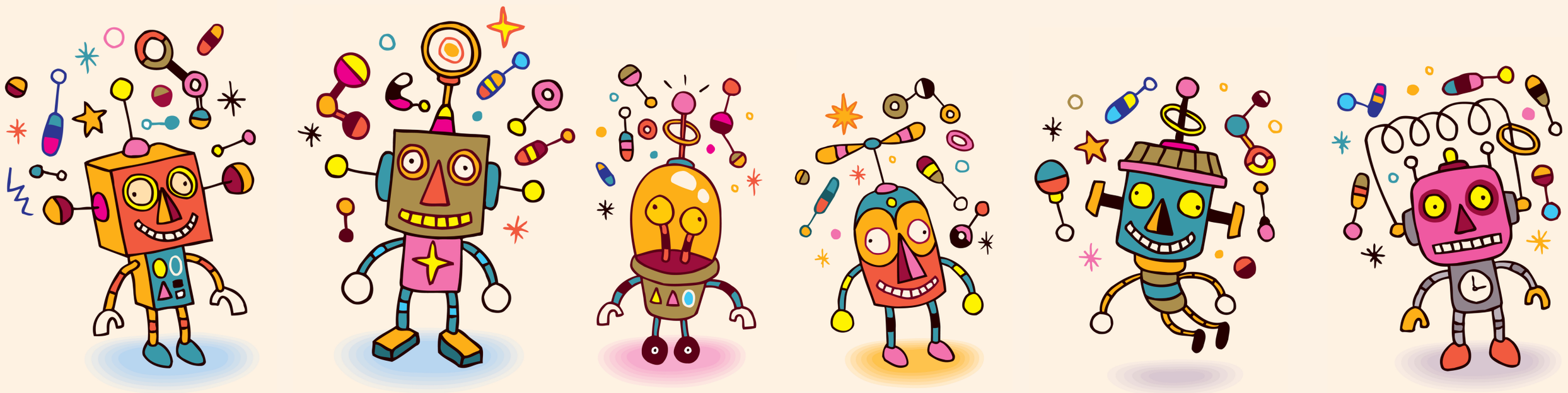
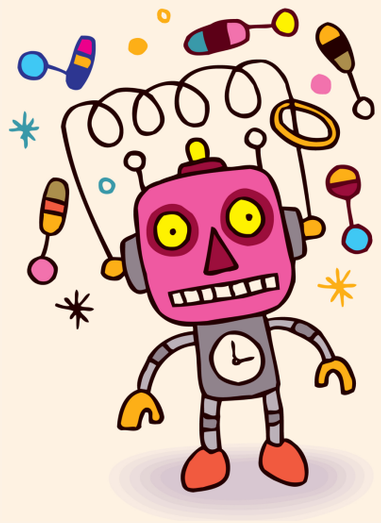
CREATIVE MACHINES



Joseph Wilk

ROMANCE

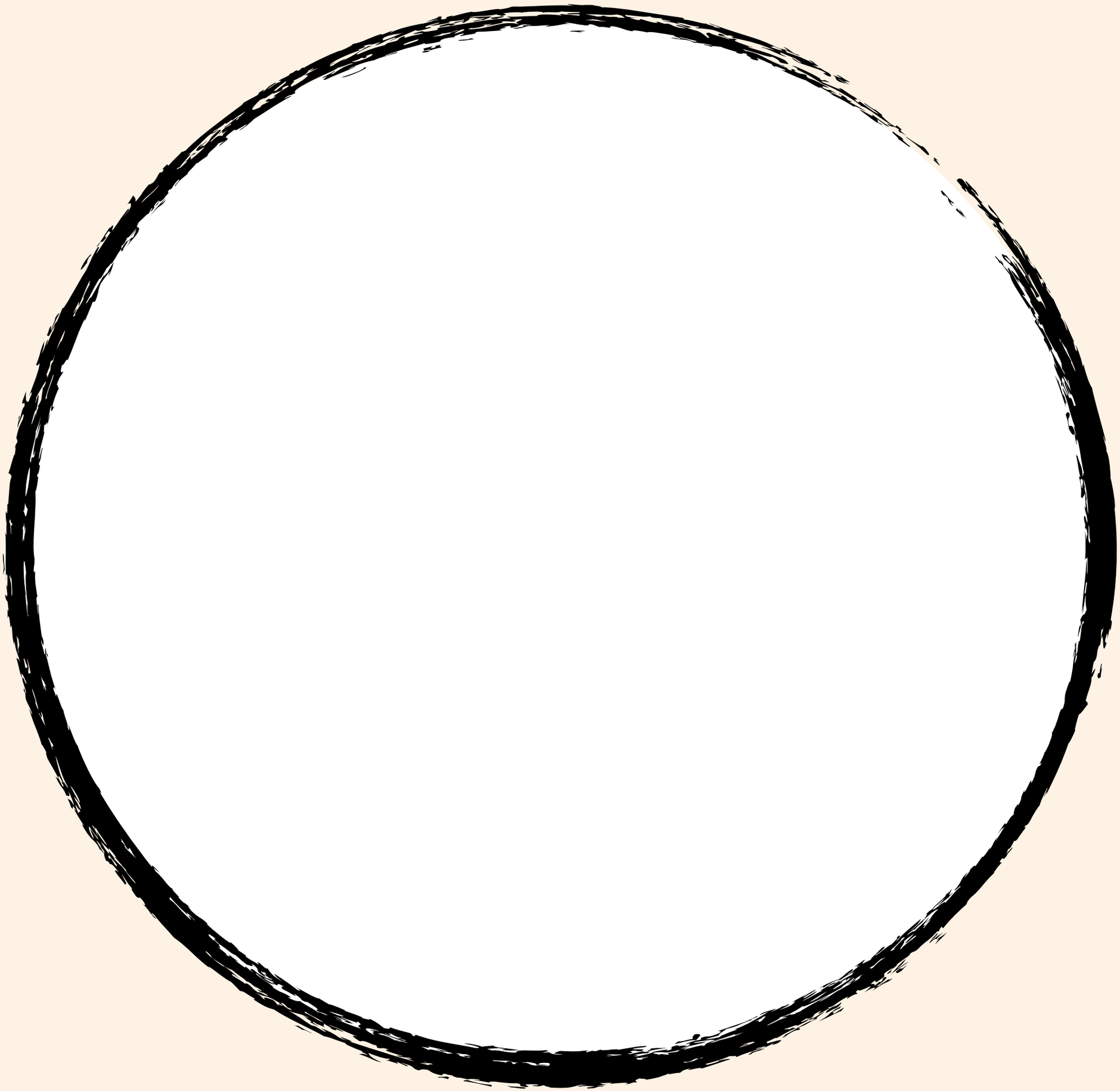
CREATIVITY



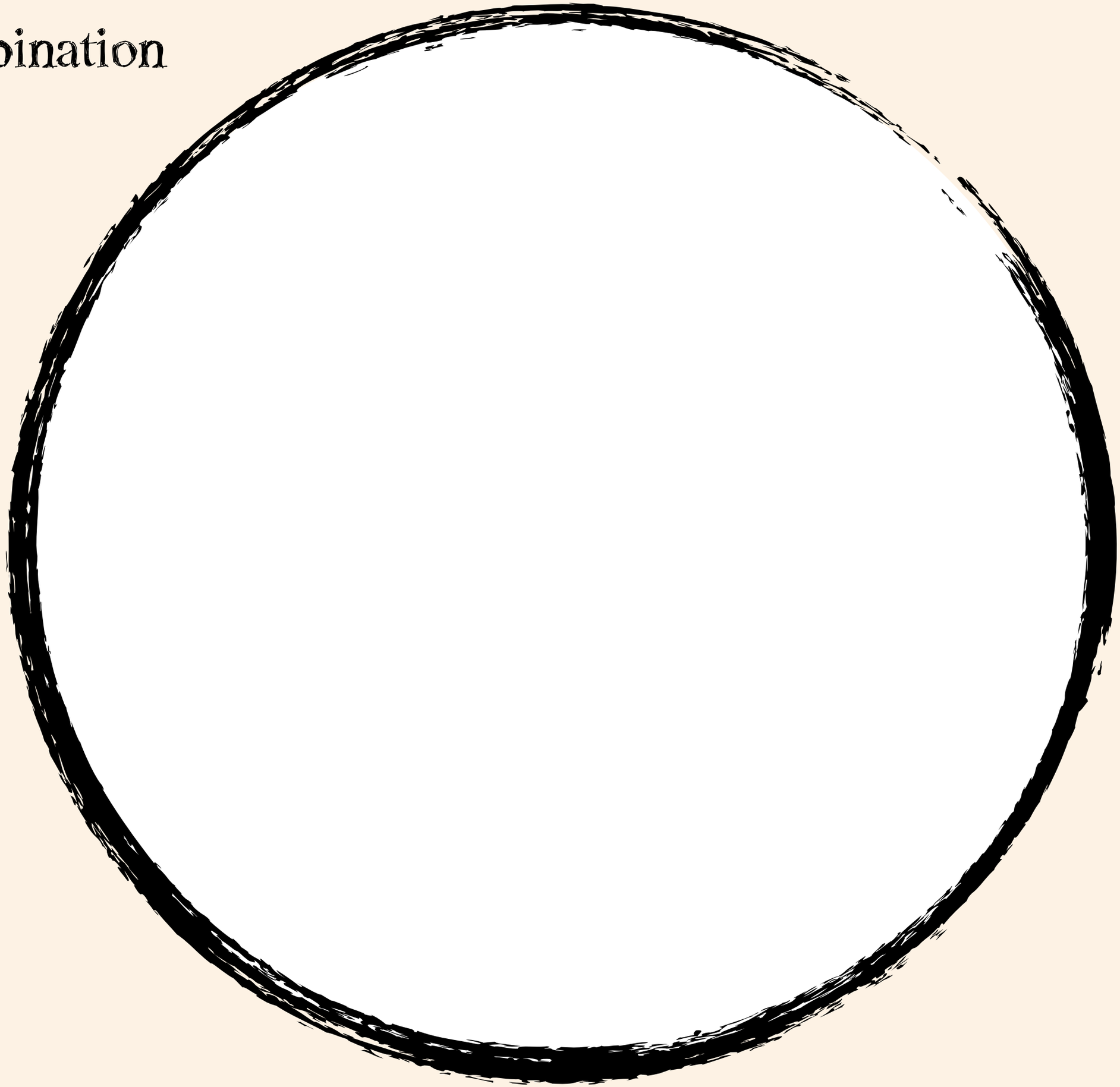
Creativity

Psychological

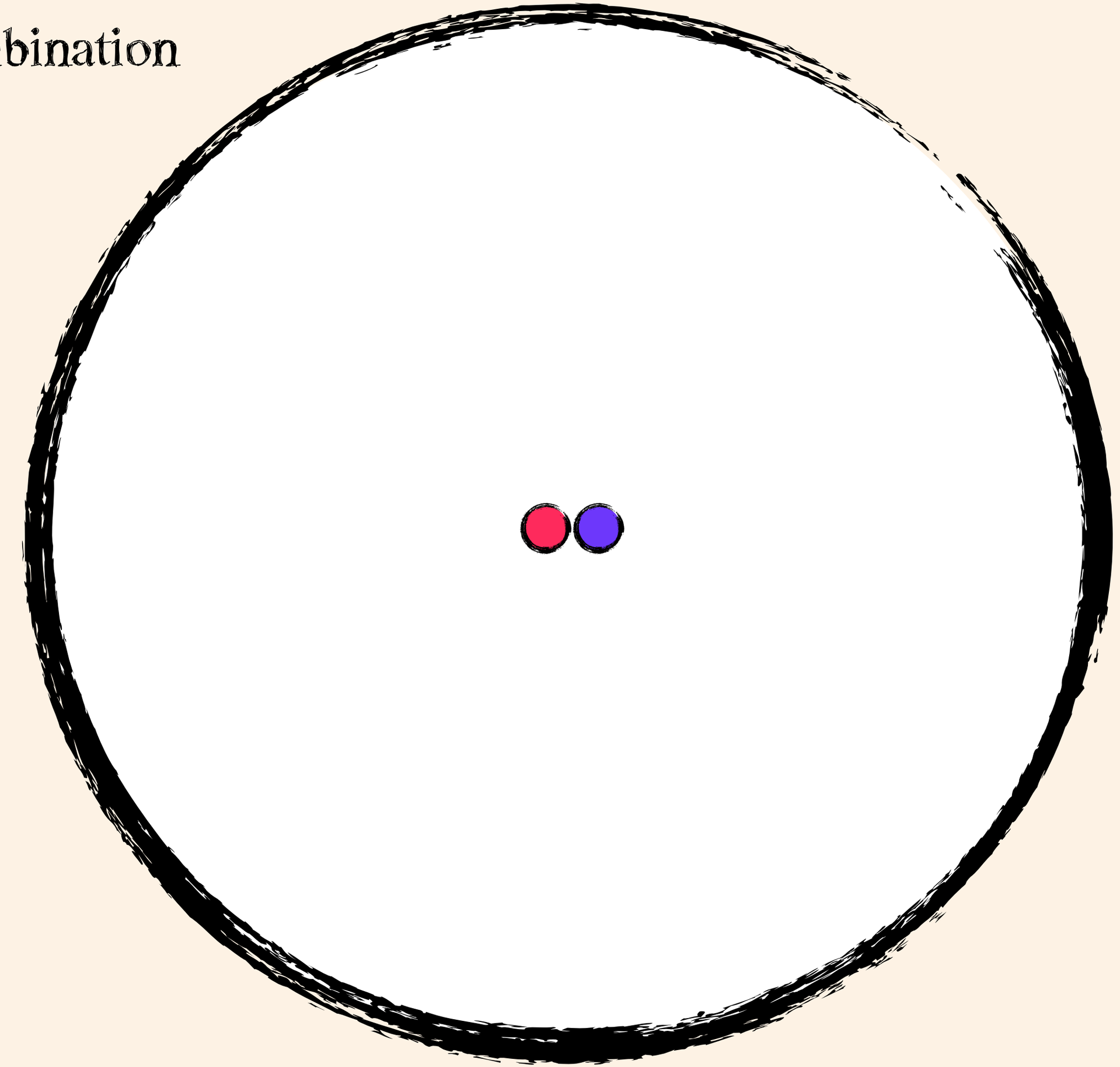
Historical



Combination

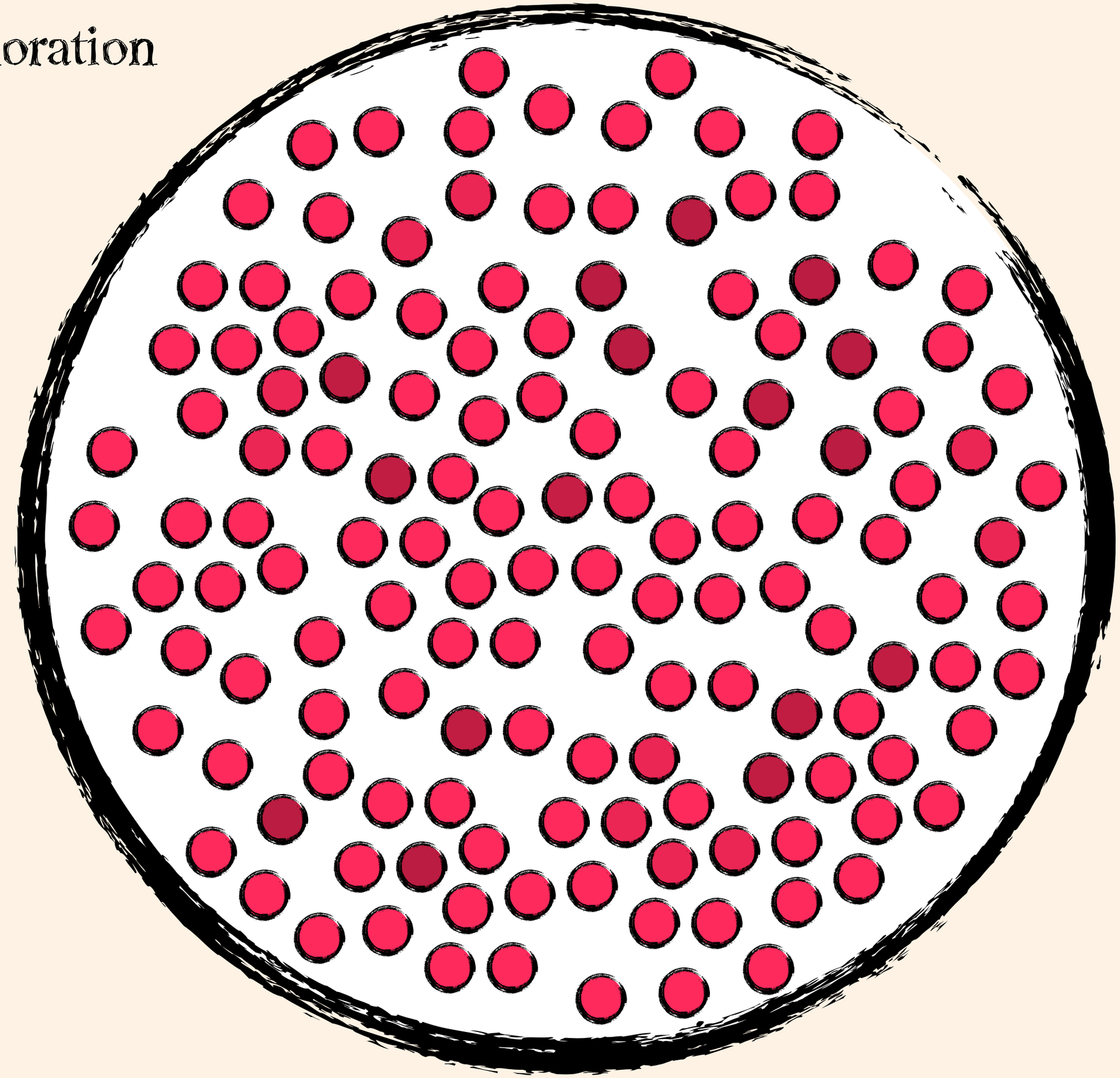


Combination

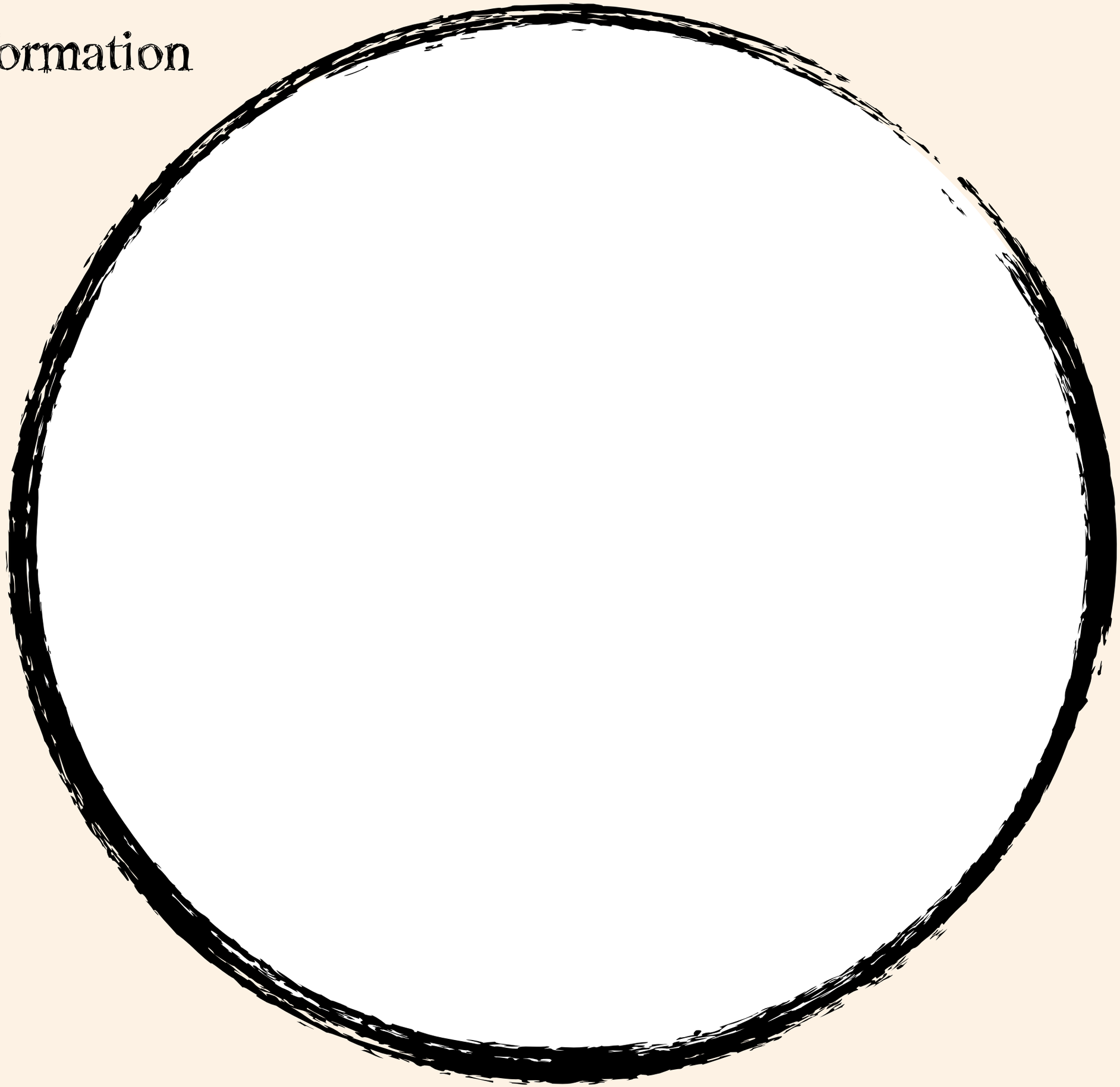


Exploration

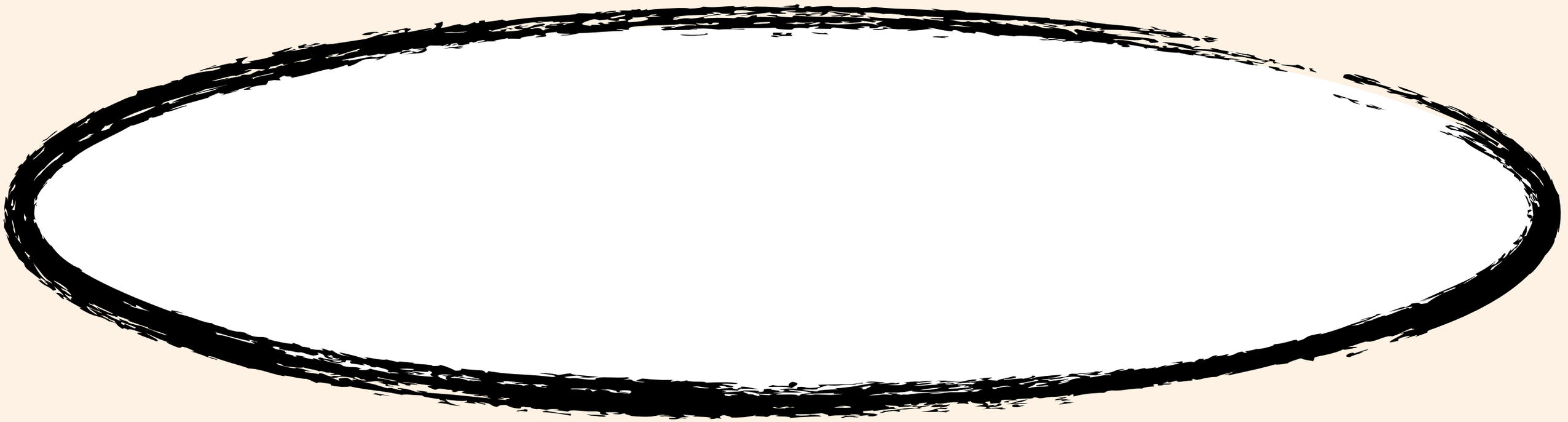
Exploration



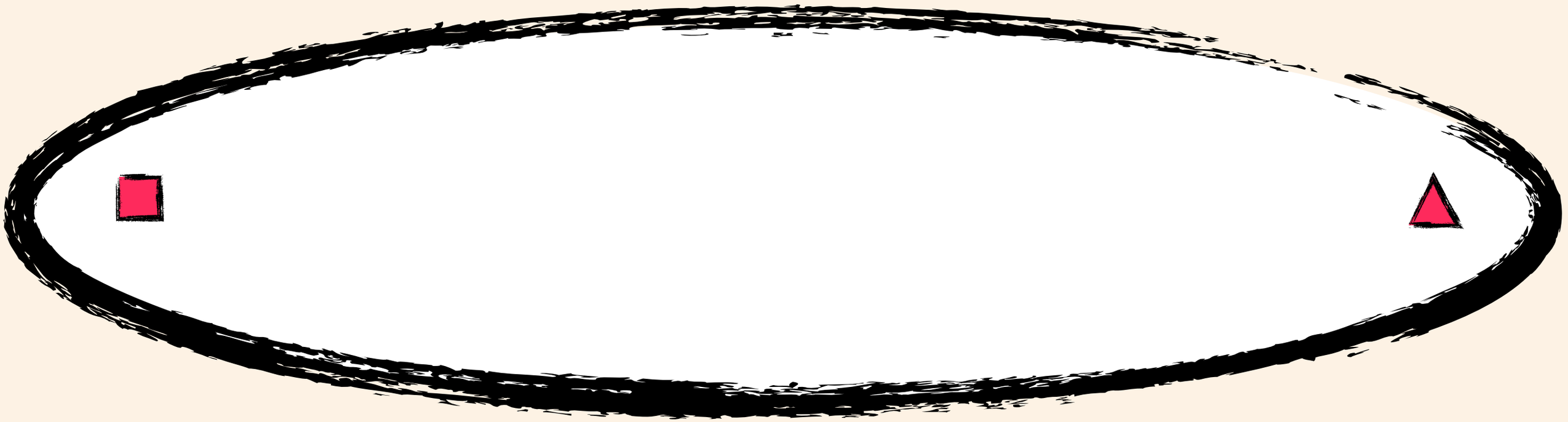
Transformation



Transformation



Transformation

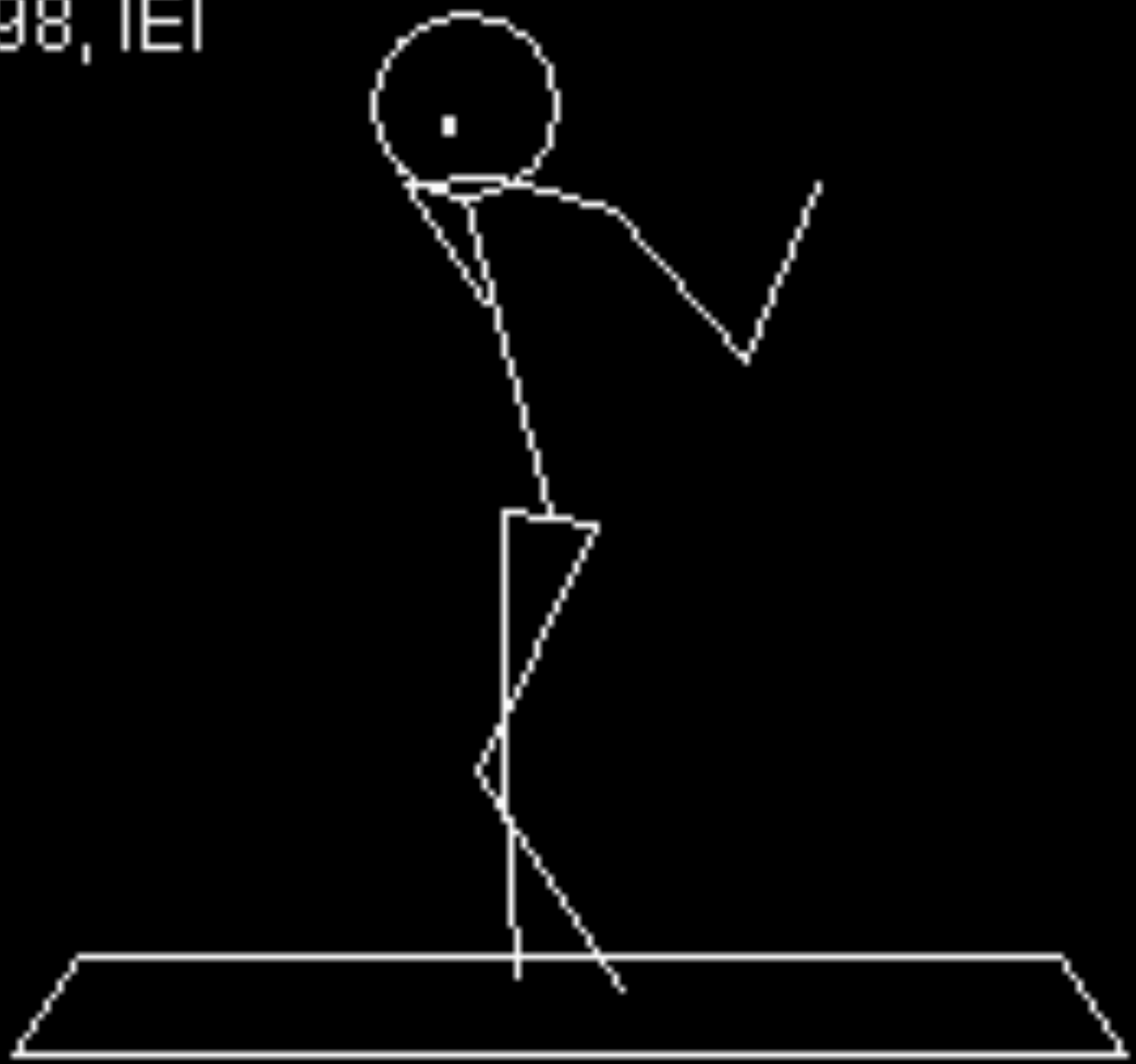


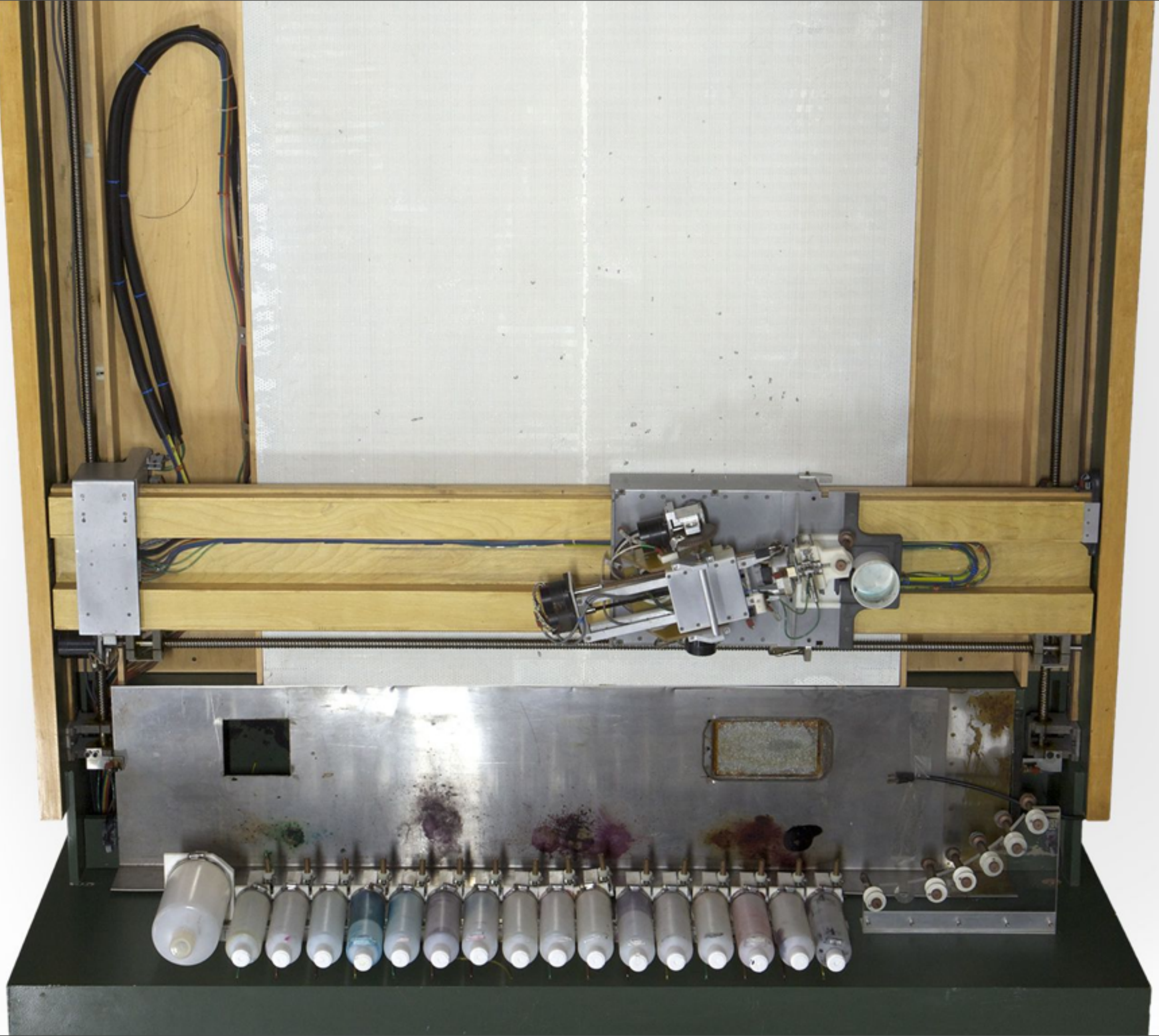
Creative Machines

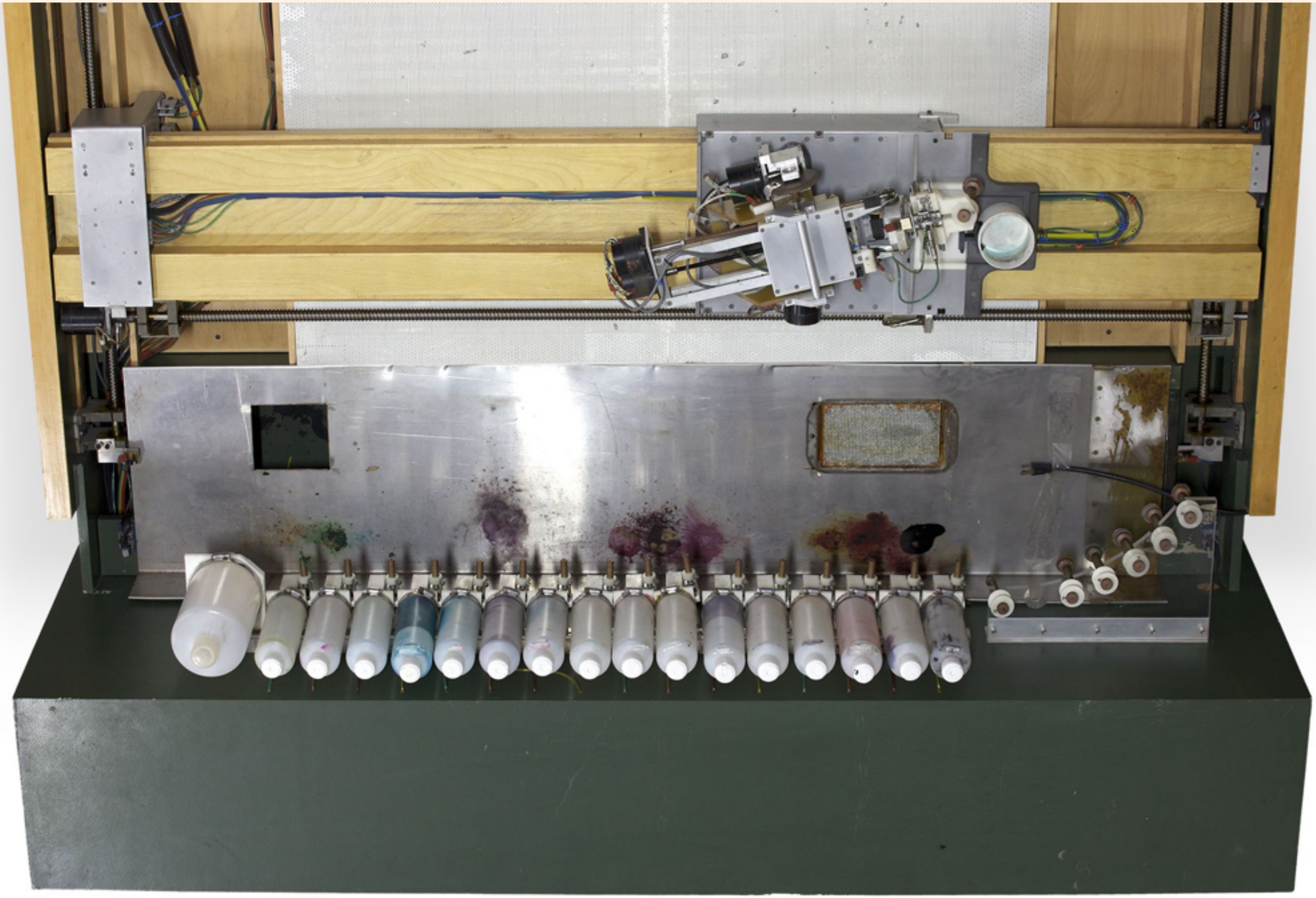
Death = Entropy

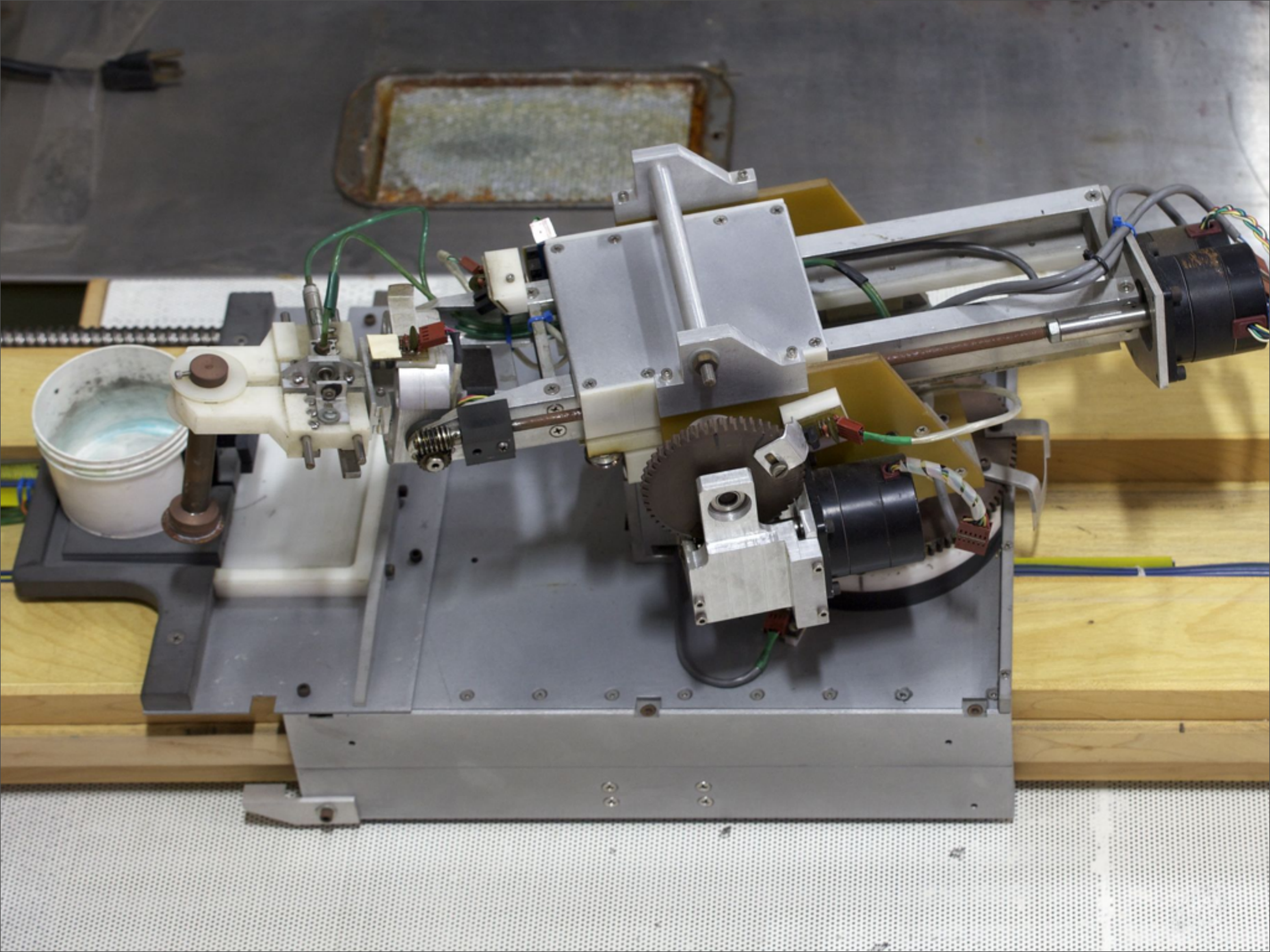
Chaos*

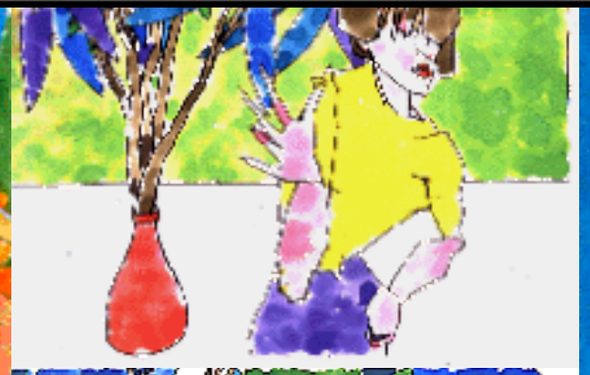
Death Dance
(c) 1998, IEI











MUSIC

COMPLEXITY

#Computer models of musical creativity

Experiments with models for computers generating music.

##Examples

```clojure
(require '[musical-creativity.musician :as musician] :reload)
(use '[musical-creativity.instruments] :reload)

;Cosine
(require '[musical-creativity.composers.cosine :as cosine] :reload)
(musician/play (cosine/compose) piano)

;Cellular automata
```

-UU-:----F1 README.md   Top (8,61)   Git-master (Clojure Paredit nREPL ElDoc

```
; nREPL 0.1.8
user>
```

-UUU:**--F1 *nrepl*   All L2   (nREPL Paredit ElDoc) ----------------------

Capture Human Creativity

```
(musician/play (ca/compose) organ)
(musician/play (ca/compose ca/rule-22) sawish)

;Sonify data
(require '[musical-creativity.composers.sonify-data :as sonify-data] :reloa\
d)
(musician/play (sonify-data/compose) piano)

;Sonify text
(require '[musical-creativity.composers.sonify-words :as sonify-words] :rel\
oad)
(musician/play (sonify-words/compose "Strange Loop Strange Loop") piano)
(musician/play (sonify-words/compose "potato potato") ping)

;Network (Using adaptive resonance theory)
(require '[musical-creativity.composers.network :as network] :reload)
-UU-:----F1  README.md        14% (27,59)   Git:master  (Clojure Paredit nREPL

user> (overtone.live/stop)
nil



user>
-UUU:**--F1  *nrepl*         Bot L84    (nREPL Paredit ElDoc) ---------------
nil
```

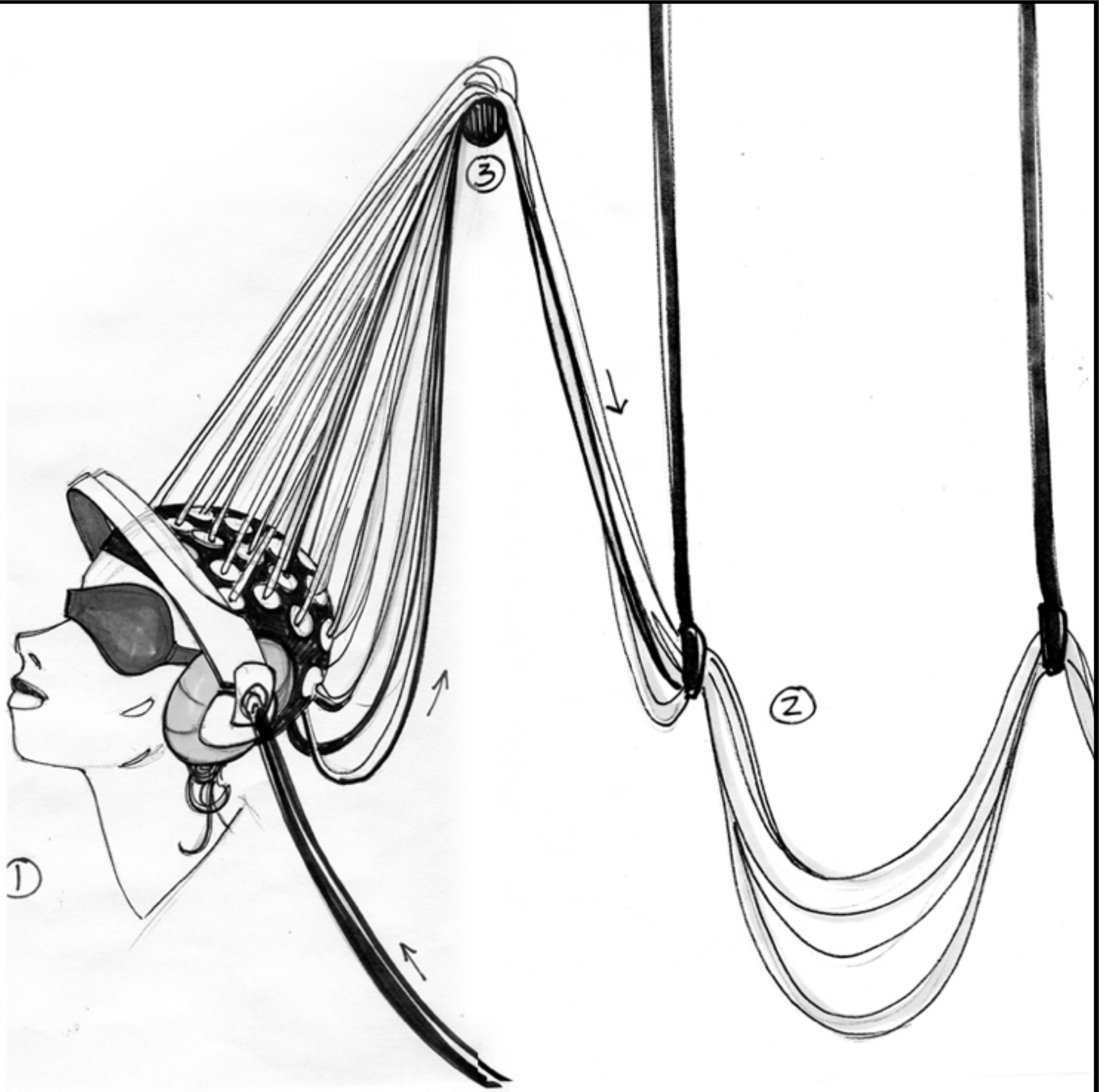
Delta

Theta

Alpha

Beta

Gamma



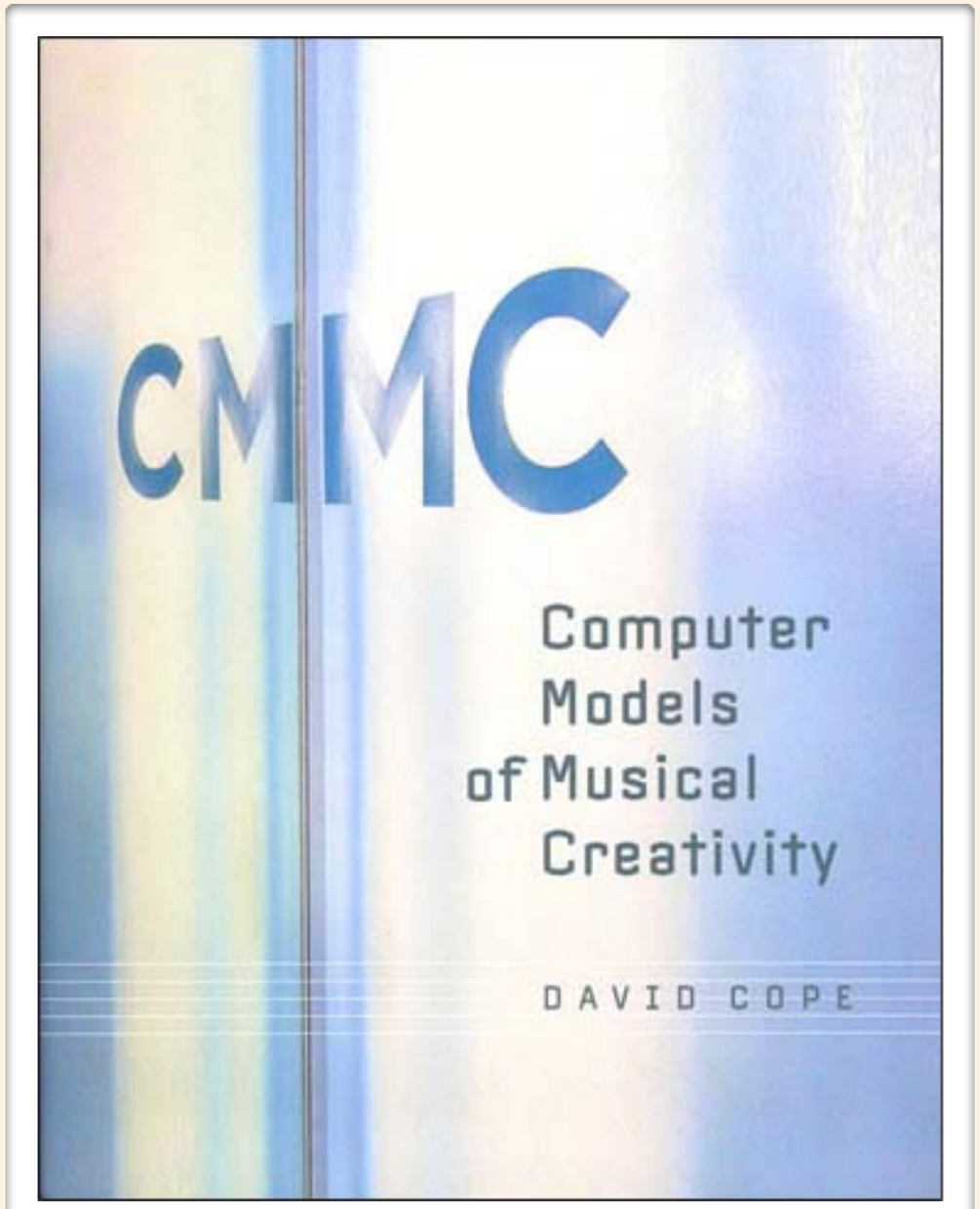

I'M
SUPER
Test subject: X

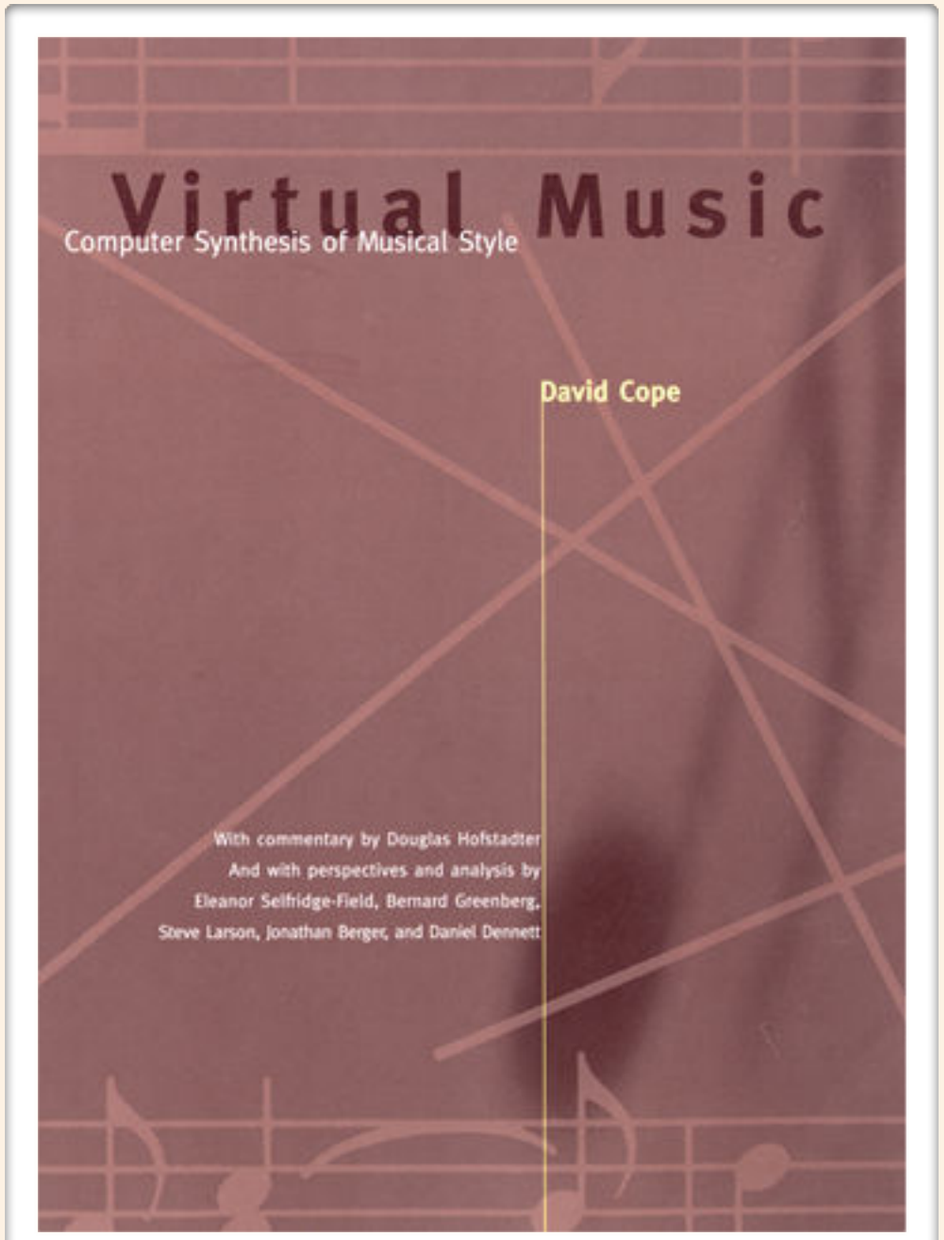
NeuroSky

HUMAN
BRAIN

```
        s = {'timestamp': t,
             'meditation': meditation,
             'attention': attention,
             'raw_spectrum': spectrum,
             'delta_waves': waves_vector[0],
             'theta_waves': waves_vector[1],
             'alpha_waves': (waves_vector[2]+waves_vector[3])/2,
             'low_alpha_waves': waves_vector[2],
             'high_alpha_waves': waves_vector[3],
             'beta_waves': (waves_vector[4]+waves_vector[5])/2,
             'low_beta_waves': waves_vector[4],
             'high_beta_waves': waves_vector[5],
             'gamma_waves': (waves_vector[6]+waves_vector[7])/2,
             'low_gamma_waves': waves_vector[6],
             'mid_gamma_waves': waves_vector[7]}

             s = json.dumps(s)
             fp.write(s)
        gevent.sleep(0.4)
```


CMMC
Computer
Models
of Musical
Creativity
DAVID COPE


Virtual Music
Computer Synthesis of Musical Style
David Cope
With commentary by Douglas Hofstadter
And with perspectives and analysis by
Eleanor Selfridge-Field, Bernard Greenberg,
Steve Larson, Jonathan Berger, and Daniel Dennett

An Integrated Model Of Musical Creativity

Allusion

"Creativity relies in part on the juxtaposition of allusions to the work of others"

David Cope

```
(ns musical-creativity.allusion.sorcerer
  (:require [musical-creativity.util :refer :all]
            [musical-creativity.events :as events]))

(def *choices* '(chopin-1 bach-1 bach-2 bach-3 bach-4 beethoven-1 beethoven\
-2 boccherini-1 chausson-1 schumann-1))

(def chopin-1 '((0 59 1000 1 64)(1000 64 1500 1 64)(2500 63 500 1 64)(3000 \
64 500 1 64)
                (3500 67 500 1 64)(4000 71 1000 1 64)(5000 71 1000 1 64)(60\
00 71 1000 1 64)(7000 76 1000 1 64)
                (8000 75 500 1 64)(8500 76 500 1 64)(9000 78 750 1 64)(9750\
 76 250 1 64)(10000 76 1000 1 64)
                (11000 75 1000 1 64)))
```

-UU-:**--F1 sorcerer.clj   Top (1,0)   Git-master (Clojure Paredit nREPL

```
; nREPL 0.1.8
user>
```

-UUU:**--F1 *nrepl*   All L2   (nREPL Paredit ElDoc) ----------------

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

https://github.com/josephwilk/musical-creativity/blob/master/src/musical_creativity/allusion/sorcerer.clj

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

Chopin

Bach

Bach

Bach

Beethoven

Beethoven

Boccherini

Chausson

Schumann

RECOMBINANCE

"Creativity does not originate from a vacuum"

David Cope

Composition

beat beat beat beat beat beat beat beat beat

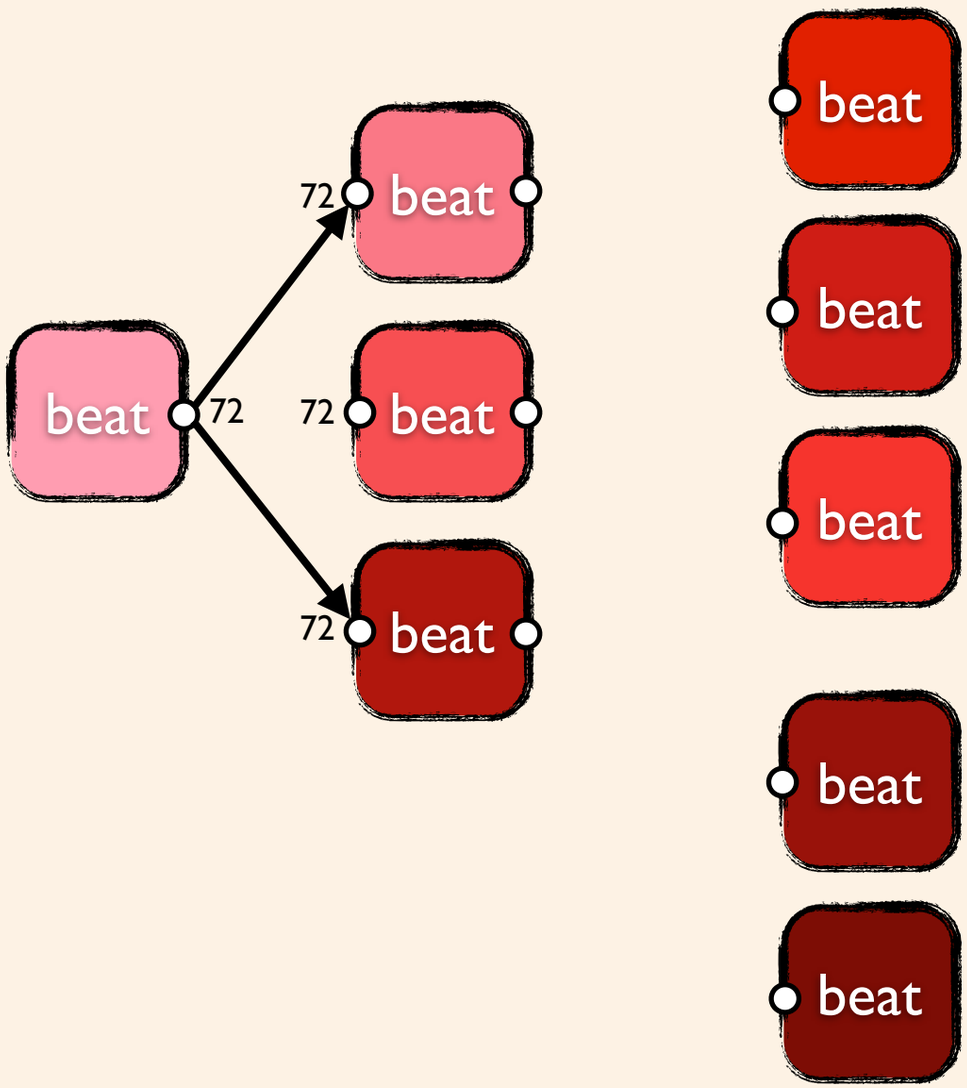
beat beat beat beat beat beat beat beat beat

beat

beat beat beat beat beat beat beat beat

beat beat beat beat beat beat beat beat beat

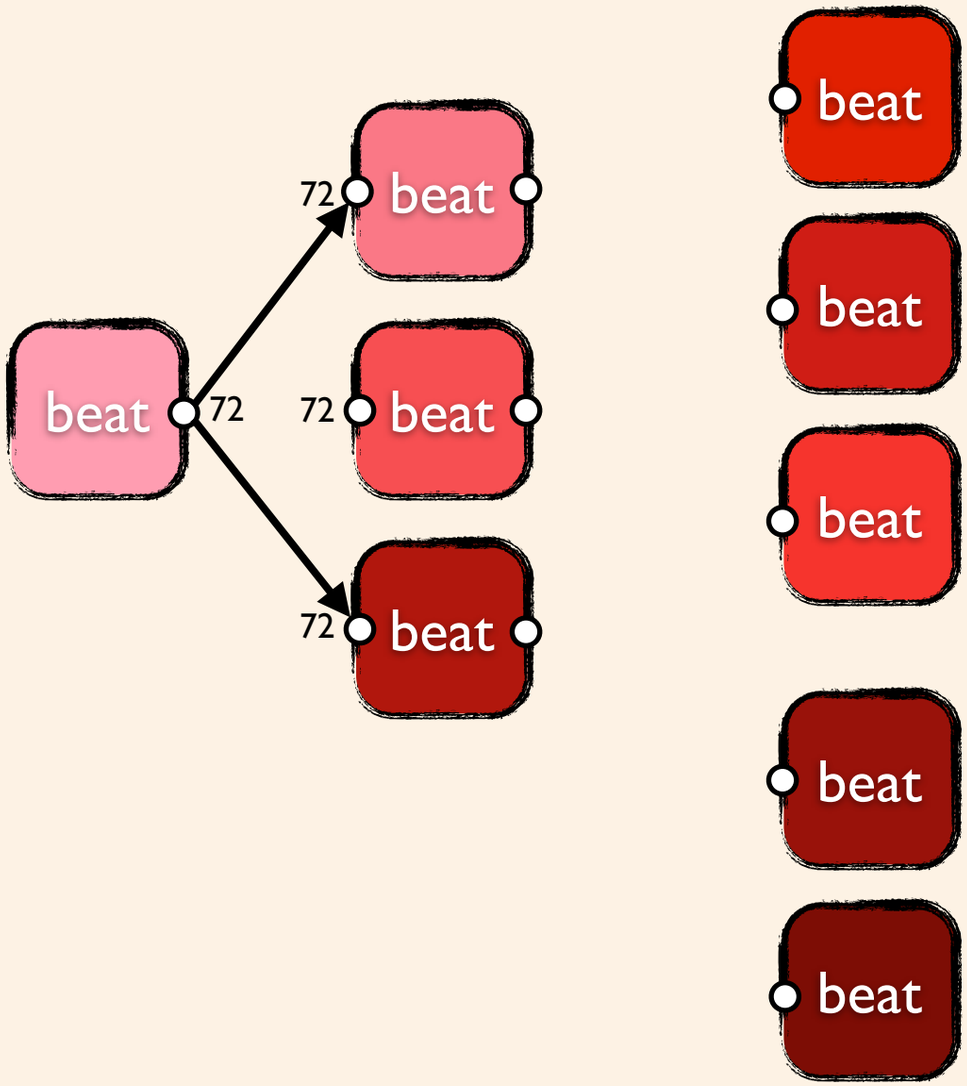
Leading voice



Beat store

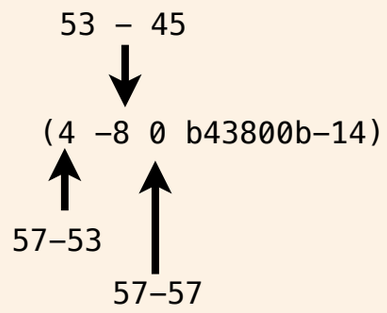
```
{:start-notes '(53 57 62 65)
 :destination-notes '(45 57 64 72)
 :events '((22000 53 500 4 96)
           (22000 57 1000 3 96)
           (22000 62 1500 2 96)
           (22000 65 1000 1 96)
           (22500 55 500 4 96)
           (23000 57 1000 4 96)
           (23000 57 1000 3 96)
           (23000 64 1000 1 96)
           (23500 61 500 2 96))
 :voice-leading '((4 -8 0 b43800b-14)
                  (9 -8 2 b43800b-14)
                  (12 -8 7 b43800b-14)
                  (5 0 2 b43800b-14)
                  (8 0 7 b43800b-14)
                  (3 2 7 b43800b-14))}
```

Leading voice

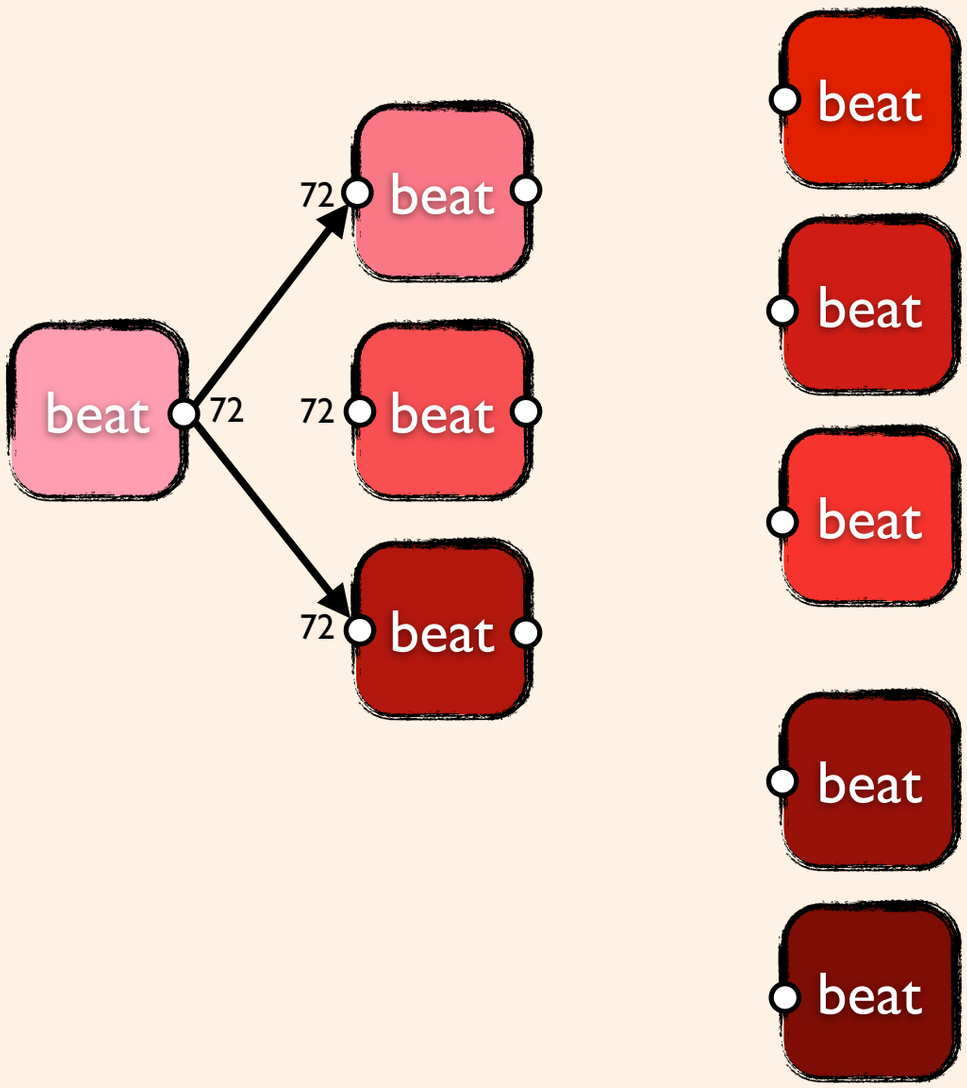


Beat store

```
{:start-notes '(53 57 62 65)
 :destination-notes '(45 57 64 72)
 :events '((22000 53 500 4 96)
           (22000 57 1000 3 96)
           (22000 62 1500 2 96)
           (22000 65 1000 1 96)
           (22500 55 500 4 96)
           (23000 57 1000 4 96)
           (23000 57 1000 3 96)
           (23000 64 1000 1 96)
           (23500 61 500 2 96))
 :voice-leading '((4 -8 0 b43800b-14)
                  (9 -8 2 b43800b-14)
                  (12 -8 7 b43800b-14)
                  (5 0 2 b43800b-14)
                  (8 0 7 b43800b-14)
                  (3 2 7 b43800b-14))}
```



Leading voice



Beat store

```
{:start-notes '(53 57 62 65)
 :destination-notes '(45 57 64 72)
 :events '((22000 53 500 4 96)
           (22000 57 1000 3 96)
           (22000 62 1500 2 96)
           (22000 65 1000 1 96)
           (22500 55 500 4 96)
           (23000 57 1000 4 96)
           (23000 57 1000 3 96)
           (23000 64 1000 1 96)
           (23500 61 500 2 96))
 :voice-leading '((4 -8 0 b43800b-14)
                  (9 -8 2 b43800b-14)
                  (12 -8 7 b43800b-14)
                  (5 0 2 b43800b-14)
                  (8 0 7 b43800b-14)
                  (3 2 7 b43800b-14))}
```

Lexicon

```
{"bach-45-57-64-72"
  {:beats ("b43800b-15"
           "b43600b-62"
           "b43300b-10"
           "b43100b-2")}}
```

```
(require '[musical-creativity.band :as band] :reload)
(band/play (improvise/compose))

;Recombinance (Based on Bach Choral no. 140)
(require '[musical-creativity.composers.recombinance :as recombinance] :reload)
(require '[data.bach :as bach] :reload)
(recombinance/create-database-from bach/chorale-140-data)
(musician/play (recombinance/compose) organ)
(musician/play (recombinance/compose) piano)
(musician/play (recombinance/compose) slow-piano)

(musician/play (recombinance/compose-original) piano)

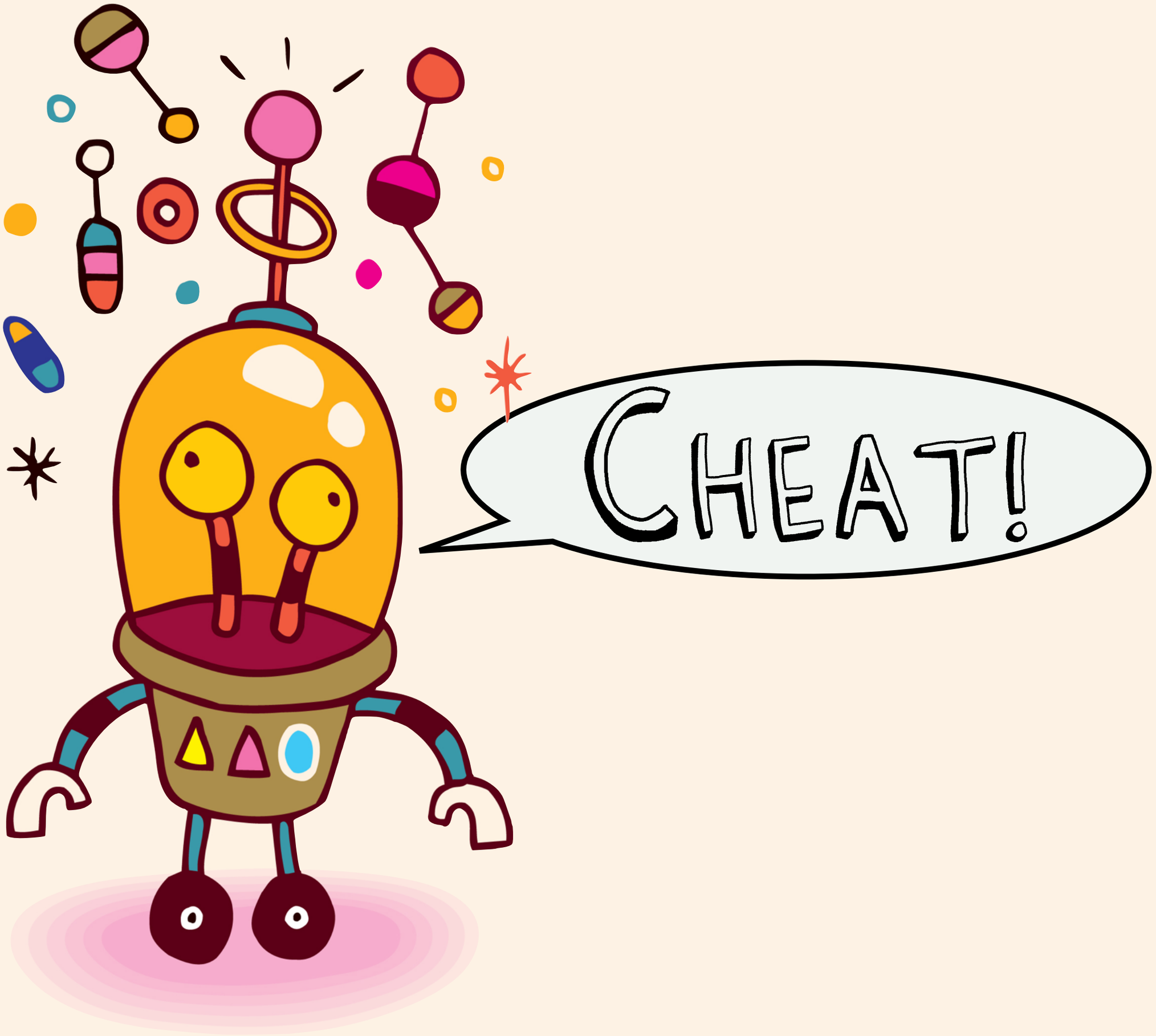
;Gradus (with each run Gradus learns how to better compose)
(require '[musical-creativity.composers.gradus :as gradus] :reload)
```

```
-=--:----F1  README.md      48% (52,13)   Git-master  (Clojure Hi nREPL Fill) --------------
   / __  /_  _____  _____/ /_____  ____  ___
  / / / / | / / _ \/ ___/ __/ __ \/ __ \/ _ \
 / /_/ /| |/ /  __/ /  / /_/ /_/ / / / /  __/
 \____/ |___/\___/_/   \__/\____/_/ /_/\___/

   Collaborative Programmable Music. v0.8


Hello Elise. Do you feel it? I do. Creativity is rushing through your veins today!


user>
-=--:**--F1  *nrepl*        Bot L17    (nREPL ElDoc) ------------------------------------
```


CHEAT!

```
(require '[musical-creativity.band :as band] :reload)
(band/play (improvise/compose))

;Recombinance (Based on Bach Choral no. 140)
(require '[musical-creativity.composers.recombinance :as recombinance] :reload)
(require '[data.bach :as bach] :reload)
(recombinance/create-database-from bach/chorale-140-data)
(musician/play (recombinance/compose) organ)
(musician/play (recombinance/compose) piano)
(musician/play (recombinance/compose) slow-piano)

(musician/play (recombinance/compose-original) piano)

;Gradus (with each run Gradus learns how to better compose)
(require '[musical-creativity.composers.gradus :as gradus] :reload)
```

```
-=--:----F1  README.md      48% (56,44)   Git-master  (Clojure nREPL Fill) ----------------
user>
```

```
-=--:**--F1  *nrepl*        All L1     (nREPL ElDoc) ------------------------------------
```

INFLUENCE

"Creativity develops within enfolding and influencing context, and not in isolation"

David Cope

```
;Improvise
;(very experimental)
(require '[musical-creativity.composers.improvise :as improvise] :reload)
(musician/play (improvise/compose) piano)

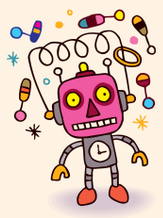
(require '[musical-creativity.band :as band] :reload)
(band/play (improvise/compose))

;Recombinance (Based on Bach Choral no. 140)
(require '[musical-creativity.composers.recombinance :as recombinance] :rel\
oad)
(require '[data.bach :as bach] :reload)
(recombinance/create-database-from bach/chorale-140-data)
(musician/play (recombinance/compose) organ)
(musician/play (recombinance/compose) piano)
-UU-:----F1  README.md       44% (49,31)   Git:master  (Clojure Paredit nREPL
user>


-UUU:**--F1  *nrepl*         All L1      (nREPL Paredit ElDoc) ---------------
```

LEARNING

INFERENCE



"Creativity requires learning and knowledge in order to produce useful rather than arbitrary results"

David Cope

```
(recombinance/create-database-from bach/chorale-140-data)
(musician/play (recombinance/compose) organ)
(musician/play (recombinance/compose) piano)
(musician/play (recombinance/compose) slow-piano)

(musician/play (recombinance/compose-original) piano)

;Gradus (with each run Gradus learns how to better compose)
(require '[musical-creativity.composers.gradus :as gradus] :reload)

(musician/play (gradus/compose) piano)

(musician/play (gradus/compose-canon) piano)

(musician/play (gradus/compose-canon data.gradus/cantus-firmus-1) piano)
(musician/play (gradus/compose-canon data.gradus/cantus-firmus-2) piano)
(musician/play (gradus/compose-canon data.gradus/cantus-firmus-7) piano)
-UU-:----F1  README.md      56% (62,0)    Git:master  (Clojure Paredit nREPL ElDoc Fill) -
  / / / / | / / _ \/ ___/ __/ __ \/ __ \/ _ \
 / /_/ /| |/ /  __/ /  / /_/ /_/ / / / /  __/
 \____/ |___/\___/_/   \__/\____/_/ /_/\___/

   Collaborative Programmable Music. v0.8


Hello Josephwilk, may this be the start of a beautiful music hacking session...


user>
-UUU:**--F1  *nrepl*        Bot L17    (nREPL Paredit ElDoc) ----------------------------
```

```
## Analogies and Inference

```clojure
(use 'overtone.live)
(use 'overtone.synth.sampled-piano)
(use 'overtone.music.pitch)
(use 'musical-creativity.experiments.analogy)

(reset! fact-store '((g-b-d*c-e-g are dominant*tonic)
                     (dominant*tonic are resolutions)
                     (f-a-c*c-e-g are subdominant*tonic)
                     (subdominant*tonic are resolutions)))

(analogy 'g-b-d*c-e-g)

(map #(do (sampled-piano :note (overtone.music.pitch/note %)) (Thread/sleep 400)) [\
:g4 :b4 :d4])
(map #(do (sampled-piano :note (overtone.music.pitch/note %)) (Thread/sleep 400)) [\
:c4 :e4 :g4])
-UU-:----F1  Readme.md       Top (9,13)    Git:master  (Clojure Hi Paredit nREPL ElDo

   Collaborative Programmable Music. v0.8

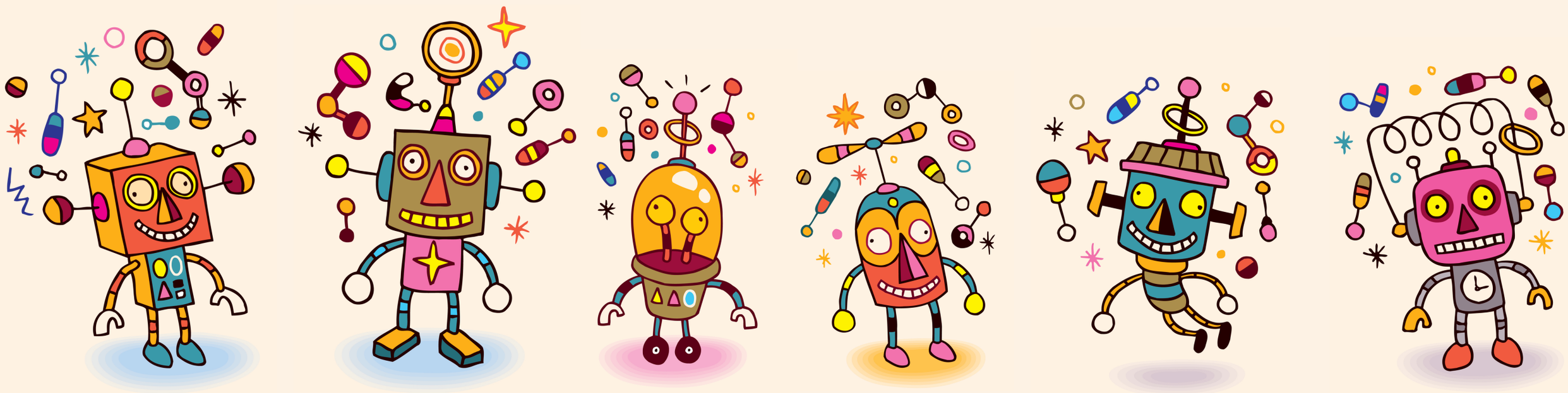

Hello Josephwilk, may this be the start of a beautiful music hacking session...


user>
```
```

DAVID COPE

EMILY HOWELL

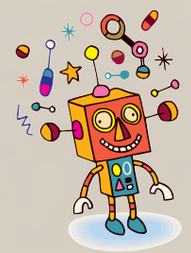
Creative Machines

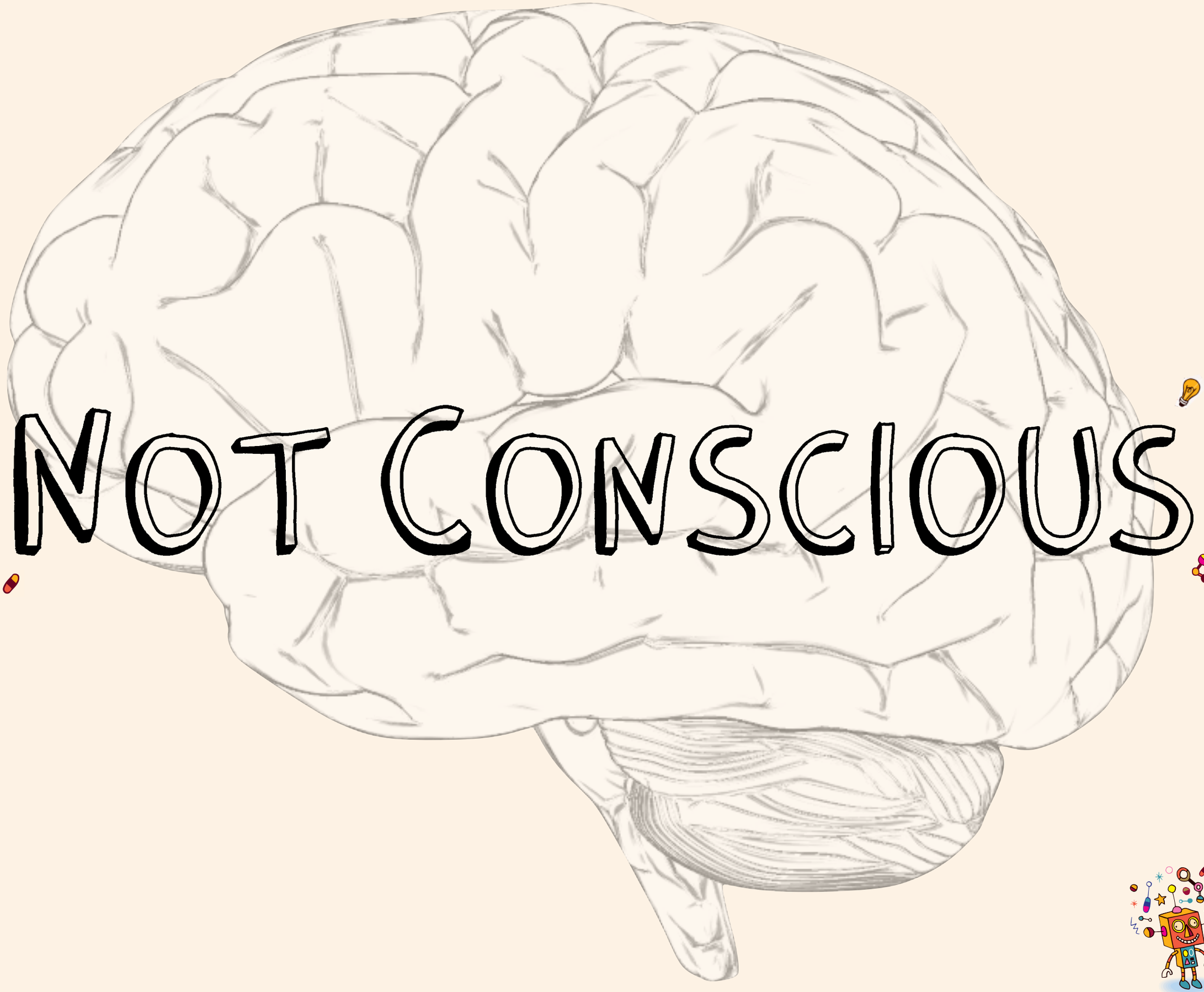


Joseph Wilk

“it can do whatever we know how to order it to perform”

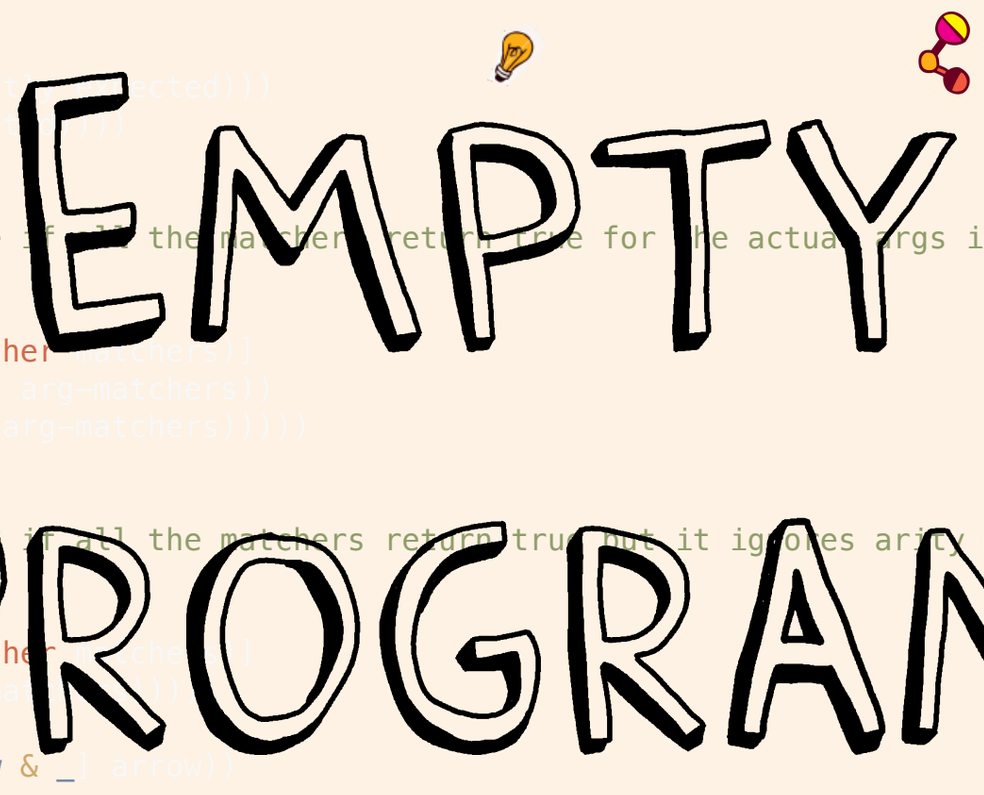
“it has no pretensions whatever to originate anything”

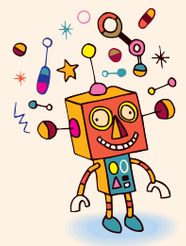




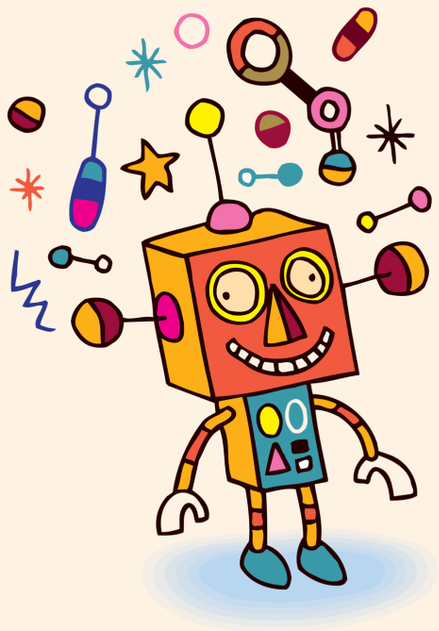
Not Conscious

```clojure
(ns ^{:doc "The semi-sweet representation of provided forms."}
 midje.parsing.3-from-lexical-maps.from-fake-maps
 (:use midje.clojure.core
       midje.checking.core
       [midje.checkers :only [exactly]]
       [midje.checking.checkers.defining :only [checker?]]
       [midje.parsing.arrow-symbols])
 (:require [midje.util.exceptions :as exceptions]))

(defn- mkfn arg-matcher
 "Based on an expected value, generates a function that returns
 true if the actual value matches it."
 [expected]
 (if (and (extended-fn? expected)
          (not (checker? expected)))
   (fn [actual] (extended-= actual (exactly expected)))
   (fn [actual] (extended-= actual expected))))

(defn mkfn arg-matchers-with-arity
 "Generates a function that returns true if all the matchers return true for the actual args its passed."
 [matchers]
 (fn [actual-args]
   (let [arg-matchers (map mkfn:arg-matcher matchers)]
     (and (= (count actual-args) (count arg-matchers))
          (extended-list-= actual-args arg-matchers)))))

(defn mkfn arg-matchers-without-arity
 "Generates a function that returns true if all the matchers return true but it ignores arity matching."
 [matchers]
 (fn [actual-args]
   (let [arg-matchers (map mkfn:arg-matcher matchers)]
     (extended-list-= actual-args arg-matchers))))

(defmulti mkfn:result-supplier (fn [arrow & _] arrow))

(defmethod mkfn:result-supplier => [_arrow_ result] (constantly result))

(defmethod mkfn:result-supplier =streams=> [_arrow_ result-stream]
 (let [the-stream (atom result-stream)]
   (fn []
     (when (empty? @the-stream)
       (throw (exceptions/user-error "Your =stream=> ran out of values.")))
     (let [current-result (first @the-stream)]
       (swap! the-stream rest)
       current-result))))

(defmethod mkfn:result-supplier =throws=> [_arrow_ throwable]
 (fn []
   (when-not (instance? Throwable throwable)
```

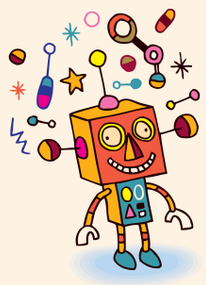


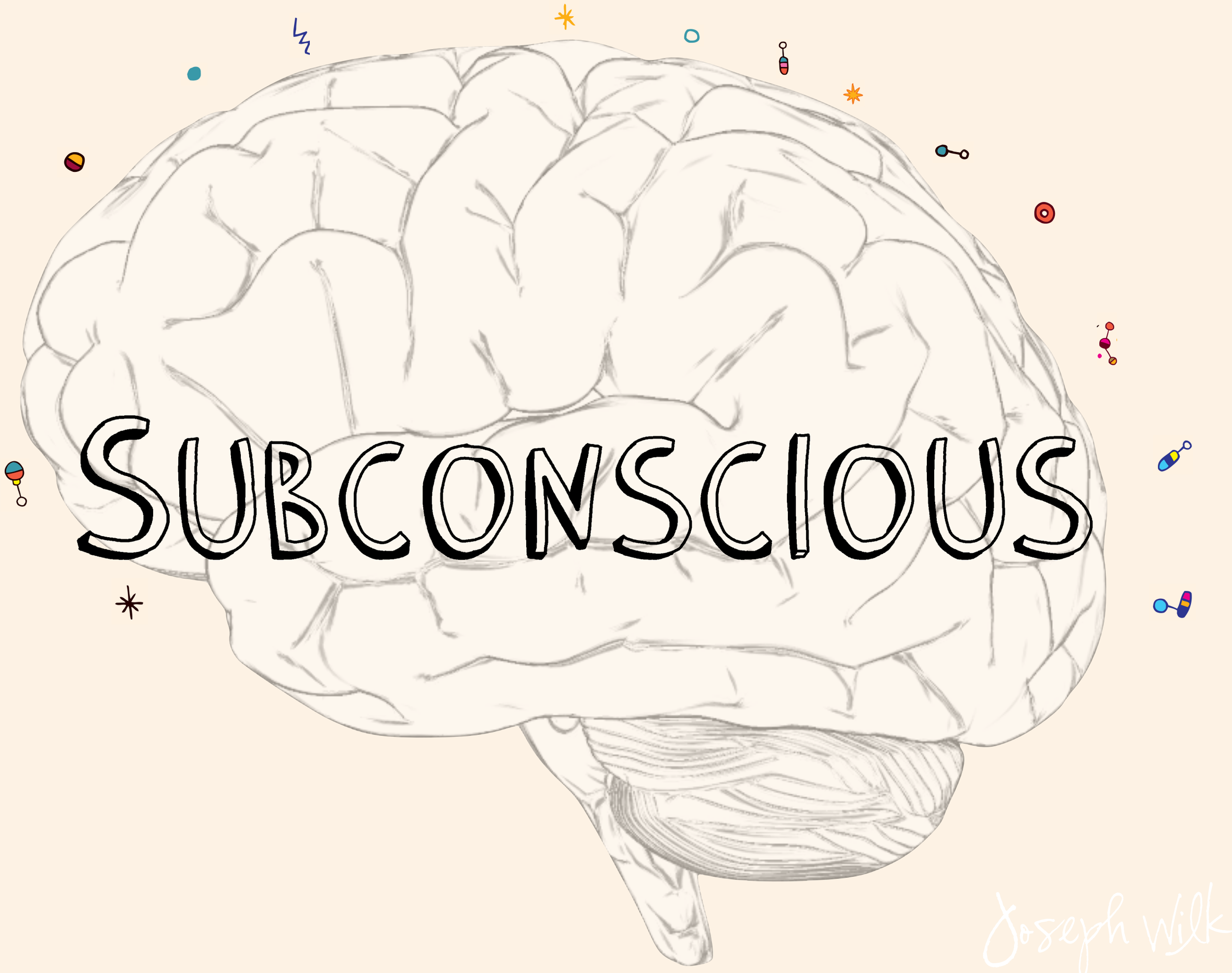


NON HUMAN



CREATIVITY




SUBCONSCIOUS
Joseph Wilk