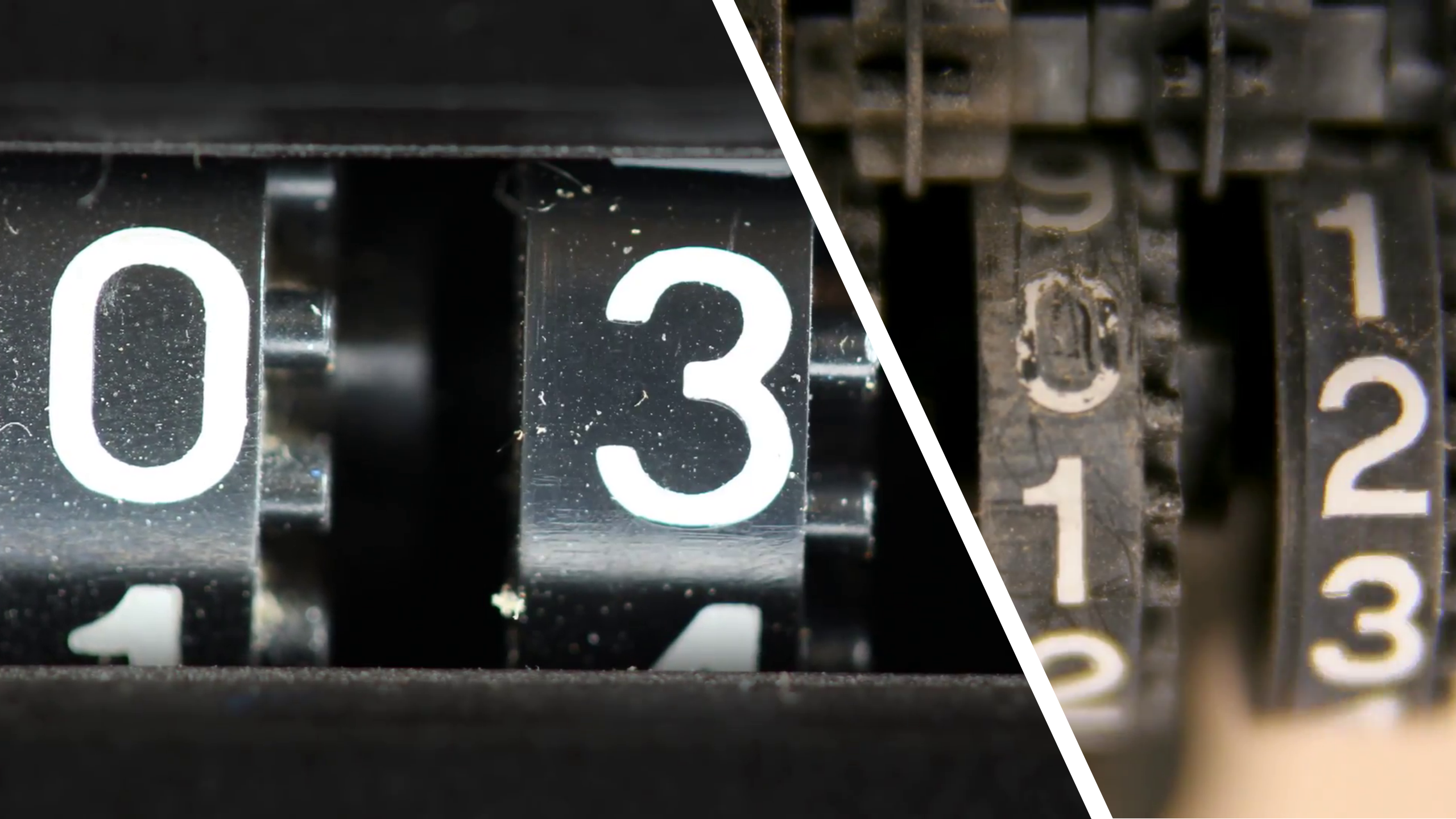
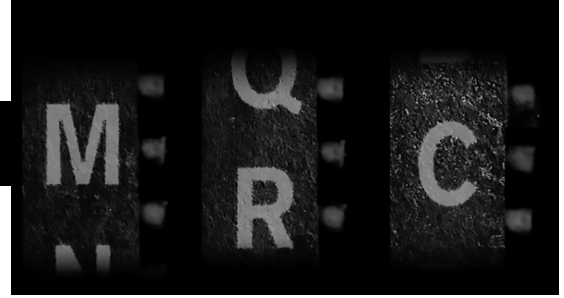
algorithm counter

Kio Griffith


0 3
0 1 2 3

algorithm counter

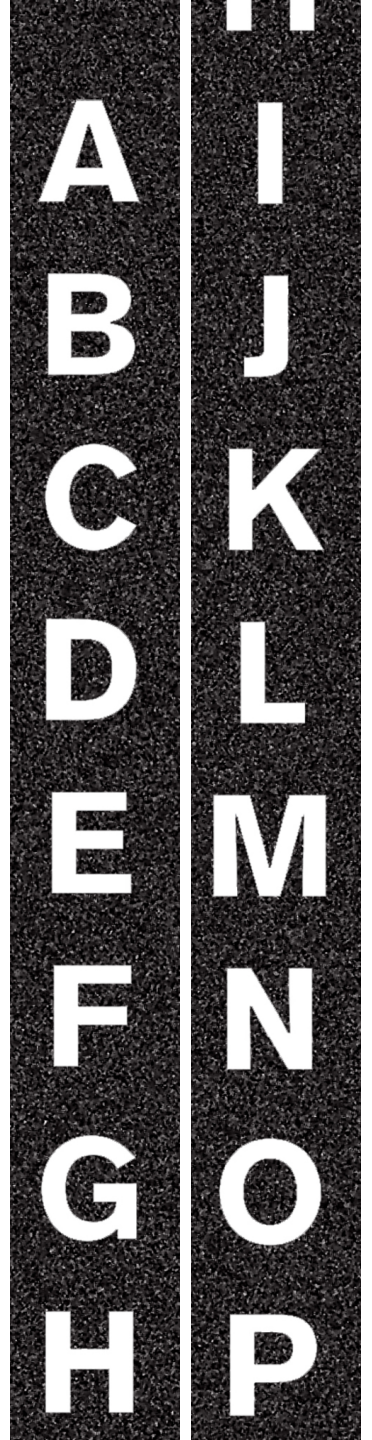
concept

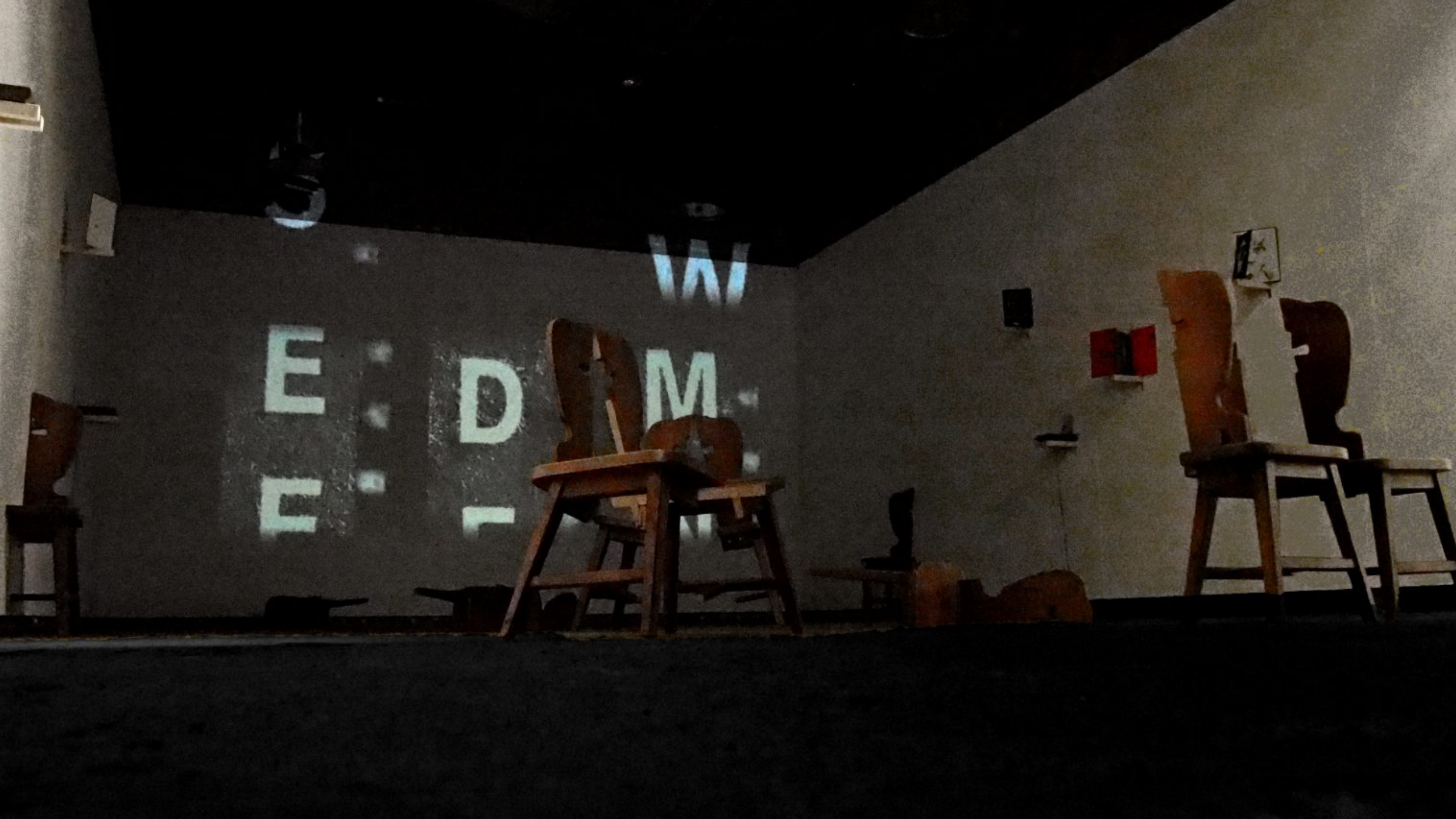


“algorithm counter” is a chance operation machine formulating various matrices of words and language associations randomly arranged and contextualized by ascending and descending letters in flux.

“algorithm counter” is a language calculator, a stream of consciousness timer, and an interminable messaging billboard. the projection mapping could be organized in digits—the modular single-wheeled digits can operate independently or work as components in groupings.

video link : https://vimeo.com/250318371




E D M


L X
M Y Y T W D R V


S V C
T W D
S V C J N X S L A A
T W D R V O V T M Y Y
C L X X S V C U B C
R V M Y Y T W D U C D

As a chance operation machine forming various matrices of word and language associations arranged and contextualized by ascending and descending directions and speeds, “Algorithm Counter” is a language calculator, a stream of consciousness timer, and interminable messaging billboard. the visual projection of an algorithm counter that could be mapped out (in digits) over an external or internal wall surface, whether it being an interior of a gallery space or a facade of a building.

strategies // actions

-motion : : forward / reverse / speed

-language conversion

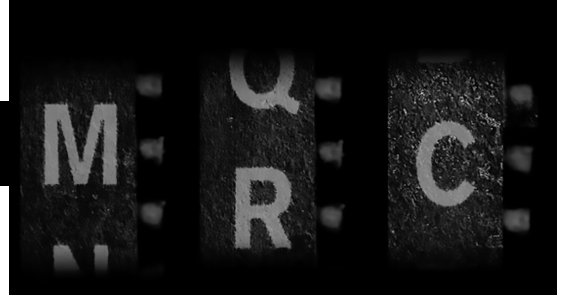
-trigger / pitch / sound / notes

- read acronyms

- generate digits

algorithm counter

list of actions



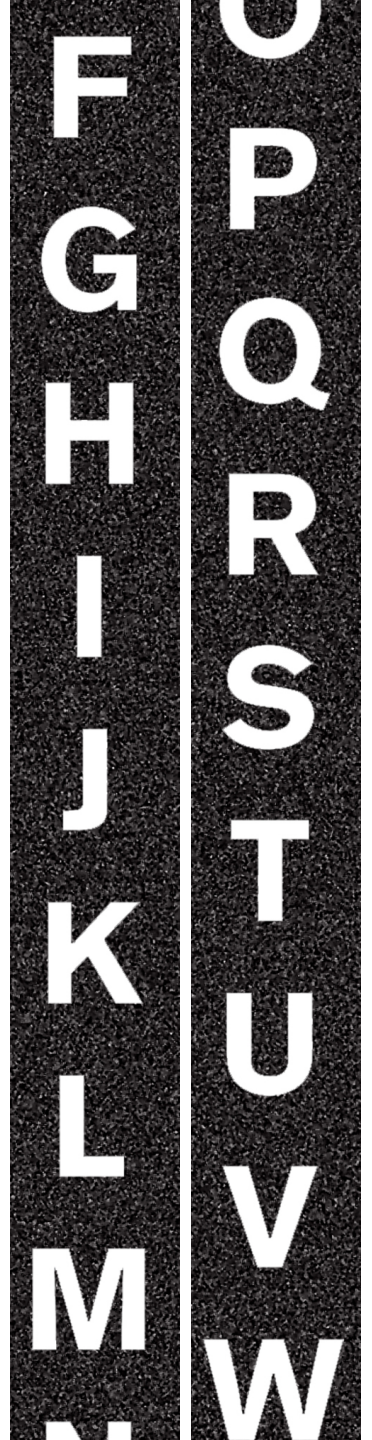
-motion : : forward / reverse / speed

-generate digits

-trigger / pitch / sound / notes

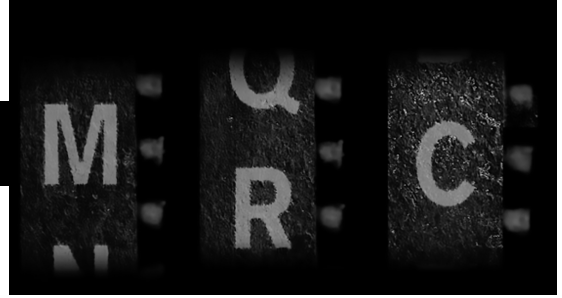
- read acronyms

-language conversion



algorithm counter

list of actions



-motion : : forward / reverse / speed

-generate digits

(advanced)

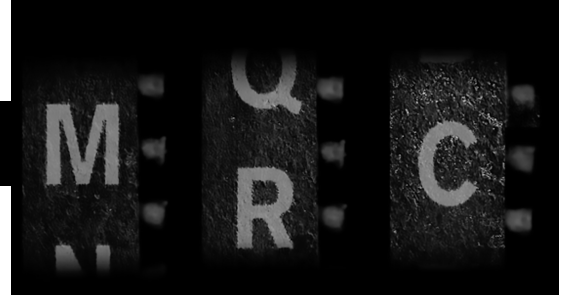
-trigger / pitch / sound / notes

- read acronyms

-language conversion

algorithm counter

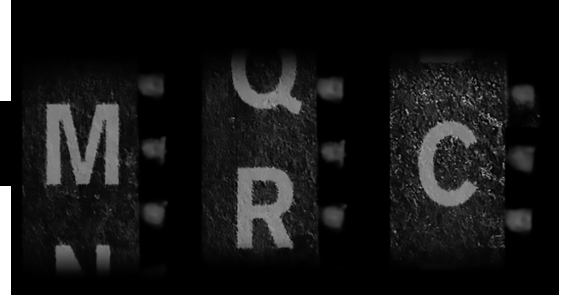
work in progress




```
/*** Spinning Wheel GL ***/

/*

Change Log:

2014_01_11 2:00pm - GL - initial version
2014_01_11 5:20pm - AGF - changed code to stable demo; added parameters

*/


/*** PARAMETERS THAT CAN BE EXPERIMENTED WITH ***/

int WINDOW_WIDTH = 1000;
int WINDOW_HEIGHT = 100;

//MAX_SPEED is The maximum number of pixels that the images will move duri
int MAX_SPEED = 100;

//SPEED_MULTIPLIER is the the number of pixels the image should move per
//the number of mouse movements. For example, if SPEED_MULTIPLIER = 5,
//the if the user moves 20 pixels, then the image will move 20*5 = 100 pix
//(or MAX_PIXELS if the result is greater than MAX_PIXELS).
int SPEED_MULTIPLIER = 3;

//DAMPEN_SPEED is The rate at which the speed is slowed down each frame.
//1.0 = never slow down.
//0.0 = slow down instantly.
//0.95 seems like a good default value
float DAMPEN_SPEED = 0.95;

//MIN_MOUSE_MOVEMENT is the minumum amount of movement needed to start a cl
//movement of the image. This is to avoid weird changes in direction when
//barely touched. 10 seems a good default value.
```

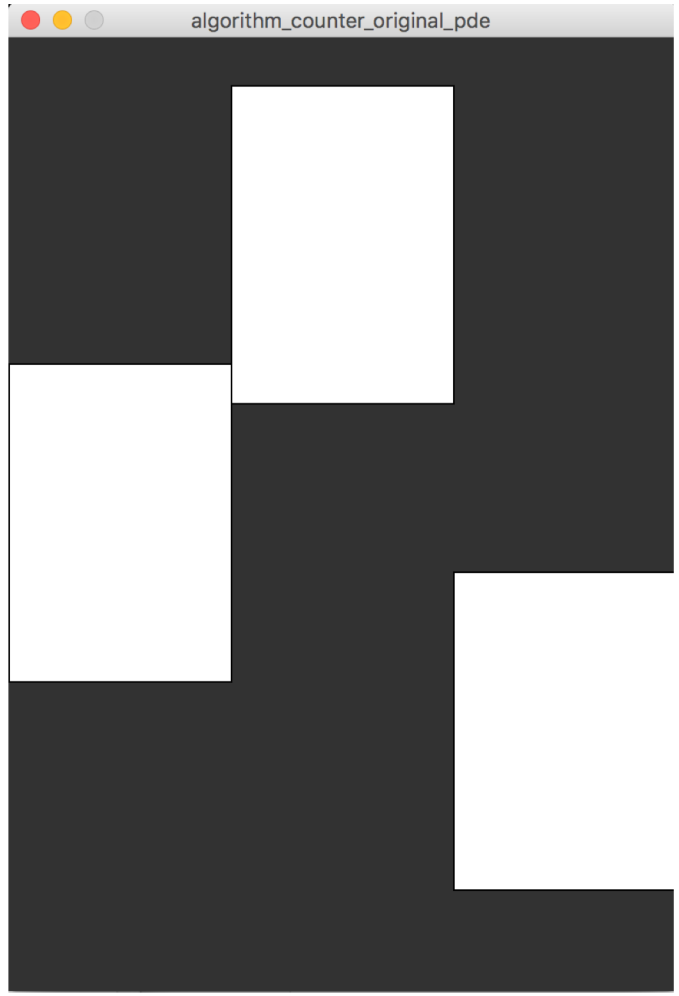
algorithm counter

work in progress



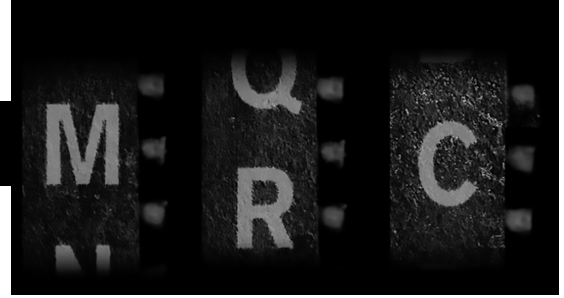
algorithm_counter_original_pde

```
  } else if (y < -SCREEN_MARGIN) { //wrap around to right side of screen
    y += (height + (SCREEN_MARGIN));
  }

  //draw rectangle
//  rect(x, y, RECT_WIDTH, (height/3));
//}
  rect(x-140, y*1.4, RECT_WIDTH,(height/3));
  rectMode(CENTER);
   rect(x, y*0.6, RECT_WIDTH,(height/3));
  rectMode(CENTER);
   rect(x+140, y*2, RECT_WIDTH,(height/3));
  rectMode(CENTER);
}

void mouseMoved(MouseEvent me) {

//  gDiff = (mouseX-gPrevMouseX);
  gDiff = (mouseY-gPrevMouseY);

  if (abs(gDiff) > MIN_MOUSE_MOVEMENT) { //ignore tiny mouse movements

    //determine direction of mouse movement
//    if (gDiff > 0) {
//      gDir = 1;
//    } else if (gDiff < 0) {
//      gDir = -1;
//    }
     if (gDiff > 0) {
      gDir = 2;
    } else if (gDiff < 0) {
      gDir = -2;
    }

    gSpeed = abs(gDiff) * SPEED_MULTIPLIER;
```



algorithm counter

work in progress

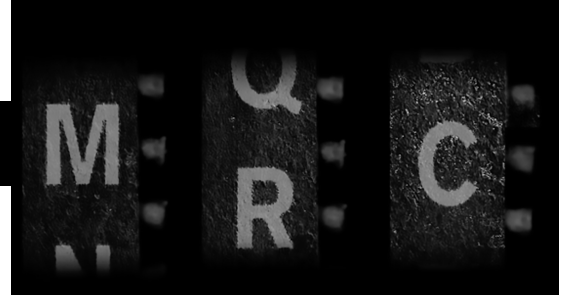


algorithm_counter_original_pde_pde

```
  gPrevMouseY = mouseY;
  gSpeed = 15.0;
  x = 0;
  y1 = 0;
  y2 = bg.height;
}

void draw()
{
  background(50);

  //dampen the current speed


  y1 += gSpeed;
  image(bg, 0, y1, bg.width, bg.height);
  if (y1>=height) {
    y1 = (y1 - 2*bg.height);
  }

  y2 += gSpeed;
  image(bg, 0, y2, bg.width, bg.height);
  if (y2>=height) {
    y2 = (y2 - 2*bg.height);
  }

  gSpeed *= DAMPEN_SPEED;
}

void mouseMoved(MouseEvent me) {
  gSpeed=30;
}
```

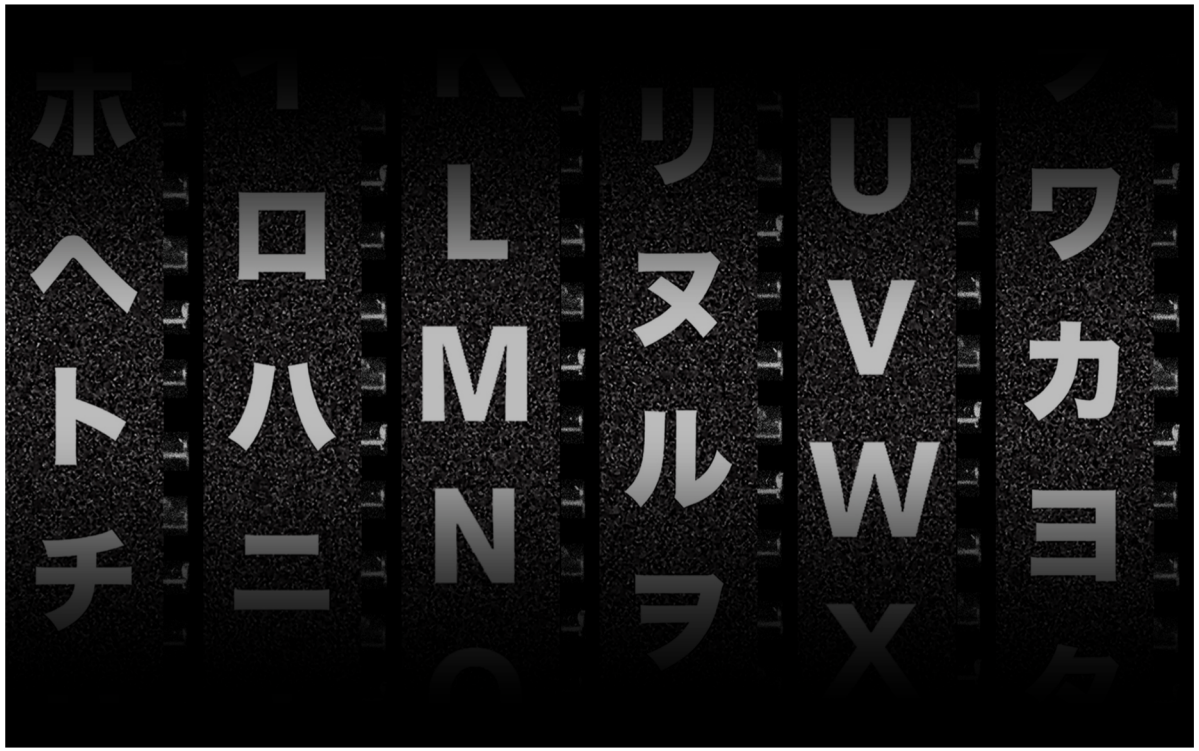
algorithm counter

work in progress



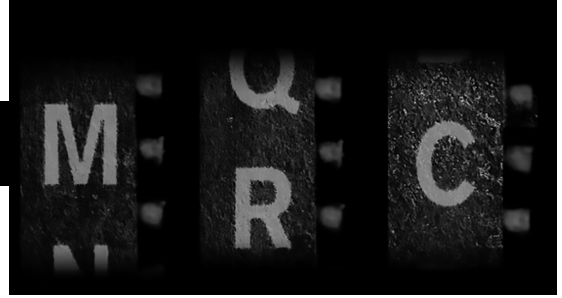
algorithm_counter_version3_fast_pde_pde

```
//dampen the current speed
gSpeed5 *= DAMPEN_SPEED/1;

//  ----------------------

  // determine position
y61 += gSpeed6;

//continuity of image running. controlling anchor position
if (y61>=height) {
  y61 = (y61 - 2*bg.height);
}
if ((y61+bg.height)<0) {
  y61 = y61 + bg.height*2 ;
}
//image drawing
image(bg, 1400, y61, bg.width, bg.height);

y62 += gSpeed6;

if (y62>=height) {
  y62 = (y62 - 2*bg.height);
}

if ((y62+bg2.height)<0) {
  y62 = y62 + bg.height*2 ;
}

image(bg, 1400, y62, bg.width, bg.height);

//changing speed
//dampen the current speed
gSpeed6 *= DAMPEN_SPEED/1;
```



algorithm counter

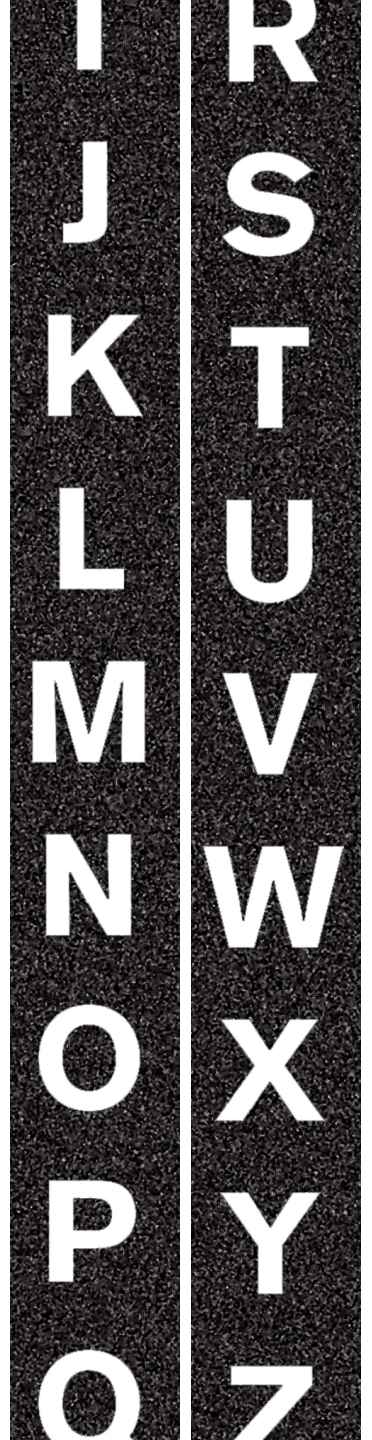
references



Oblique Strategies - Brian Eno

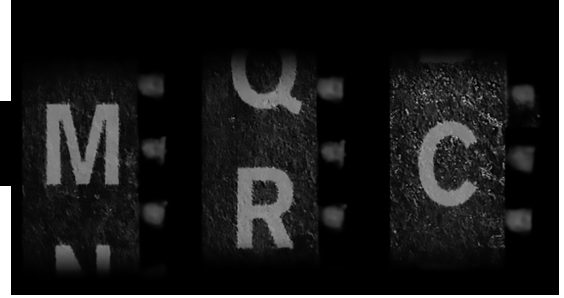
Chance Operation - Fluxus

Information Theory - Claude E. Shannon



algorithm counter

references

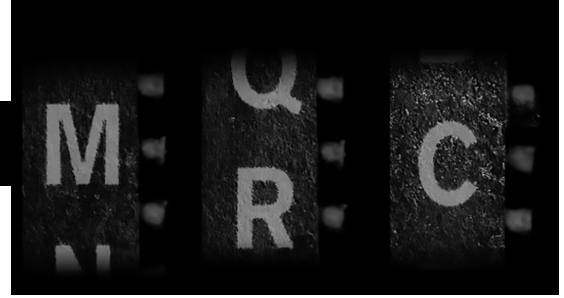


Oblique Strategies - Brian Eno

(subtitled Over One Hundred Worthwhile Dilemmas) is a card-based method for promoting creativity jointly created by Brian Eno and Peter Schmidt, first published in 1975. Physically, it takes the form of a deck 7-by-9-centimetre (2.8 in × 3.5 in) printed cards in a black box. Each card offers a challenging constraint intended to help artists (particularly musicians) break creative blocks by encouraging lateral thinking.

algorithm counter

references

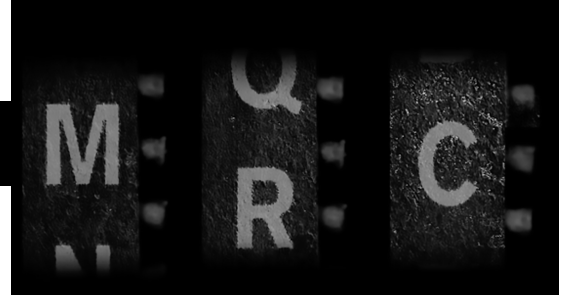


Chance Operation – Fluxus

An Anthology of Chance Operations (An Anthology) was an artist's bookpublication from the early 1960s of experimental neodada art and music composition that used John Cage inspired indeterminacy. It was edited by La Monte Young and DIY co-published in 1963 by Young and Jackson Mac Low in New York City. Its full title is: An Anthology of chance operations concept art anti-art indeterminacy improvisation meaningless work natural disasters plans of action stories diagrams Music poetry essays dance constructions mathematics compositions.

algorithm counter

references



Information Theory - Claude E. Shannon

A key measure in information theory is "entropy". Entropy quantifies the amount of uncertainty involved in the value of a random variable or the outcome of a random process. For example, identifying the outcome of a fair coin flip (with two equally likely outcomes) provides less information (lower entropy) than specifying the outcome from a roll of a die (with six equally likely outcomes). The field is at the intersection of mathematics, statistics, computer science, physics, neurobiology, information engineering, and electrical engineering. The theory has also found applications in other areas, including statistical inference, natural language processing, cryptography, neurobiology, human vision, the evolution and function of molecular codes (bioinformatics), model selection in statistics, thermal physics, quantum computing, linguistics, plagiarism detection, pattern recognition, and anomaly detection. Important sub-fields of information theory include source coding, channel coding, algorithmic complexity theory, algorithmic information theory, information-theoretic security, and measures of information.