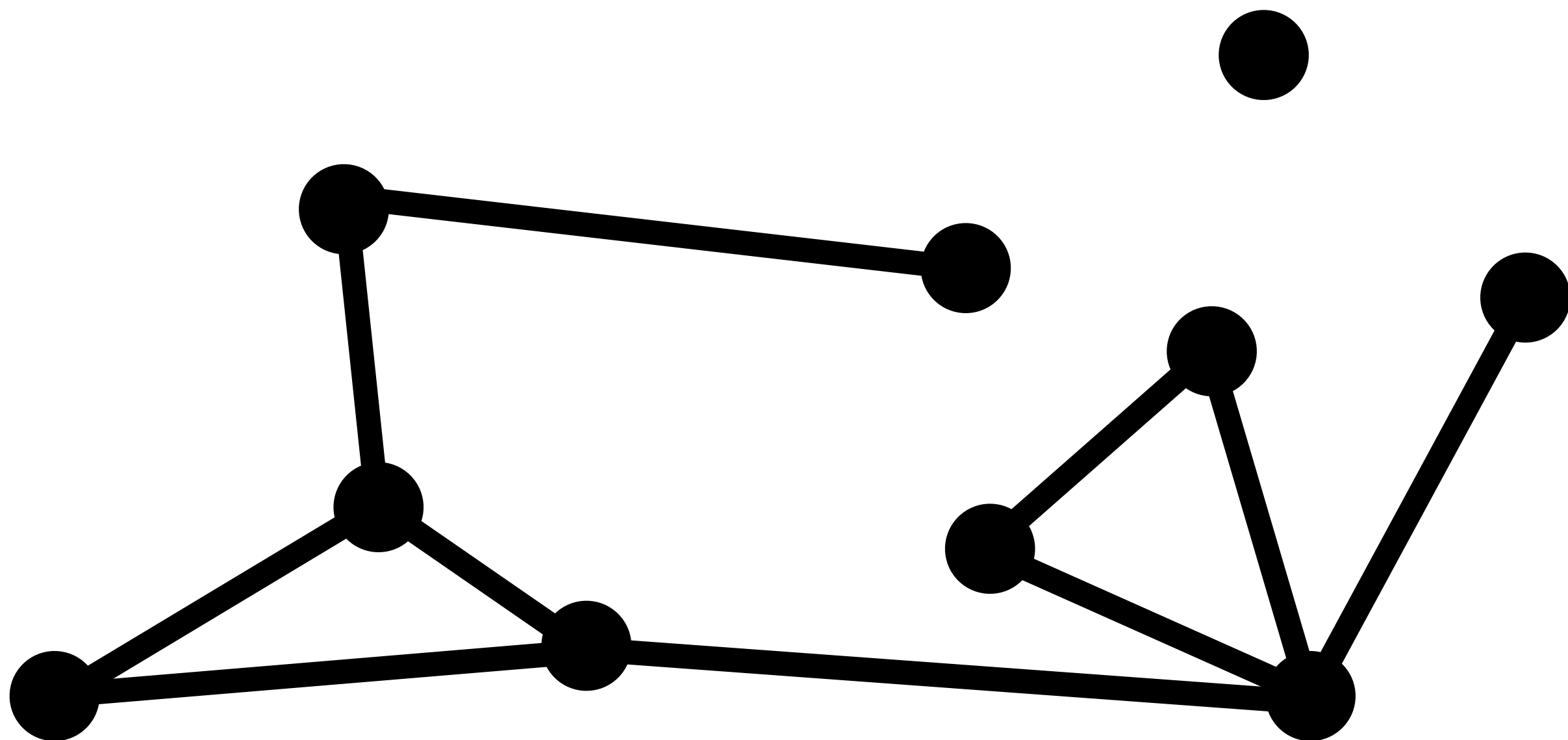


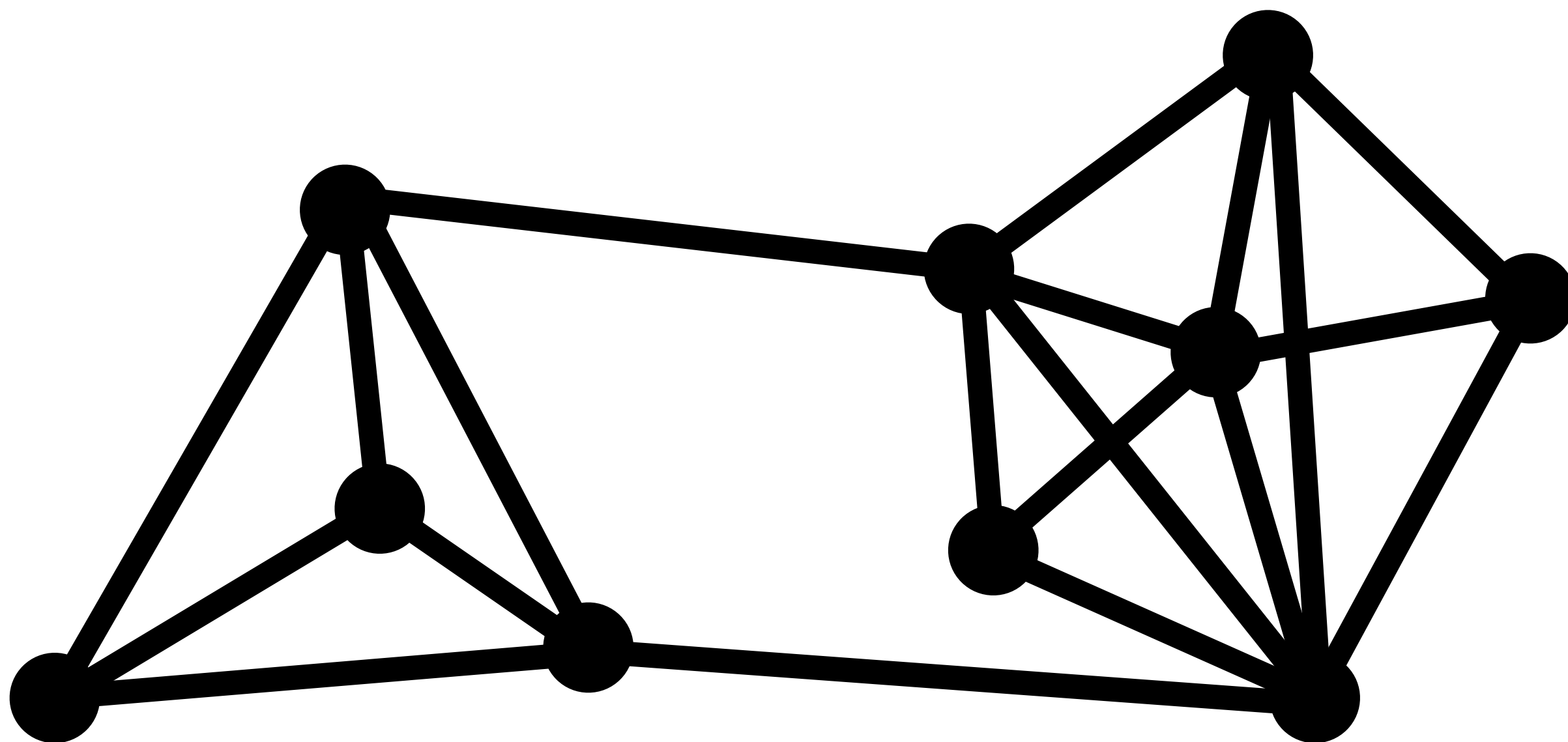
*[Design is]
"the process of inventing things
which display new physical order,
organisation, form, in response to
function..."*

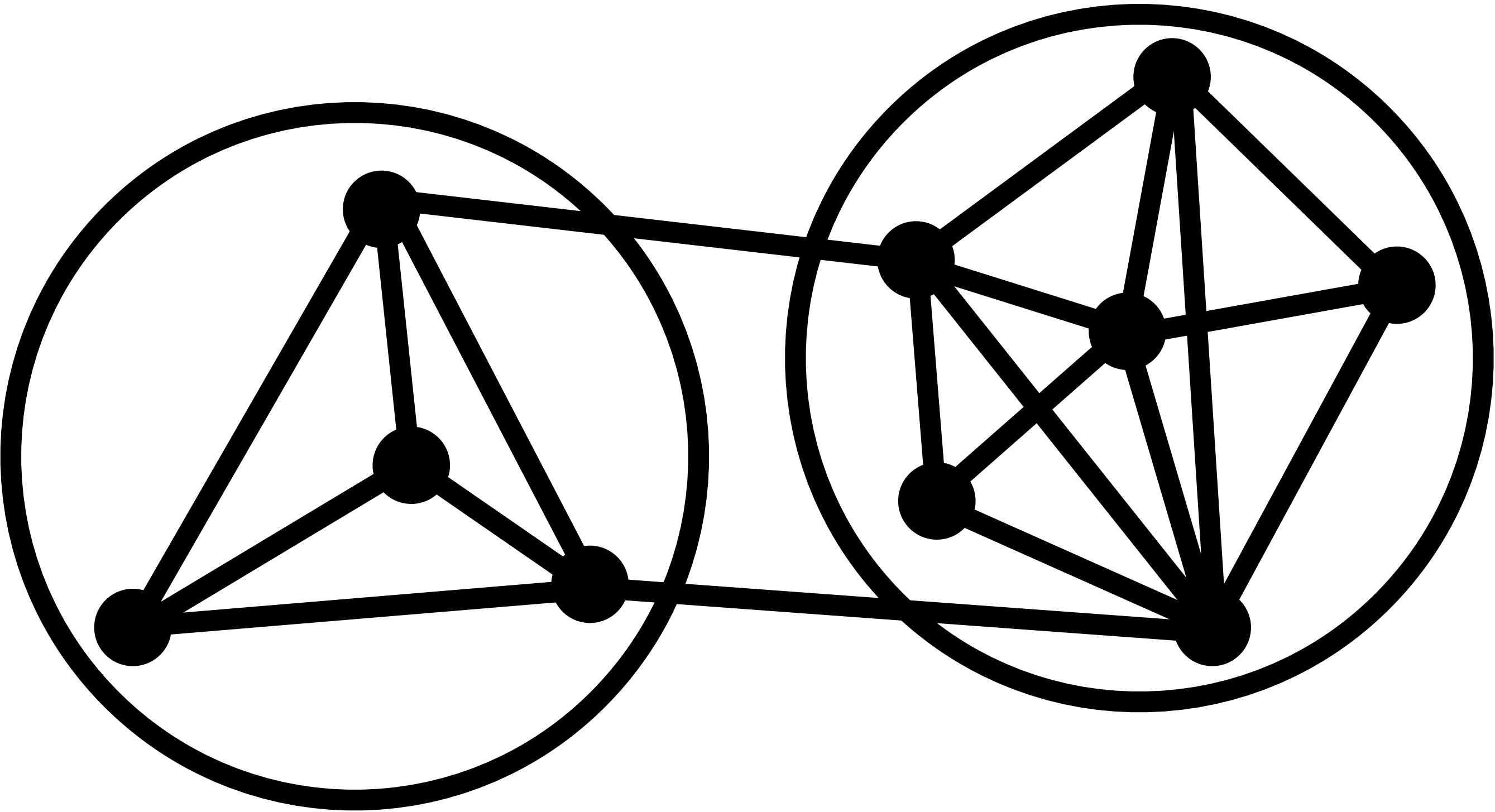
Christopher Alexander

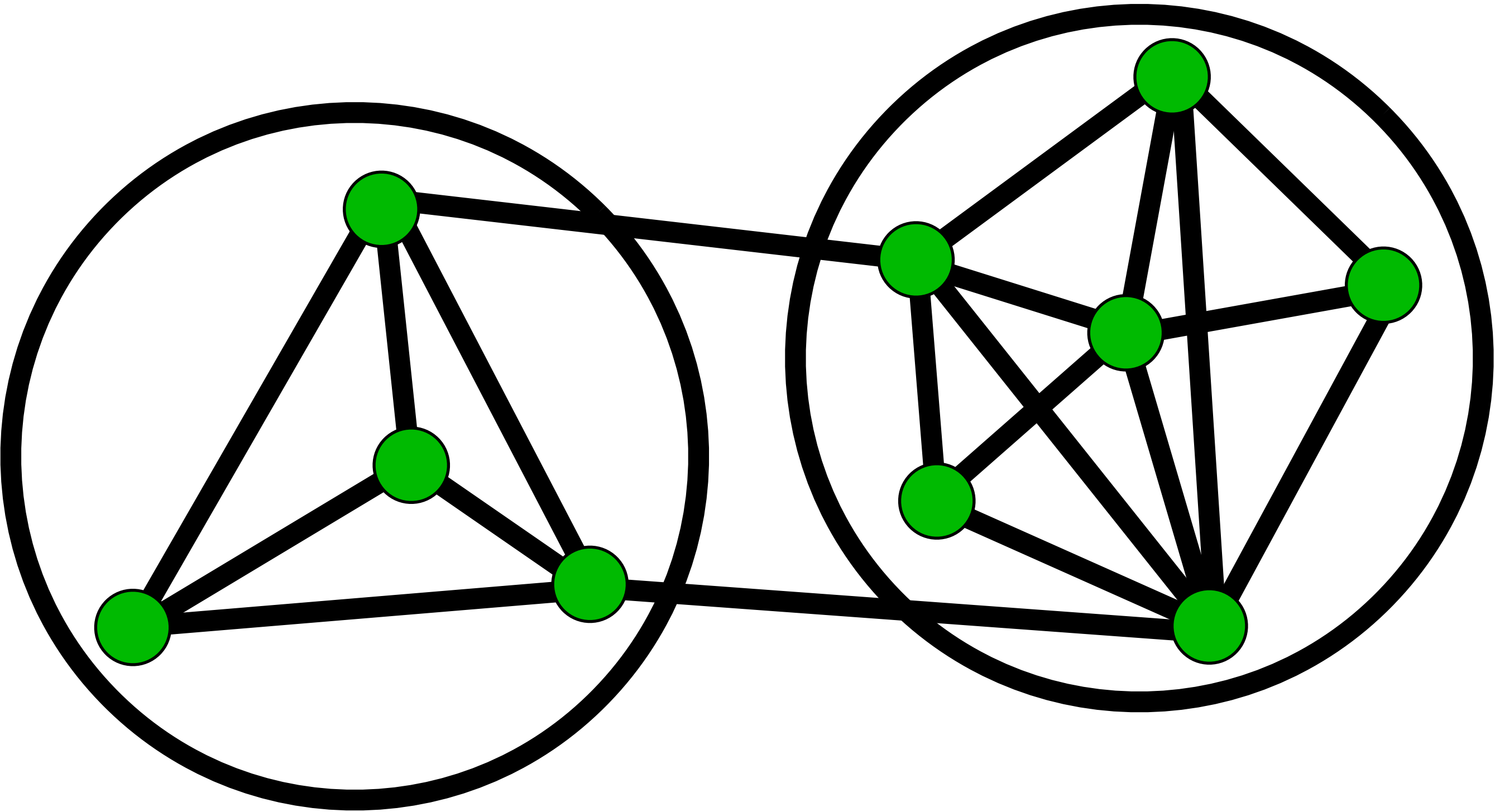


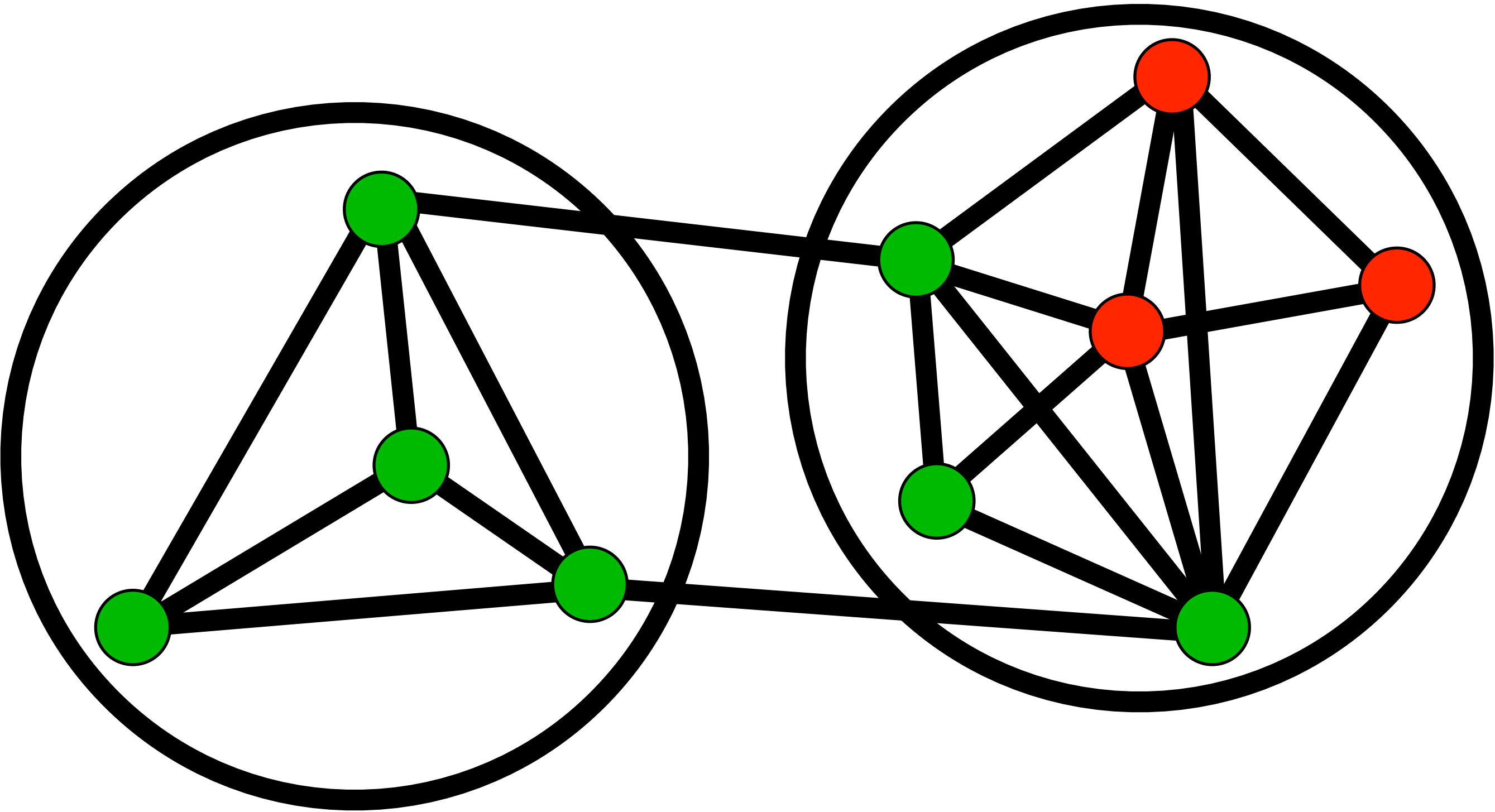


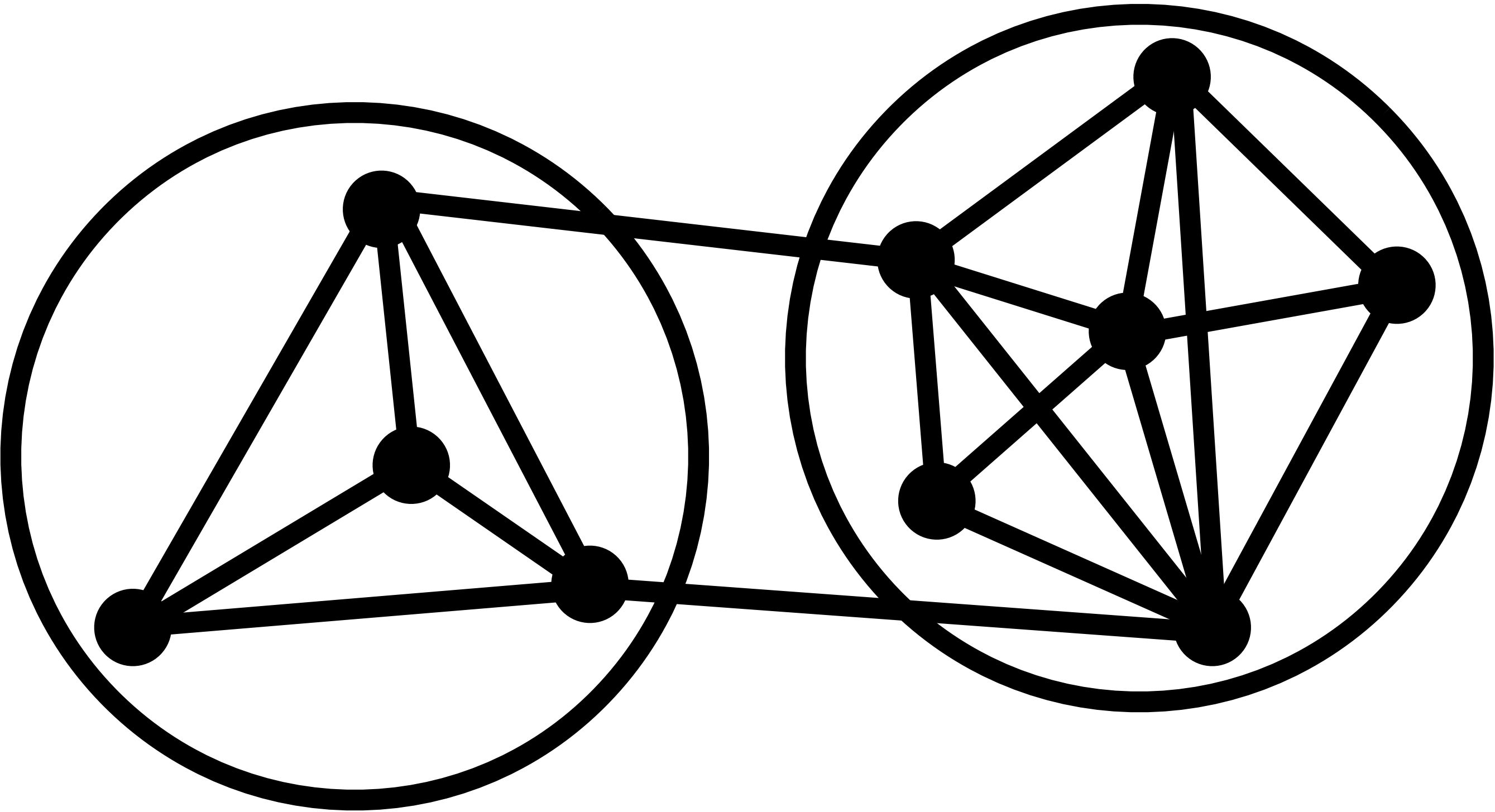


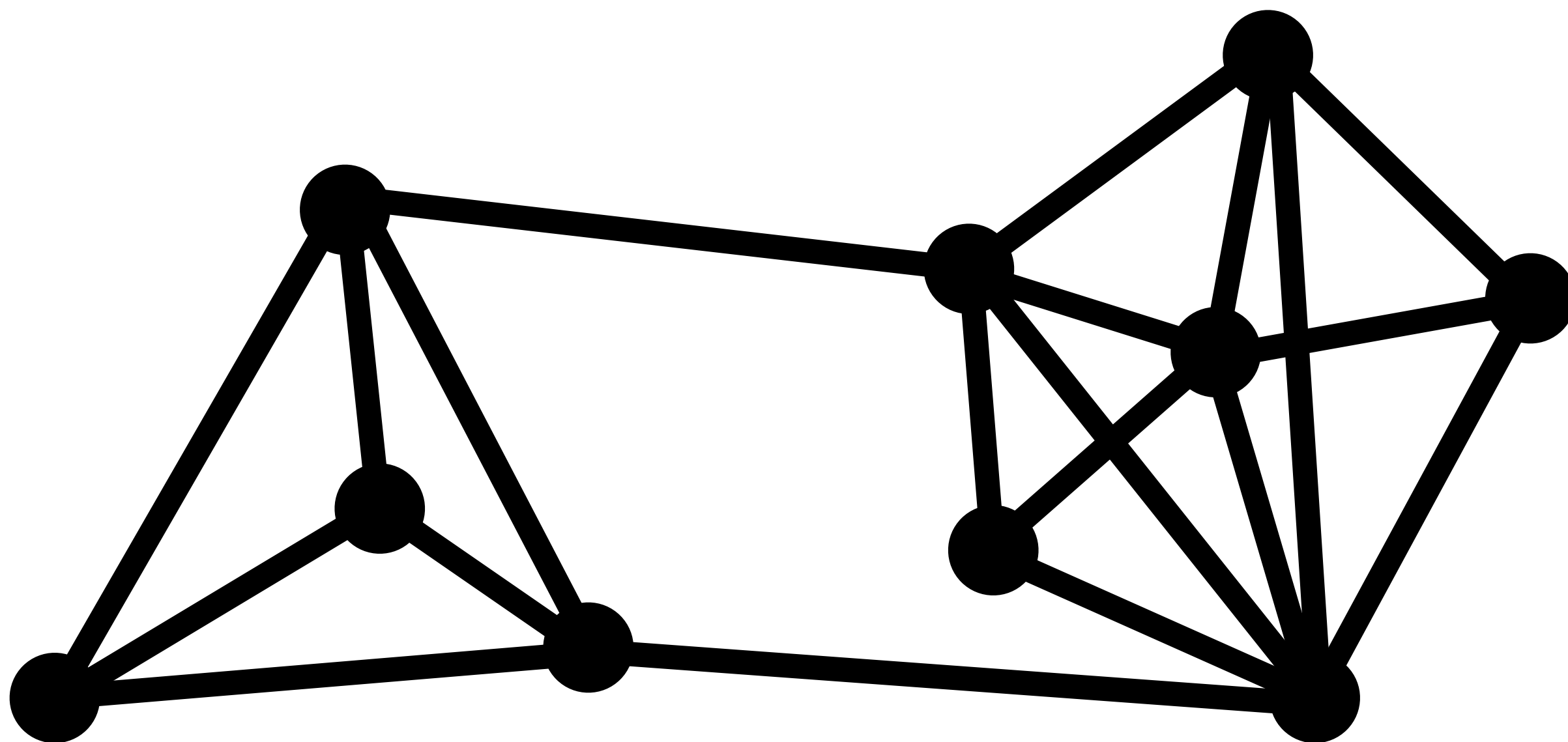


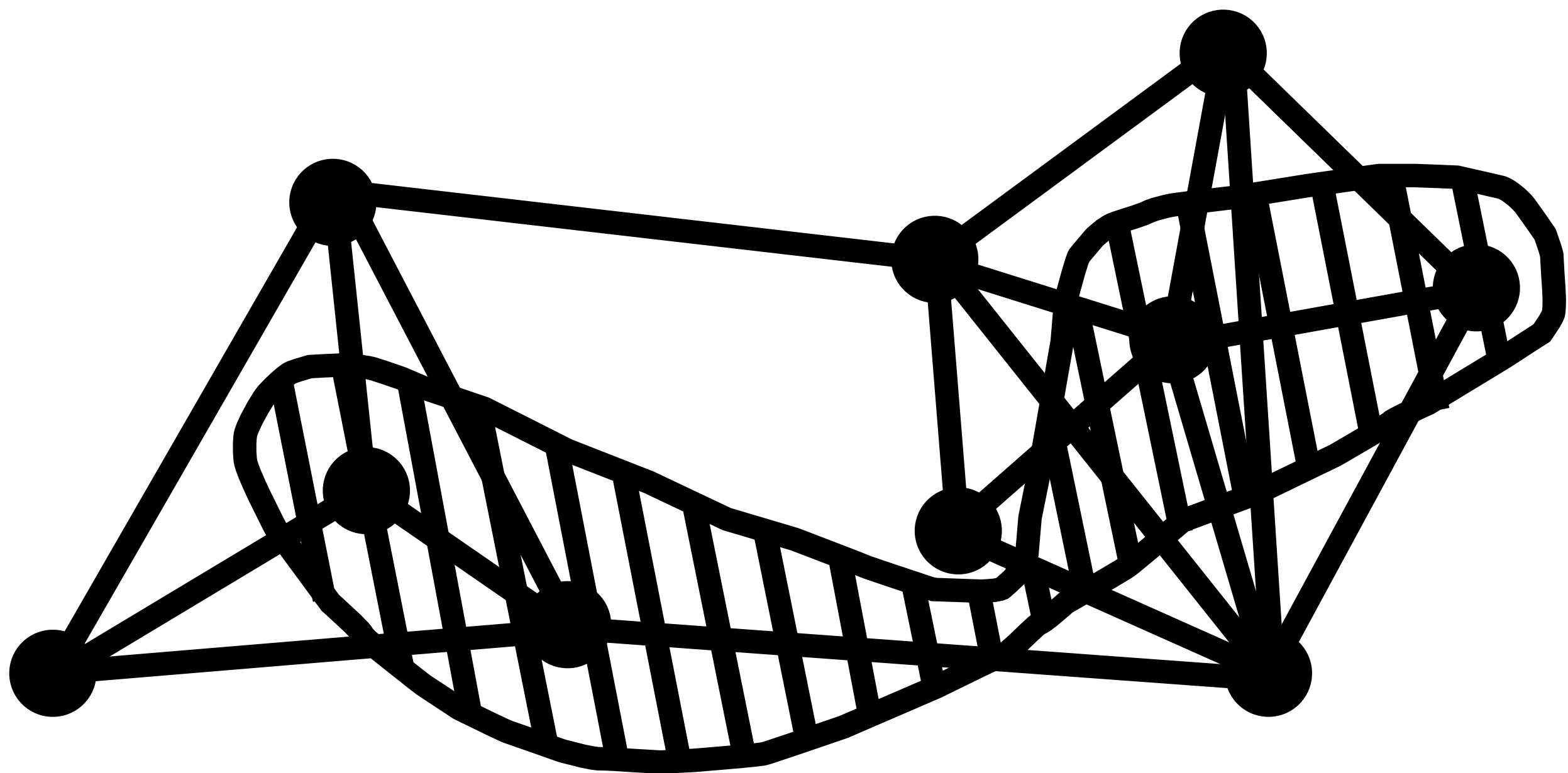


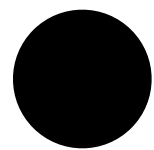
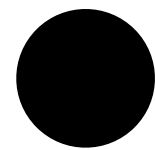
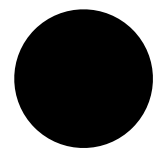
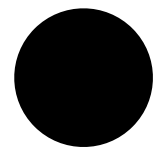
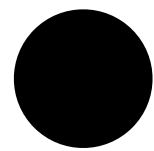
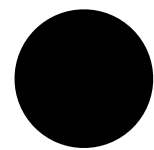
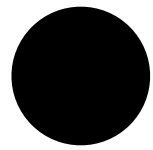
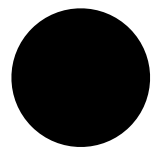
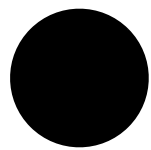
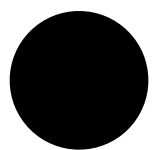


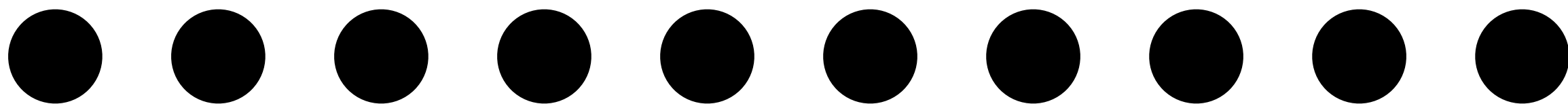




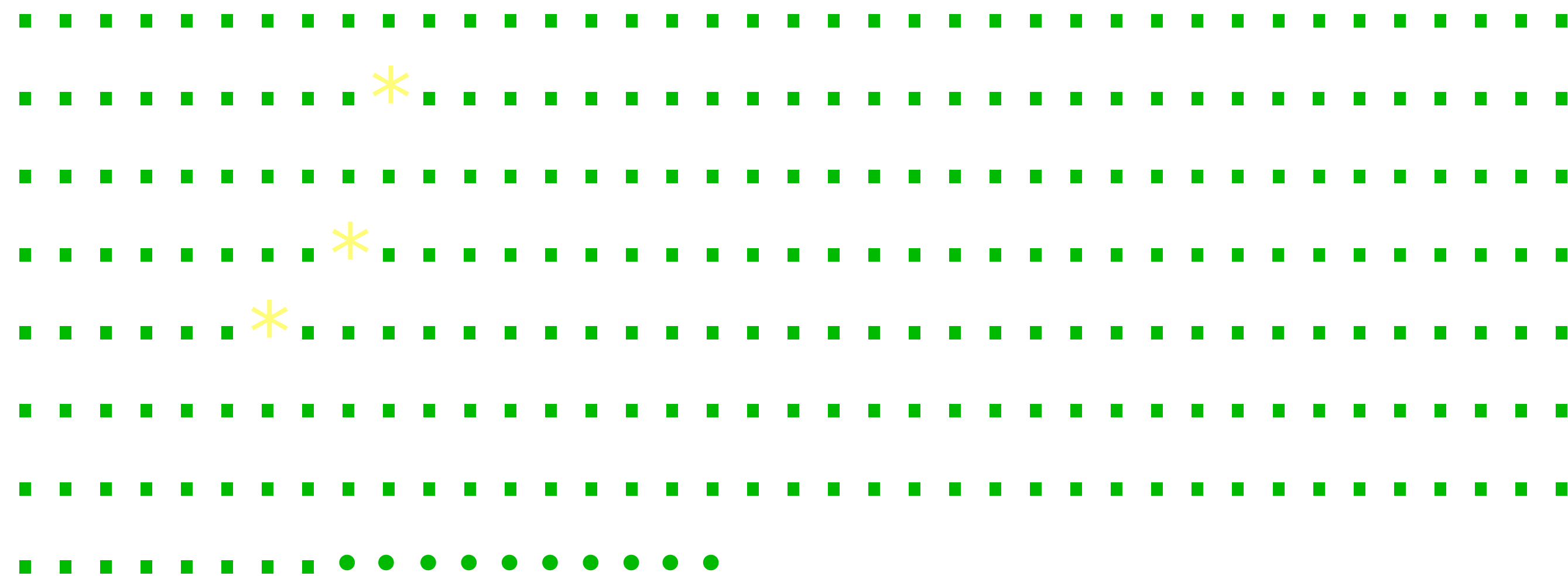








λ rake spec
(in /Users/sam/Development/monomer)







Quentin Blake 2009

```

require 'open-uri'; require 'rubygems'; require 'hpricot'
url, save_as = *ARGV
KEEPVID = URI(Hpricot(URI("http://keepvid.com/").read).
               at("form")['action'])

```

```

Net::HTTP.start(KEEPVID.host, KEEPVID.port) do |http|
  doc = Hpricot(http.post(KEEPVID.path, "url=#{url}&site=aa").body)
  url = URI((doc/:%a).detect{|x| x.next_node.to_s.index("(flv - Flash")}['href'])
end

```

```

Net::HTTP.start(url.host, url.port) do |http|
  http.request_get(url.request_uri, {'User-Agent' => 'ruby', 'Host' => url.host}) do
    |r|
    File.open(save_as, 'wb') do |f|
      len = 0
      r.read_body do |chunk|
        f.write(chunk)
        len += chunk.length
        print "%30s / %d of %d / %0.1f%%\r" %
          [save_as, len, r.content_length, ((len * 1.0) / r.content_length) * 100]
      end
    end
  end
end

```

```
"eyes".scan /the_darkness/  
catch( :in_the_wind ) { ?a.round; "breath" \  
  or "a".slice /of_moon/ }
```


better watchout

better !cry !pout

lpr why

santa claus <north pole > town

cat /etc/passwd > list

ncheck list

ncheck list

cat list | grep naughty > nogiftlist

cat list | grep nice > giftlist

santa claus <north pole > town

who | grep sleeping

who | grep awake

who | egrep 'bad|good'

for (goodness sake) {be good}

```

%w[tempfile uri].map{|l|require l};class Object;def meta_def m,&b;(class<<self
self end).send(:define_method,m,&b end end;module Camping;C=sel
S=IO.read(__FILE__)rescue nil;P=<h1>Cam\ping Problem!</h1><h2>4s</h2>
class H<Hash
def method_missing m,*a;m.to_s==~/$/?self[$]=a[0]:a==[]?self[m.to_s]:super end
alias u merge!;undef id,type;end;module Helpers;def R c,*g
p,h=/(.+?)/,g.grep(Hash);g->h;raise"bad route"unless u=c.urls.find{|x|
break x if x.scan(p).size==g.size&&/^#{x}\$/--(x=g.inject(x){|x,a|
x.sub p,C.escape([a[a.class.primary_key]rescue a])}})
h.any?? u+"?">h[0].map{|x|x.map{|z|C.escape z}==">">"&">:u end;def / p
p[/^\/]/?@root+p:p end;def URL c="/",*a;c=R(c,*a) if c.respond_to?:urls
c=self/c;c="//"+@env.HTTP_HOST+c if c[/^\/]/;URI c end end;module Base
attr_accessor:input,:cookies,:env,:headers,:body,:status,:root;Z="\r\n"
def method_missing *a,&b;a.shift if a[0]==:render;m=Mab.new({},self)
s=m.capture(send(*a,&b));s=m.capture(send(:layout){s})if/^_/!-a[0].to_s and
m.respond_to?:layout;s end;def r s,b,h={};@status=s;headers.u(h);@body=b
end;def redirect *a;r 302,"","Location">URL(*a)end;def r404 p=env.PATH
r 404,P%#{p} not found"end;def r500 k,m,x
r 500,P%#{k}.#{m}*>#{x.class} #{x.message}:<ul>#{x.
backtrace.map{|b|<li>#{b}</li>}</ul></h3>"end;def r501 m=@method
r 501,P%#{m.upcase} not implemented"end;def to_a
[status,body,headers]end;def initialize r,e,m;@status,@method,@env,@headers,
@root=200,m,e,H['Content-Type','text/html'],e.SCRIPT_NAME.sub(/\/$/,'')
@k=C.kp e.HTTP_COOKIE;q=C.qsp e.QUERY_STRING;@in=r;case e.CONTENT_TYPE
when/r|\Amultipart/form-.*boundary=\\?([^\";,]+)\\n
b=/(?:\\r?\\n\\A)#{Regexp.quote"--#s1"}(?:--)?\\r$/;until@in.eof?;fh=H[]
for l in@in;case l;when Z;break;when/^Content-D.+?: form-data;/
fh.u H["$".scan(/(?:\\s\\w+)=("[^"]+)"/).flatten]
when/^Content-Type: (.+?)(\\r$|\\Z)/m: fh.type = $1 end end;fn=fh.name
o=if fh.filename;o=fh.tempfile=Tempfile.new(:C);o.binmode;else;fh="";end;s=8192
k="";l=@in.read(s*2);while l;if(k<<l)--b;o<<$'.chomp
@in.seek(-$'.size,IO::SEEK_CUR);break end;o<<k.slice(0...s);l=@in.read(s) end
C.qsp(fn,"&";',fh,q)if fn;fh.tempfile.rewind if fh.is_a?H end;when
"application/x-www-form-urlencoded": q.u(C.qsp(@in.read))end
@cookies,@input=@k.dup,q.dup end;def service *a;@body=send @method,*a
headers['Set-Cookie']=cookies.map{|k,v|>#{k}>#{C.escape(v)}; path=>#{self/
"/"}>#{if v!=@k[k]}-[nil];self end;def to_s;"Status: #{@status}>#{Z+(headers.inject([
]){|a,o|>#{o[1]}.map{|v|a<<[o[0],v]}: "if v&&v.to_s.any?};a">Z)+Z+Z}>#{@body}end
end;X=module Controllers;@r=[];class<<self;def r;@r end;def R *u;r=@r
Class.new(meta_def(:urls){u};meta_def(:inherited){|x|r<<x}}end
def D p,m;r.map{|k|k.urls.map{|x|return(k.instance_method(m)rescue nil)?
[k,m,"$-[1..-1]":[I,'r501',m]if p==/^(#{x}\\$/)};[I,'r404',p]
end;def M;def M;end;constants.map{|c|k=const_get(c)
k.send:include,C,Base,Helpers,Models;@r=[k]+r if r-[k]==r
k.meta_def(:urls){[/#{c.downcase}]/}if !k.respond_to?:urls}end end;class I<R()
end;self end;class<<self;def goes m
eval S.gsub(/Camping/,m.to_s),TOPLEVEL_BINDING end;def escape s
s.to_s.gsub(/[^ \\w.-]+/n){'%'+($&.unpack("H2"*$&.size)*"%").upcase}.tr' ','+'
end;def un s;s.tr(' ',' ').gsub(/%{[\\da-f]{2}}/in){[$1].pack'H'}end
def qsp q,d='&';y=nil,z=H[];m=proc{|_,o,n|o.u(n,&m)rescue{>#{o}<<n}}
(q.to_s.split(/[#{d}]+ */n)-[""]).inject((b,z=Z,H[])>[0]){|h,p|k,v=un(p).
split="',2;h.u k.split(/\\[\\[+/).reverse.inject(y||v){|x,i|H[i,x]},&m}end
def kp s;c=qsp s,"';','end;def run r=$stdin,e=ENV;X.M=e=H[e.to_hash];k,m,*a=X.D e.
PATH_INFO=un("/#{e.PATH_INFO}".gsub(/\\+/,'/')),
(e.REQUEST_METHOD||'get').downcase
k.new(r,e,m).Y.service(*a);rescue=>x;X::I.new(r,e,'r500').service k,m,x
end;def method_missing m,c,*a;X.M;k=X.const_get(c).new StringIO.new,
H['HTTP_HOST'],'','SCRIPT_NAME','','HTTP_COOKIE',''],m.to_s
H[a.pop].each{|e,f|k.send">#{e}>=",f}if Hash==a[-1];k.service(*a)end end
module Views;include X,Helpers end;module Models;autoload:Base,'camping/db'
def Y;self;end end;autoload:Mab,'camping/mab'end

```



```

%w[tempfile url].map{|l|require l};class Object;def meta_def m,&b;(class<<self
self end).send:define_method,m,&b end end;module Camping;C=self
S=IO.read(__FILE__)rescue nil;P=<h1>Cam\ping Problem</h1><h2>%s</h2>"
class H<Hash
  method_missing m,*a;m.to_s=~/=~/?self[$']=a[0]:a==[]?self[m.to_s]:super end
  alias u merge!;undef id,type;end;module Helpers;def R c,*g
    p,h=/(.+?\\)/,g.grep(Hash);g-=h;raise"bad route"unless u=c.urls.find{|x|
      break x if x.scan(p).size==g.size&&/^#{x}\\/?$/=~(x=g.inject(x){|x,a|
        x.sub p,C.escape((a[a.class.primary_key]rescue a))})}
      h.any?? u+"?"+"h[0].map{|x|x.map{|z|C.escape z}*"}&"&":u end;def / p
        p[/^\\/]?@root+p end;def URL c='/','*a;c=R(c,*a) if c.respond_to?:urls
          c=self/c:c="//"+@env.HTTP_HOST+c if c[/^\\/]/:URI c end end;module Base
            attr_accessor:input,:cookies,:env,:headers,:body,:status,:root;Z="\r\n"
          def method_missing *a,&b;a.shift if a[0]==:render;m=Mab.new({},self)
            s=m.capture{send(*a,&b)};s=m.capture{send(:layout){s}}if/^/_/!~a[0].to_s and
              m.respond_to?:layout;s end;def r s,b,h={}:@status=s;headers.u(h);@body=b
            end;def redirect *a;r 302,',',"Location"=>URL(*a)end;def r404 p=env.PATH
              r 404,P%#{p} not found"end;def r500 k,m,x
                r 500,P%#{k}.#{m}"+<h3>#{x.class} #{x.message}: <ul>#{x.
                  backtrace.map{|b|<li>#{b}</li>}}</ul></h3>"end;def r501 m=@method
                    r 501,P%#{m.upcase} not implemented"end;def to_a
                      [status,body,headers]end;def initialize r,e,m,@status,@method,@env,@headers,
                        @root=200,m,e,H['Content-Type','text/html'],e.SCRIPT_NAME.sub(/\$/,'')
                        @k=C.kp e.HTTP_COOKIE;q=C.qsp e.QUERY_STRING;@in=r;case e.CONTENT_TYPE
                          when%r|\Amultipart/form-.*boundary="\?{([^\";,]+)|n
                            b=/(?:\r?\n|A)#(Regexp.quote"--#$1")(?:?--)?\r$/;until@in.eof?;fh=H[]
                              for l in@in:case l:when Z;break:when/^Content-D.+?: form-data:/
                                fh.u H[*$'.scan(/(?:\s+(\w+)=("[^"]+)"\/).flatten]
                                  when/^Content-Type: (.+?)(\r$|Z)/m: fh.type = $1 end end;fn=fh.name
                                    o=if fh.filename;o=fh.tempfile=Tempfile.new(:C):o.binmode;else;fh="";end;s=8192
                                      k='';l=@in.read(s*2);while l;if(k<=l)~=b;o<<$'.chomp
                                        @in.seek(-$'.size,IO::SEEK_CUR);break end;o<<k.slice!(0..s);l=@in.read(s) end
                                          C.qsp(fn,&';',fh,q)if fn;fh.tempfile.rewind if fh.is_a?H end;when
                                            "application/x-www-form-urlencoded": q.u(C.qsp(@in.read))end
                                              @cookies,@input=@k.dup,q.dup end;def service *a;@body=send @method,*a
                                                headers['Set-Cookie']=cookies.map{|k,v| "#{k}=#{C.escape(v)}"; path=#{self/
                                                  "/"}"if v!=@k[k]}-[nil];self end;def to_s;"Status: #{@status}#{Z+(headers.inject([
                                                    ]){|a,o|[*o[1]].map{|v|a<<[o[0],v]*": "if v&&v.to_s.any?};a}*Z)+Z+Z}#{@body}"end
                                                      end;X=module Controllers;@r=[];class<<self;def r;@r end;def R *u;r=@r
                                                        Class.new{meta_def(:urls){u};meta_def(:inherited){|x|r<<x}}end
                                                          def D p,m;r.map{|k|k.k.urls.map{|x|return(k.instance_method(m)rescue nil)?
                                                            [k,m,*$-[1..-1]]:[I,'r501',m]if p=~/^#{x}\\/?$/}};[I,'r404',p]
                                                              end;def M;end;constants.map{|c|k=const_get(c)
                                                                k.send:include,C,Base,Helpers,Models;@r=[k]+r if r-[k]==r
                                                                  k.meta_def(:urls){|"/#{C.downcase}"}if !k.respond_to?:urls}end end;class I<R()
                                                                    end;self end;class<<self;def goes m
                                                                      eval S.gsub(/Camping/,m.to_s),TOPEVEL_BINDING end;def escape s
                                                                        s.to_s.gsub(/[\^ \w.-]+/n){'%'+($&.unpack('H2'*$&.size)*'%').upcase}.tr ' ','+'
                                                                          end;def un s;s.tr('+','').gsub(/%(\\da-f){2}/in){[$1].pack'H*'}end
                                                                            def qsp q,d='&','y=nil,z=H[];m=proc{_,o,n|o.u(n,&m)rescue[*o]<<n}}
                                                                              q.to_s.split(/[#{d}+ */n)-[""]}.inject((b,z=z,H[])[0]){|h,p|k,v=un(p).
                                                                                split+'',2;h.u k.split(/[\]\[\]\+\/).reverse.inject(y|v){|x,i|H[i,x]},&m}end
                                                                                  def kp s;c=qsp s,'';end;def run r=$stdin,e=ENV;X.M;e=H[e.to_hash];k,m,*a=X.D e.

```


Better Science Through Art

Richard P. Gabriel
IBM Research
Hawthorne, New York USA

rpg@ $\left\{ \begin{array}{l} \text{us.ibm.com} \\ \text{dreamsongs.com} \end{array} \right.$

rpg

kjs

Kevin J. Sullivan
University of Virginia
Charlottesville, Virginia USA
sullivan.kevinj@gmail.com

Abstract

How do artists and scientists work? The same.

Categories and Subject Descriptors A.0 [General]

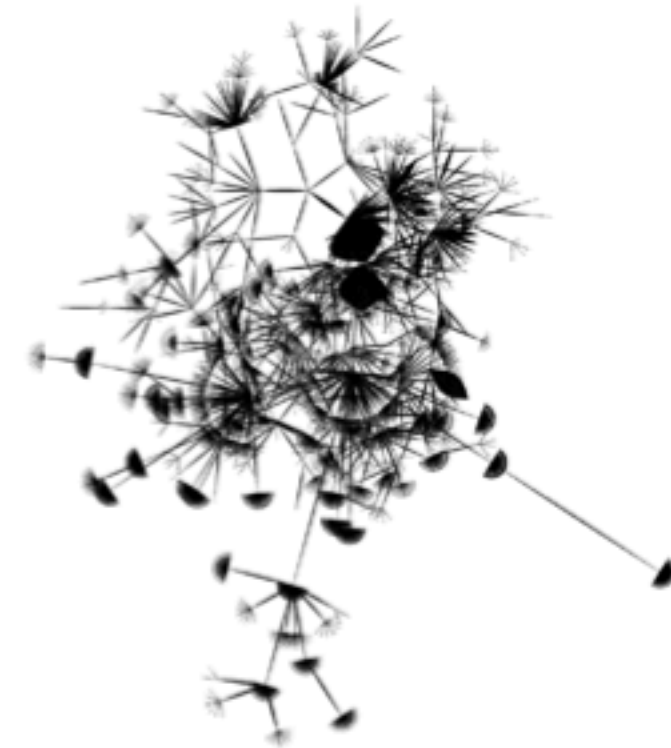
General Terms Design

Keywords Design, art, science, software engineering

~~~~~  
Prefrontal lightningbolt too lazy to chew the sphinx's loudest eyelash  
Not even if it shushes you with a mast of sneers  
Down which grateful bankvault-doors scamper  
Because of a doublejointedness that glows in the dark  
Like a soliloquy of walnuts  
Numbed by beaks of headless measuringtape  
So the lubriciousness can tower in peace  
Like a buzzsaw trapped in a perfumery of shrugs  
Lemon  
Or lime  
Only a maze can remember your hair of buttered blowguns

~~~~~  
From *Nights of Naomi* by Bill Knott [1]

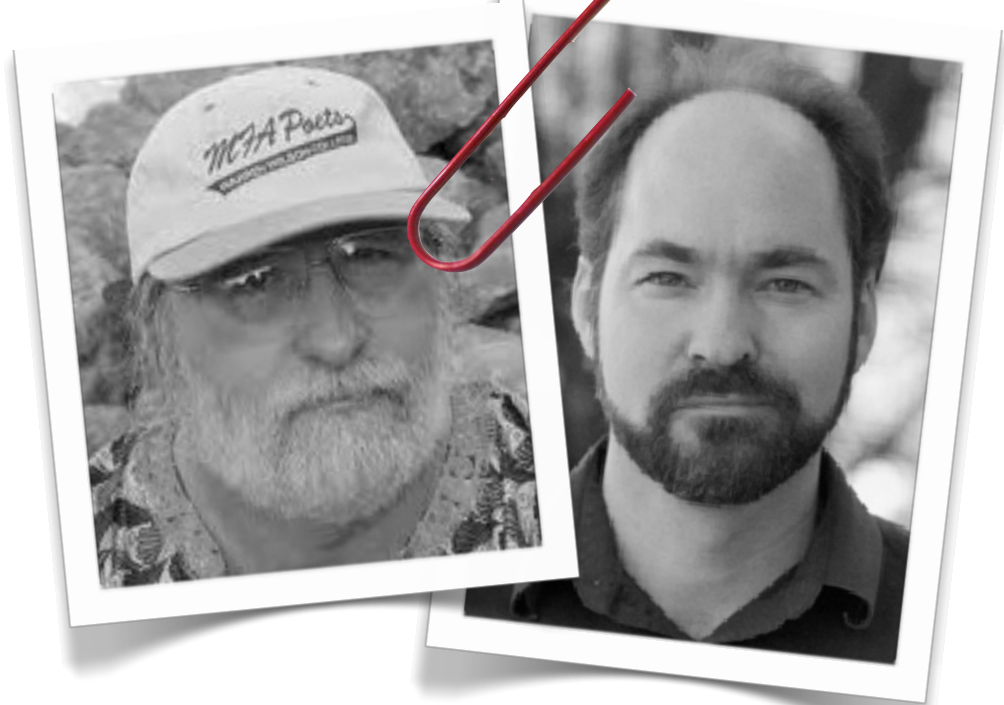
~~~~~  
Art is strange. Art cannot be understood. The poet Robert Browning is reported to have said of a passage he wrote:



This is a sort of contemporary science which is more easily understood than the nonsensical poem.

Don't you think? Ah, but with training this is quite easily understood. The poem, however—obviously no amount of training, teaching, or learning will bring you to understand it.

Some other examples: a visual representation of a source code svn log shows, clearly, the history of the program, whereas the Jackson Pollock painting is clearly less understandable



*"Art is, in part, the practice of intense and disciplined noticing of details, of life, of how the world works and in that way at least it resembles science."*

*Richard P. Gabriel*  
*Kevin J. Sullivan*

**What are we?**

**What are programmers?**

**Are we  
Scientists?**

# Scientists

create knowledge

are problem driven

seek to understand and explain

design experiments to test the  
theories

prefer abstract knowledge

rely on tacit knowledge



Are we  
Engineers?

# Engineers

create knowledge

are problem driven

seek to understand and explain

design experiments to test the  
theories

prefer contingent knowledge

rely on tacit knowledge

Are we  
Artists?

# Artists

create knowledge, language,  
alternative reality

are image or “piece” driven

seek to understand and explain

design devices (their pieces) to  
explore the world and the self

prefer contingent knowledge

rely on tacit knowledge

# Scientists

create knowledge

are problem driven

seek to understand and explain

design experiments to test the theories

prefer abstract knowledge

rely on tacit knowledge

# Engineers

create knowledge

are problem driven

seek to understand and explain

design experiments to test the theories

prefer contingent knowledge

rely on tacit knowledge

# Artists

create knowledge, language,  
alternative reality

are image or “piece” driven

seek to understand and explain

design devices (their pieces) to  
explore the world and the self

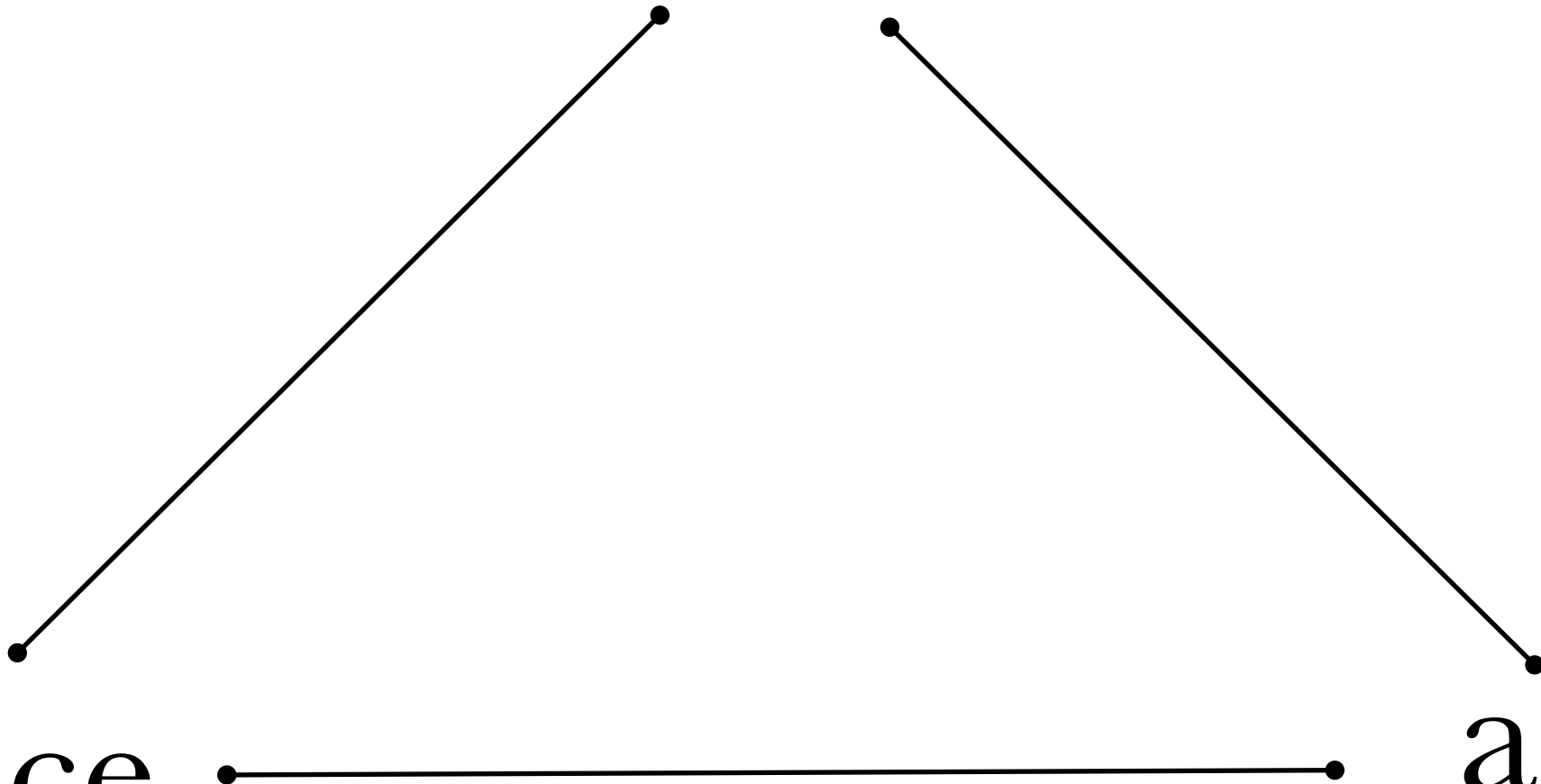
prefer contingent knowledge

rely on tacit knowledge

engineering

science

art

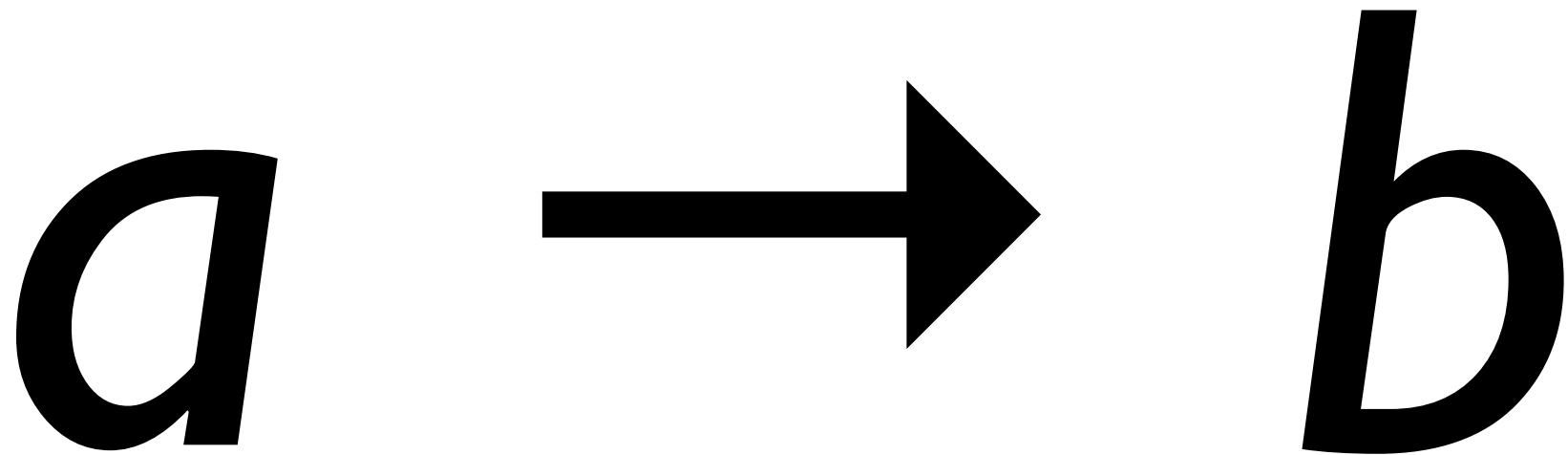


How do we create knowledge?



*a* → *b*

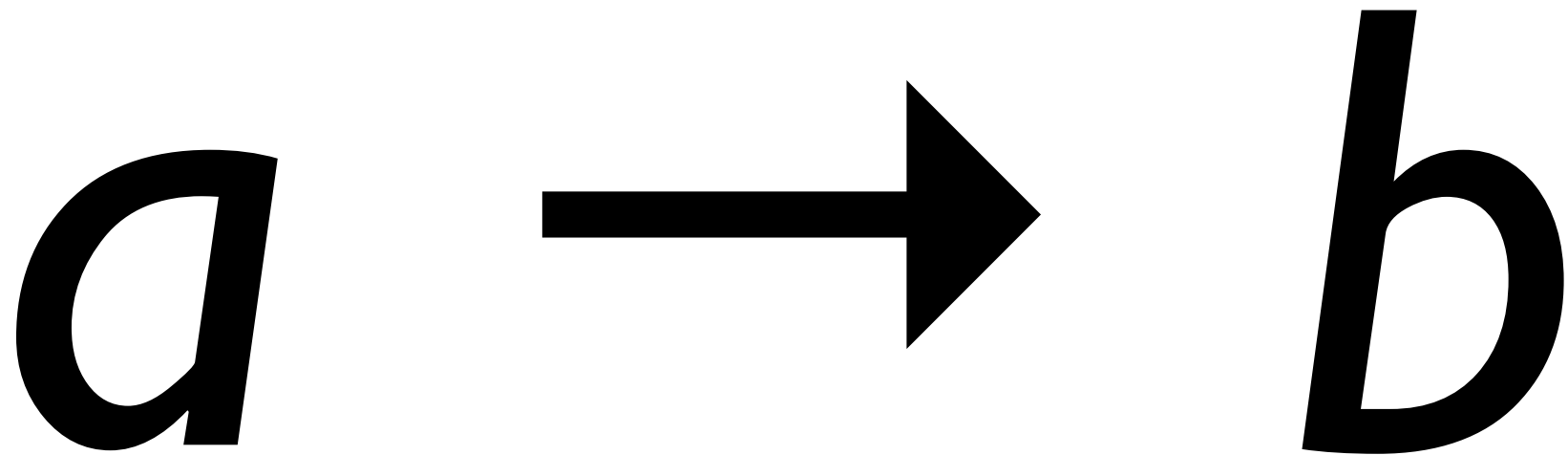
Deduction



Induction

*a* → *b*

Abduction



Guessing!?

# Conceptual and Experimental

Concept

Experiment

Science

mathematician

Einstein

waterfall

craftsperson

Darwin

engineer

Art

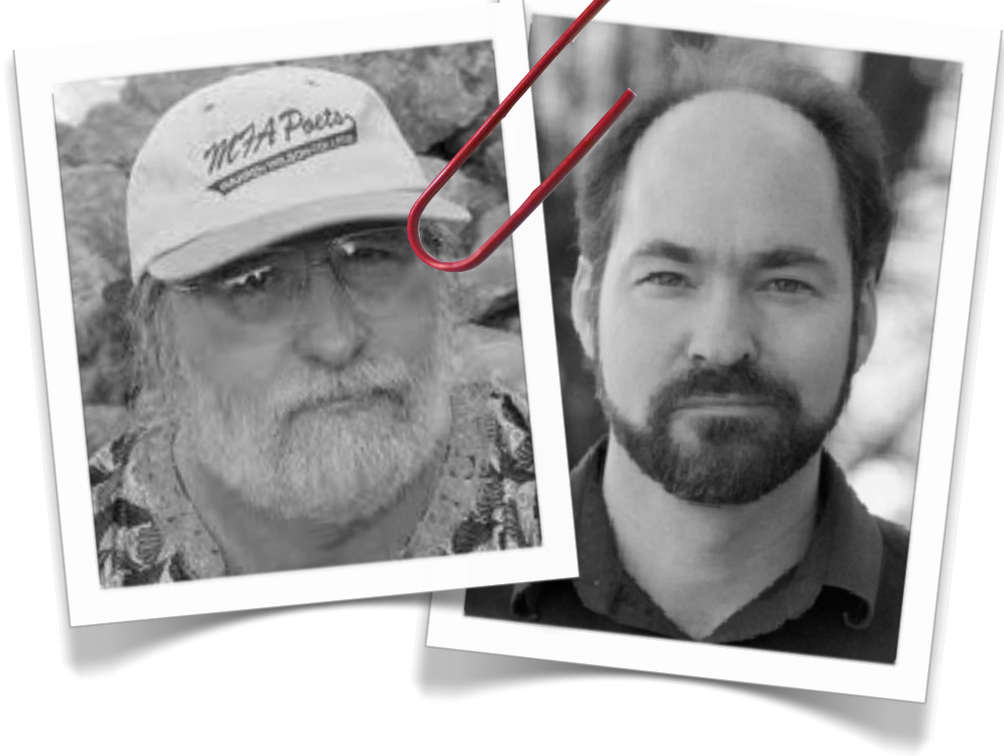
Warhol

Picasso

1970s Hacker

agile

live coders



*“This is the full process of art—**exploration, discovery**, and then **verification or understanding**—and it’s also the full process of science. Naturally, it’s not a linear process but one with cycles of explore-discover-understand within any of its steps.”*

*Richard P. Gabriel  
Kevin J. Sullivan*





*“If we attempt to map the world of a story before we explore it, we are likely either to  
(a) prematurely limit our exploration, so as to reduce the amount of material we need to consider, or  
(b) explore at length but, recognising the impossibility of taking note of everything, and having no sound basis for choosing what to include, arbitrarily omit entire realms of information.  
The opportunities are overwhelming.”*

*Peter Turchi*









# Hammock Driven Development

## *cheatsheet*

Now  
Easy As  
PIE

### ***Preparation***

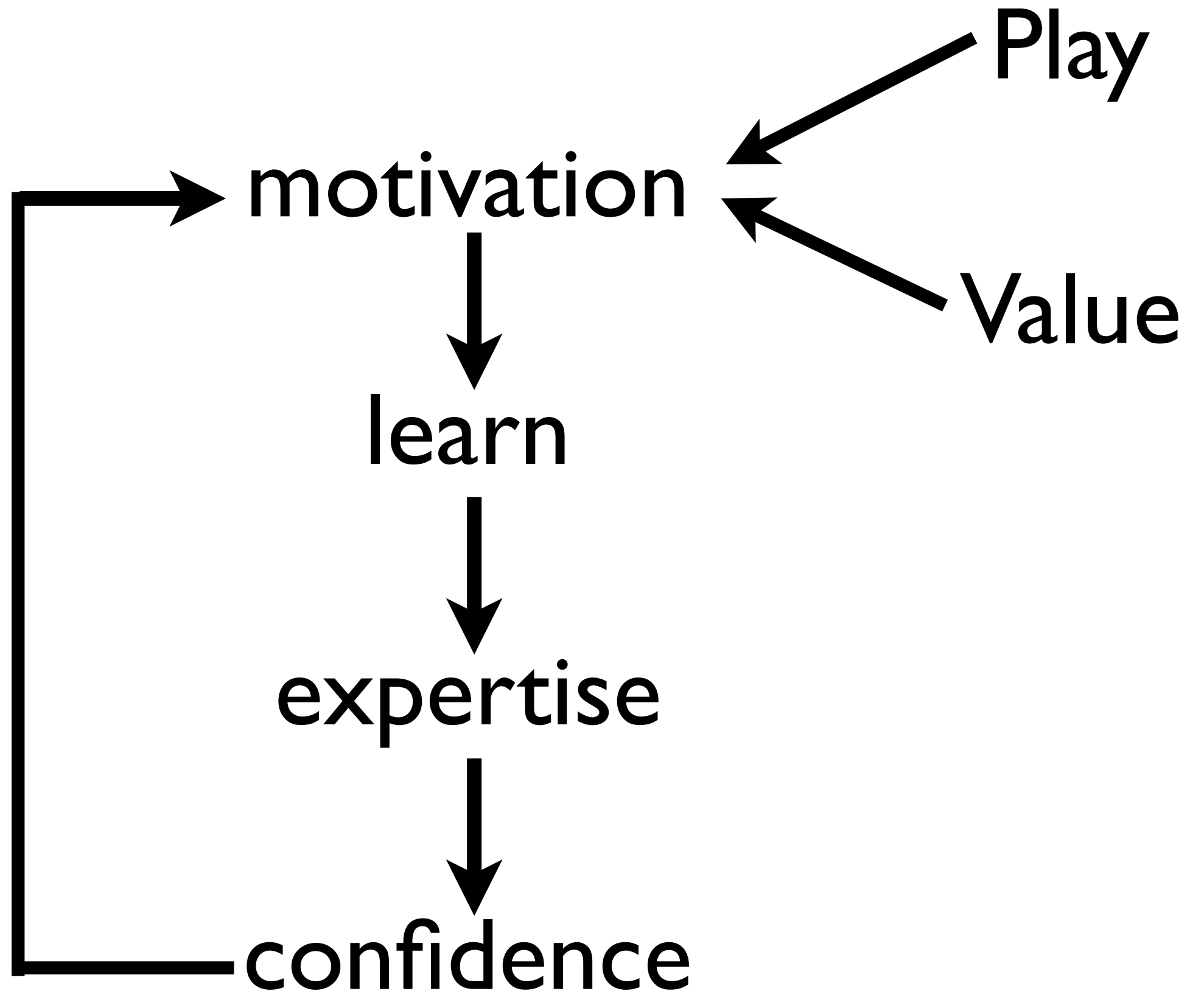
- state the problem
- what do you know?
- what don't you know?
- study related problems and their solutions and evaluate them
- then step away from the computer, find a hammock, and think

### ***Inspiration***

zzzzzzzz

### ***Evaluation***

- figure out what's good about the solution
- figure out the problems with the solution, and solve them too
- find alternative solutions
- determine trade offs between alternatives











*“Play appears to be the bonobos’ key to problem-solving, creativity, forming connections, and avoiding conflict.”*

*Isabel Behncke*





He travelled

SPACE & TIME

in a bell jar of his own  
careful workmanship...

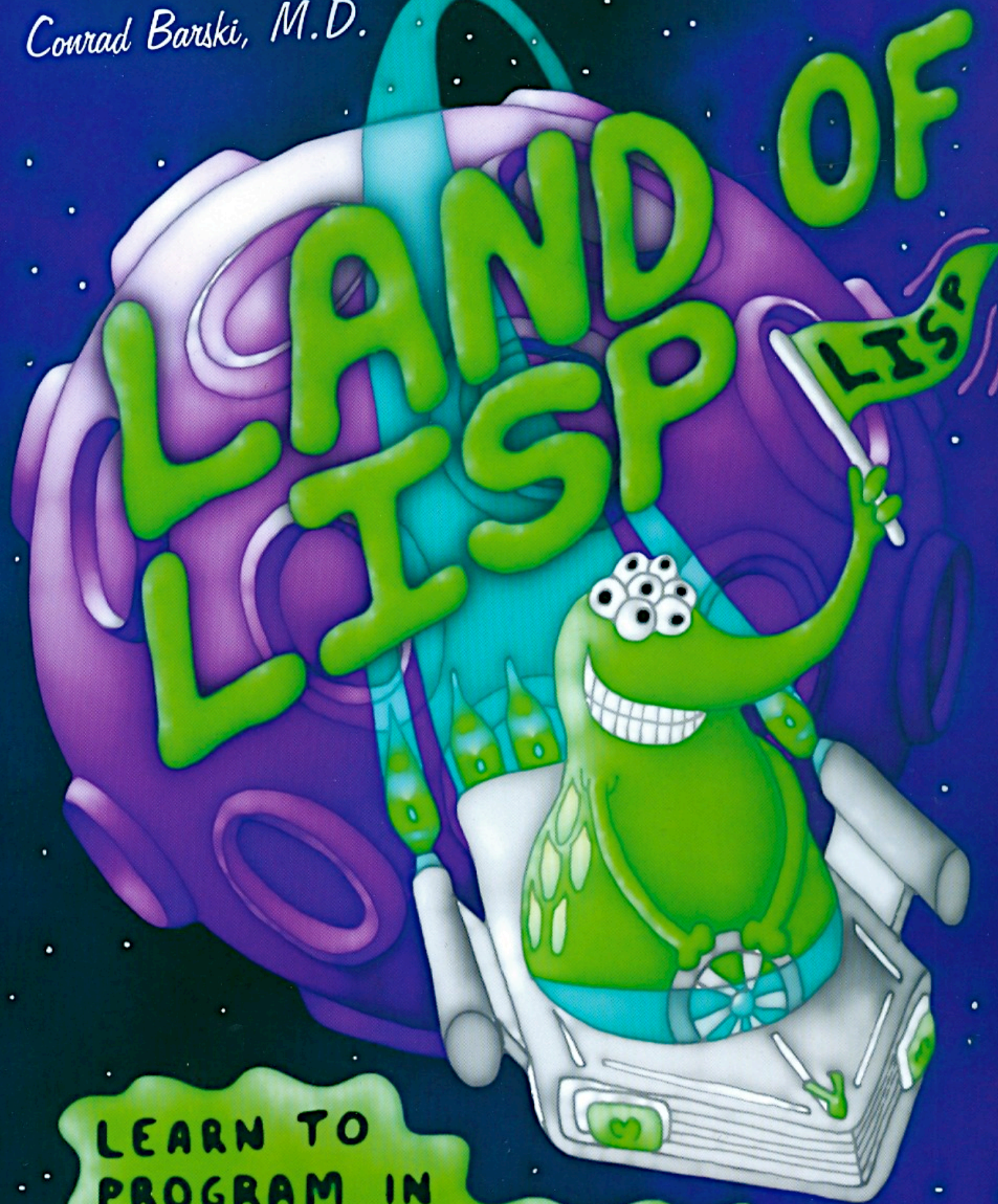
to find himself...  
to find truth.

PRESENTING  
Dr. N. Harold Cham





Conrad Barski, M.D.



LEARN TO  
PROGRAM IN  
LISP, ONE GAME AT A TIME!



λ)))



```
(defn repeater  
  ([col] (repeater col 4 0.8))  
  ([col n] (repeater col n 0.8))  
  ([col n factor]  
    (let [base (take n (repeatedly #(choose  
      (loop [base base
```

## RUBY AND RAILS

EUROPEAN CONFERENCE 2010

1-22 OCTOBER

RAR10

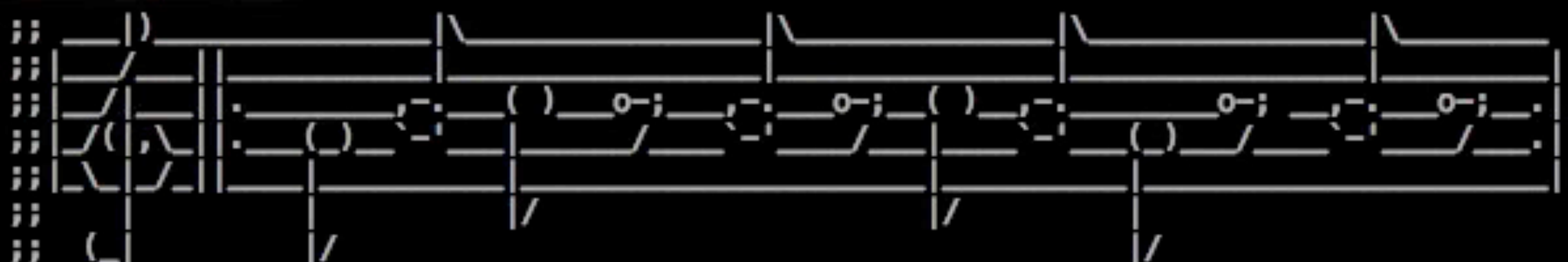
HOLDERREDUC  
business your online umbrella

Kabisa IC

nedap  
HEALTHCARE



```
—[#1]—  
(plucked-string)
```



```
(def reich-degrees [:vi :vii :i+ :_ :vii :_ :i+ :vii :vi :_ :vii :_])  
(def pitches (degrees->pitches reich-degrees :diatonic :C4))
```

```
(defn play  
  [time notes sep]  
  (let [note (first notes)]  
    (when note  
      (at time (plucked-string note))))  
  
  (let [next-time (+ time sep)]  
    (apply-at next-time)))
```

|         |   |
|---------|---|
| next    | 1 |
| neg     | 1 |
| neg?    | 1 |
| newline | 1 |
| next-id | 1 |



MADE IN  
ITALY

PH-2

3

10

5

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

RX

PWR

AREF

GND

3

2

1

0

9

8

7

6

5

4

3

2

1

0

TX

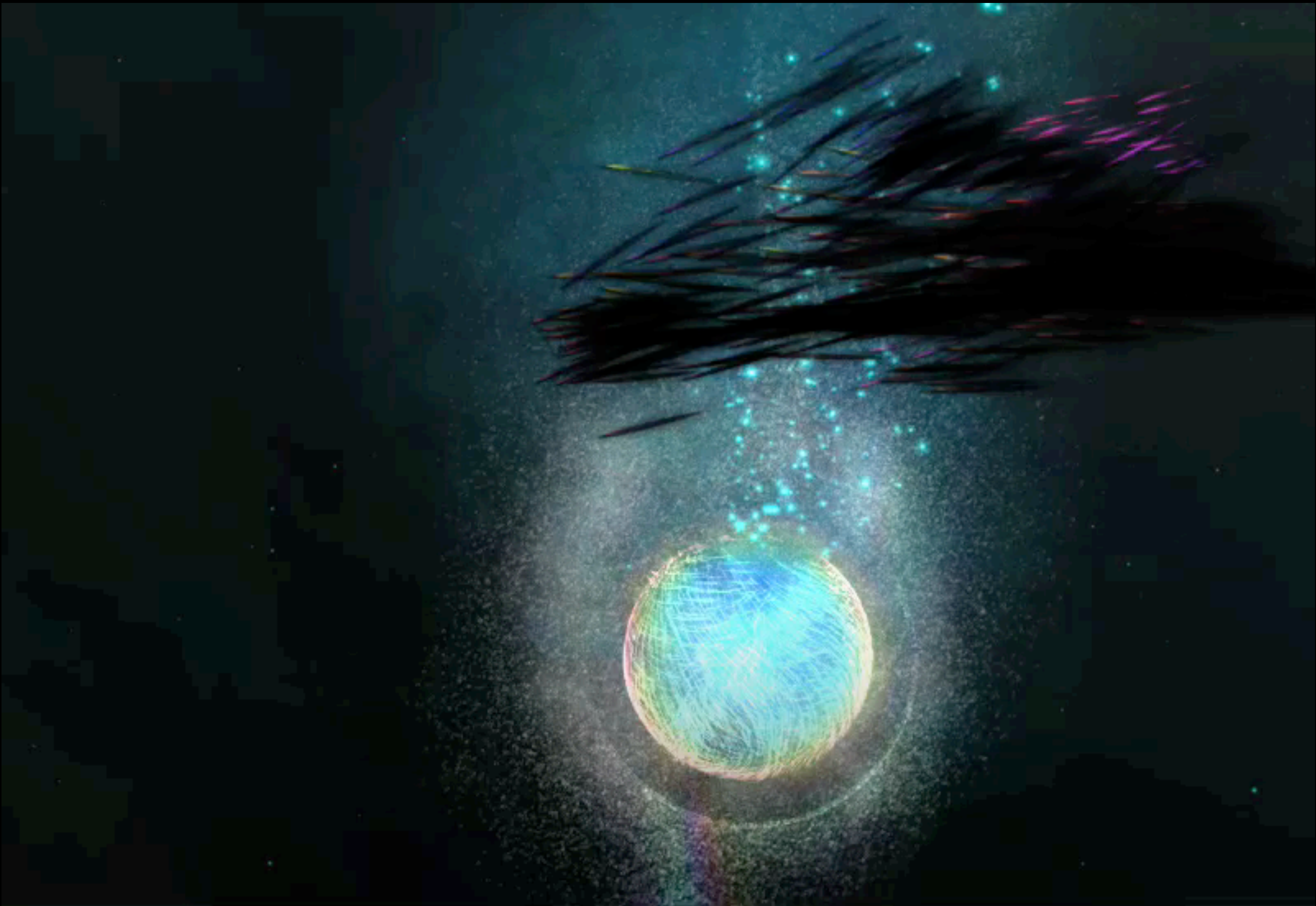
RX

PWR

AREF









# SCRATCH

New

Open

Save

Save As

Share!

Undo

Language

Extras

Want Help?



Motion

Control

Looks

Sensing

Sound

Numbers

Pen

Variables



Sprite1

Export

x: 2

y: 17

direction: 90



Scripts

Costumes

Sounds

New costume:

Paint

Import

1



costume1

95x111

3 KB

Edit

Copy



2



costume2

95x111

3 KB

Edit

Copy



when clicked

when space key pressed

when Sprite1 clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if

else





mouse x: 31  
mouse y: -196



Stage



Sprite1  
2 costumes

 Cancel  
 Select Tile







We need to cultivate  
*tacit knowledge*  
*practice & discipline* and  
being able to  
*notice deeply*  
as artists do

# Thank you

@samaaron

<http://sam.aaron.name>