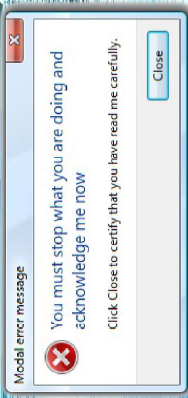
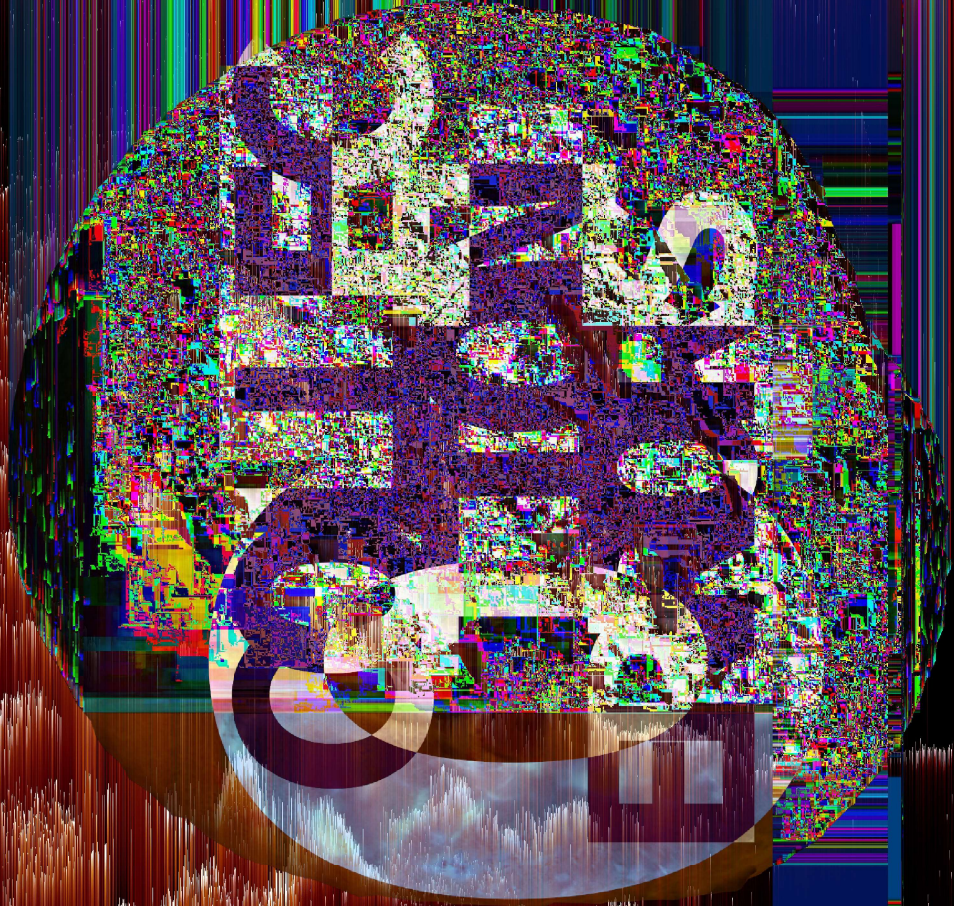


# COOKIE COLLECTIVE

NUMBER 2

MAY 2019

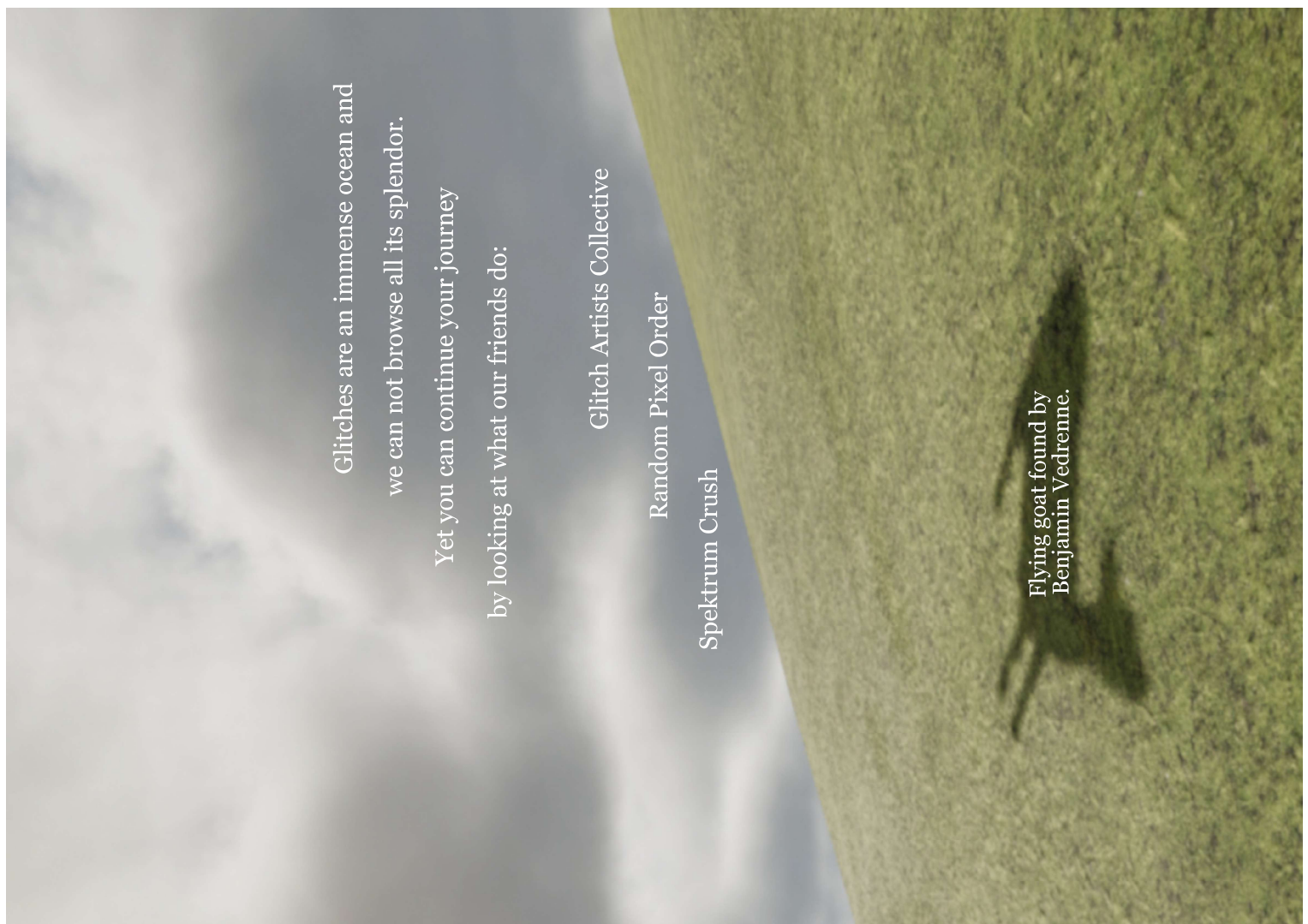


Errors are everywhere.  
Cheese is an oversight.  
Alcohol is a mistake.  
Life on earth was not planned.

Glitches are an immense ocean and  
we can not browse all its splendor.  
Yet you can continue your journey  
by looking at what our friends do:

Glitch Artists Collective  
Random Pixel Order  
Spektrum Crush

Flying goat found by  
Benjamin Vedrenne.



This fanzine was made by

Léon Denise

Florine Fouquart

Jonathan Giroux

Thomas Hooper

Ktalyze

Sébastien Labrunie

Élie Michel

Marie Mollins

Kaspar Ravel

Clara Rigaud

Sonia Saroya

Benjamin Vedrenne



Cover by Sébastien Labrunie and Jonathan Giroux.

Glitches and bugs are inescapable actors of the digital activity. Although avoided by most programmers, unexpected results may be a delight to the artist's eyes. To the point that video compression artefacts are reproduced through sophisticated filters.

Inaccurate raymarching algorithms create colorful, dynamique sceneries. Error messages from compilers sound like post-punk album titles. Unintended results lead to unseen connections.

This fanzine puts the spotlight on creative errors.

**Jonathan Giroux & Léon Denise**

Why the Blue Screen Of Death is blue?



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

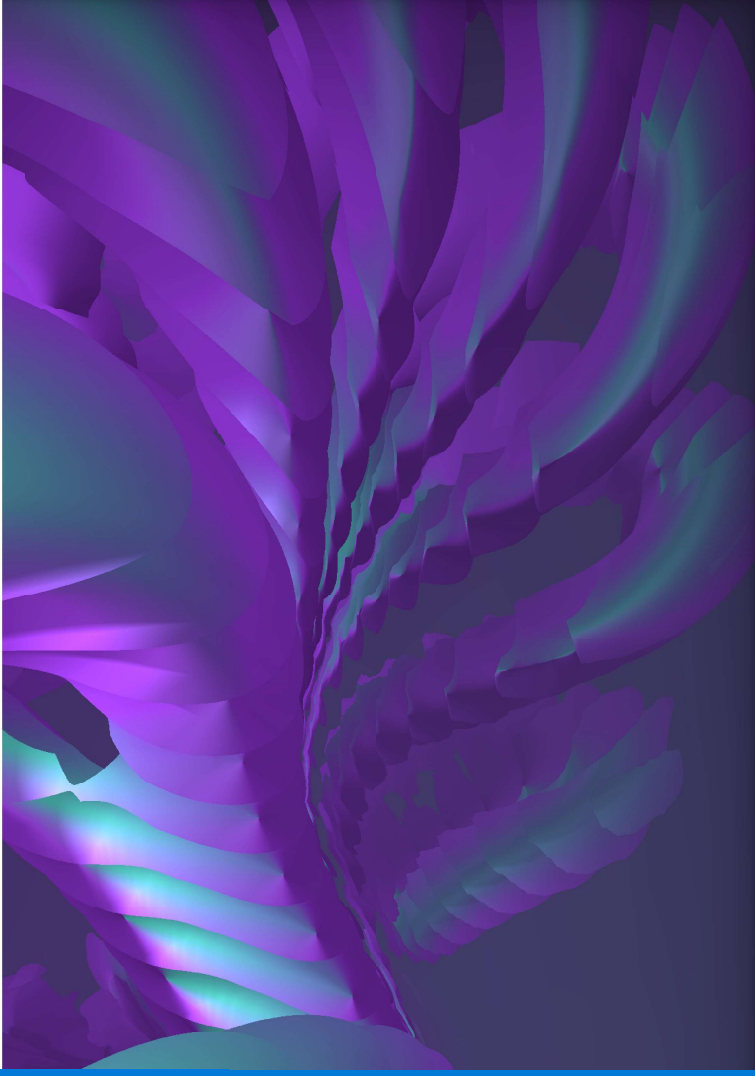
If you call a support person, give them this info:

Stop code: CRITICAL\_PROCESS\_DIED

[...] The Windows 10 BSOD is highly user friendly, we can see a small emoji which is not very happy and the interface that mutedly shows us that "Your PC ran into a problem and needs to restart. We're just collecting some error info, and then, we'll restart for you". A little percentage informs us about the data recovery progression, and if we want to gather more informations about what is happening with our computer, we can flash the QR CODE.

Everything seems to be fine, aside from a little hint down the screen which may upset us regarding to the nature of the breakdown:

« If you call a support person, give them this info:  
CRITICAL\_PROCESS\_DIED »



A wonderful aspect of ray marching is the ease of trying unusual concepts; what happens if time only subjectively moves forward when we move forward?

A ray marcher takes big jumps through a scene, and smaller and smaller steps as it approaches the geometry surface. If time accelerates by a fixed amount for each step, small or large, strange things happen...

As you near the surface you may find it gets swept away, or you could jump right into the middle of an object. This is highly dependent on how the geometry moves over time, static objects will behave normally, fast moving objects become ever more distorted.

**Thomas Hooper**  
[shadertoy.com/view/Wtfgz7](http://shadertoy.com/view/Wtfgz7)

```
***STOP: 0x000000D1 (0x00000000, 0xF73120AE, 0xC0000008, 0xC0000000)
```

A problem has been detected and Windows has been shut down to prevent damage to your computer.

```
DRIVER_IRQL_NOT_LESS_OR_EQUAL
```

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

```
*** ABCD.SYS - Address F73120AE base at C0000000, DateStamp 36B072A3
```

```
Kernel1 Debugger Using: COM2 (Port 0x2F8, Baud Rate 19200)  
Physical memory dump complete. Contact your system administrator or  
technical support group.
```

By now, if we compare the first BSOD to that of WINDOWS 10, we find ourself facing a mix of alphanumeric characters and non-alphabetical ones, I would even say cabalistic ones, without any reason of crash nor explanation, which is pretty frightening.

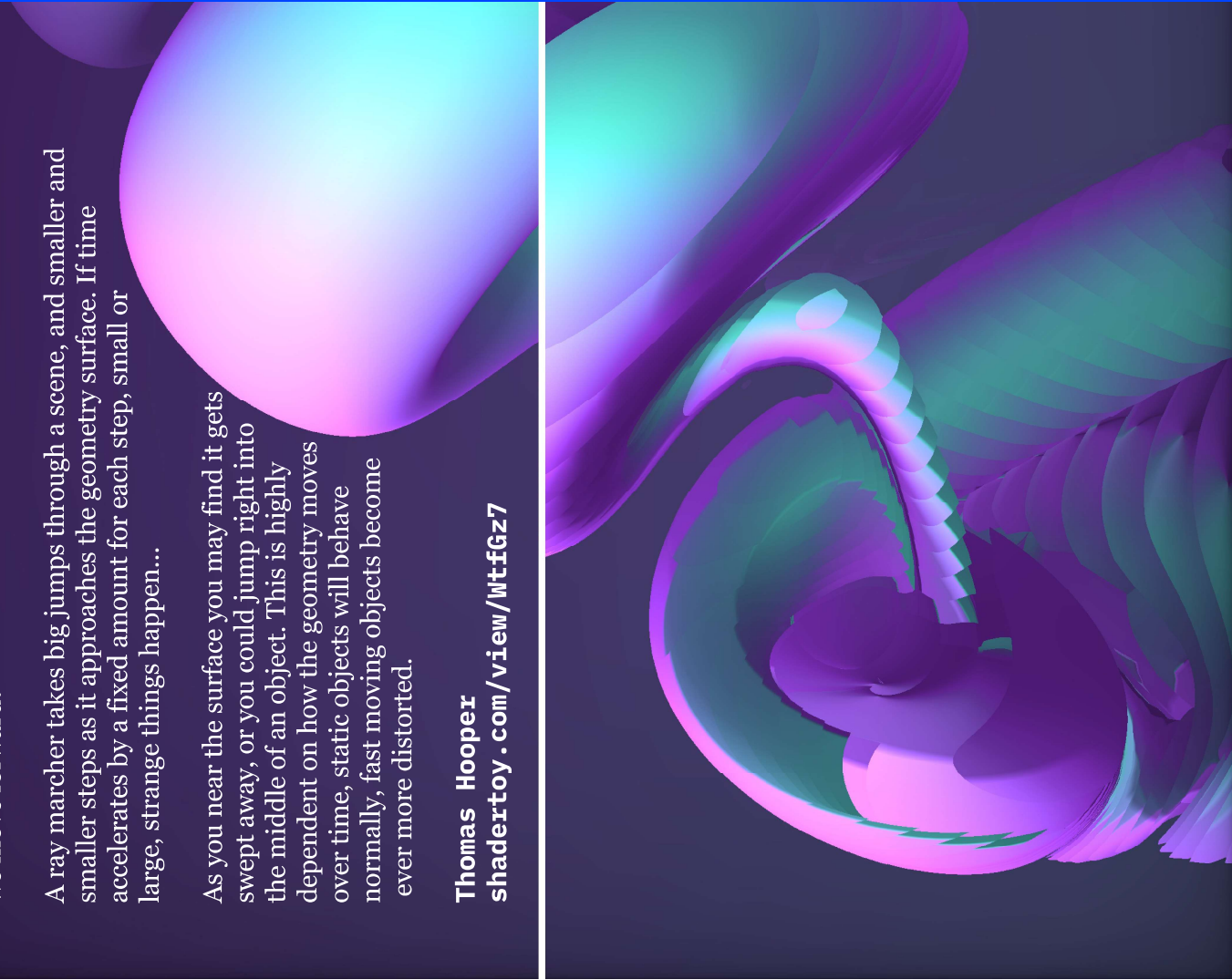
Thus, we have gone from a highly troublesome blue code we do not understand - the raw error displayed by the WINDOWS 1.0 BSOD -, to something extremely smooth, ergonomic, that clearly points out not to worry about the matter we are confronted with, the latter simply being "repackaged" from the first BSOD to the current one.

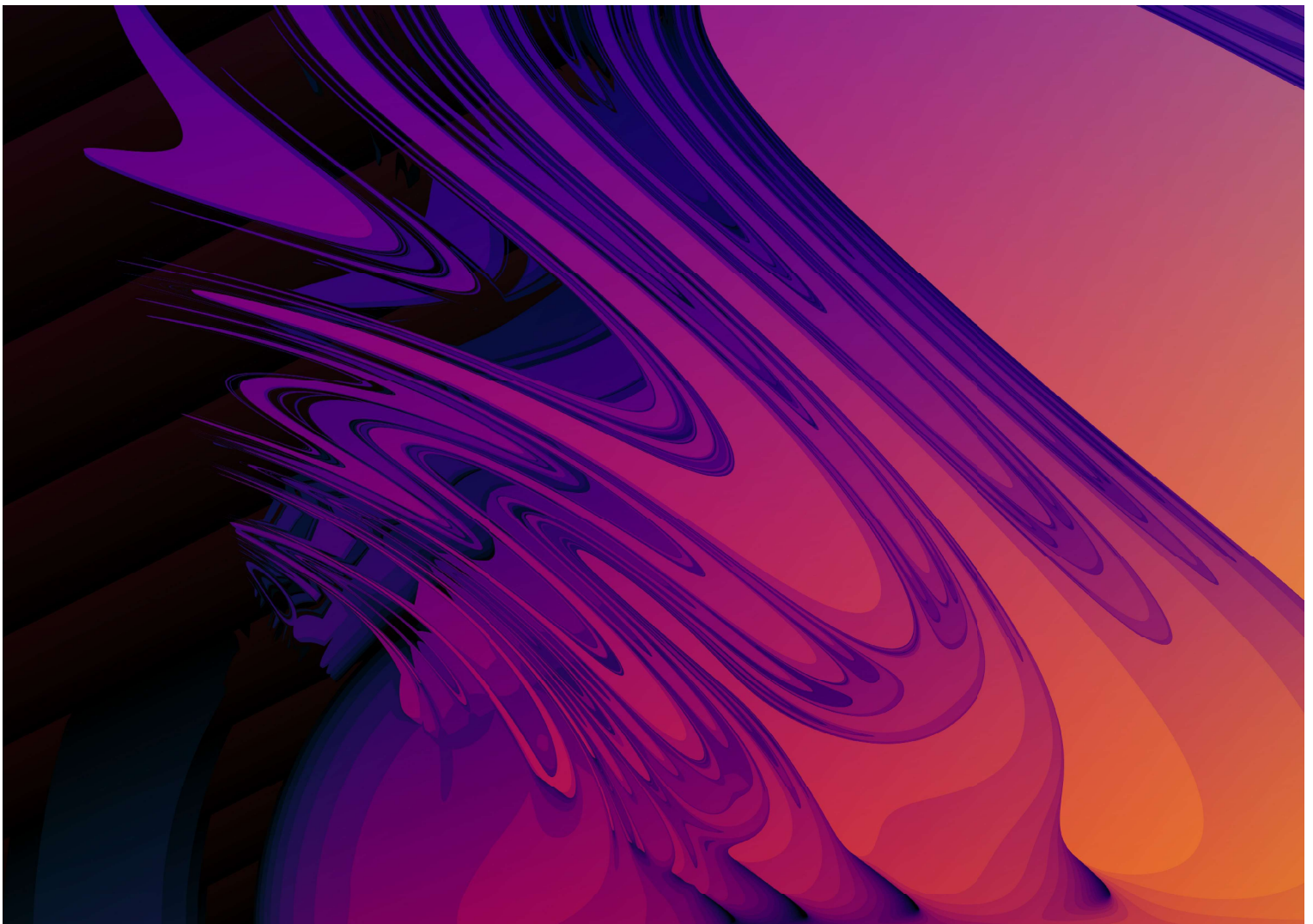
Furthermore, the Blue Screen of Death name was assigned by the very first users confronted to the crash, that is not a formal designation.

In a few decades, blue hues that are yet colorimetrically pretty close, distribute two contrasted messages. [...]

**Marie Mollins**  
[mariemollins.com](http://mariemollins.com)

**Extracts from the talk: neutral blue to: corrupted blue**





```

// Saturated Mounts
float mountsTime = 0.1;
float shapeTime = 2.7;
float colorTime = 216.84;
float sphere(vec3 p, float r) { return length(p) - r; }
float merge(float a, float b) { return min(a, b); }
float mounts(vec3 p) {
    p.x += iTime * .1;
    float h = sin(p.x) * sin(p.z * 10.) * 1.5 + 0.3
    * sin(mountsTime * 2.3);
    return p.y - h; }
float scene(vec3 p) {
    float s = sphere(p, 1.5);
    s = merge(s, mounts(p));
    return s; }
mat2 rot2(float theta) {
    float c = cos(theta), s = sin(theta);
    return mat2(c, s, -s, c); }
vec3 march(vec2 uv) {
    vec3 p = vec3(0., 1., -5.);
    vec3 d = vec3(uv, 1.);
    d.yz *= rot2(-0.2);
    d = normalize(d);
    float st = mix(.01, .1, abs(uv.x + .5*uv.y +
    sin(stepTime)));
    float i = 0.;
    for (; i < 1.; i += st) {
        float s = scene(p);
        p += s * d * 0.95;
        if (s < .01) break;
    }
    vec3 col = 0.5 + 0.5 * cos(colorTime + i * 3. +
    uv.xy + vec3(0., 2., 4));
    return (col - step(1., i)) * (1. - i); }
void mainImage( out vec4 fragColor, in vec2 fragCoord ) {
    vec2 nuv = fragCoord/iResolution.xy;
    vec2 uv = (nuv - 0.5) * iResolution.xy
    / iResolution.y;
    vec3 e = vec3(1./iResolution.x, 1./iResolution.y,
    0.);
    vec3 col = vec3(0.);
    int aasteps = 6;
    for (int x = 0; x < aasteps; ++x)
        for (int y = 0; y < aasteps; ++y)
            col += march(uv + vec2(x, y)
            * e.xy / float(aasteps));
    fragColor = vec4(col, 1.0);
}

```

Élie Michel

[shadertoy.com/view/3sBS1W](http://shadertoy.com/view/3sBS1W)

While we worry about compression algorithms decimating the quality of our profile pictures, the data collection business has reached a peak to the point of becoming a threat to the planet's climate as well as being a direct violation of human rights and ethics.

As citizens of the internet, we often struggle to find a balance between consciously participating in harmful economies or fighting them upright. but ironically, the constant flow of information && techno-capitalism products are a never ending resource/inspiration to glitches.

Being a glitch artist, I assess the inherent politics of any kind of medium by bringing it into states of hypertrophy. because only when a system is obviously unstable, may we all understand that is it not immuable. thus should we ask ourselves, do we have control over technology if we are breaking it ?

The cloud factory tells the story of an imaginary recycling facility in charge of processing broken files. both the artist and the machine take a playful approach by making glitch art, but it soon gets out of hand when the data stream becomes too much to handle, ultimately, resulting in a system-wide crash [spoilers].

**Kaspar Ravel**

[kaspar.wtf/project/thecloudfactory](https://kaspar.wtf/project/thecloudfactory)

Photo by Dina Karadžić

```

with open(filein, 'rb') as rd:
    print "Opening File\n"
    filesize = os.path.getsize(filein)
    chunk = 1024
    lim = args.lim
    idx = 1
    with open(fileout, 'wb') as wr:
        print "Streaming File\n"
        for pos in xrange(filesize-chunk,
            filesize-filesize/lim, -chunk):
            rd.seek(pos)
            buffer = rd.read(chunk)
            if buffer.find(b'idx1') >= 0:
                split = buffer.split(b'idx1', 1)
                rd.seek(0)
                wr.write(rd.read(pos))
                wr.write(split[0])
                rd.seek(len(split[0])+4,1)
                idx = rd.read()
                break

    if len(idx) == 0:
        print('Could not locate index!\n')
        raise SystemExit # quit
    print "Getting list of Tomatoes\n"
    idx, index_length = idx[4:], idx[:4]
    n = 16
    first_frame, idx = idx[:n], idx[n:]
    check = bytearray()
    check.extend(first_frame)
    regex = re.compile(b' *wb.*')
    ignored_bytes = (int(ignoredframes) - 1) * n
    idx = idx[ignored_bytes:]
    if glitchedframes != 0:
        end_bytes = int(glitchedframes) * n
        end_idx = idx[end_bytes:]
        idx = idx[:end_bytes]
    else:
        end_idx = ""
    idx = [idx[i:i+n] for i in range(0, len(idx), n)]
    if not re.match(regex, idx[i:i+n])

```

**Kaspar Ravel**

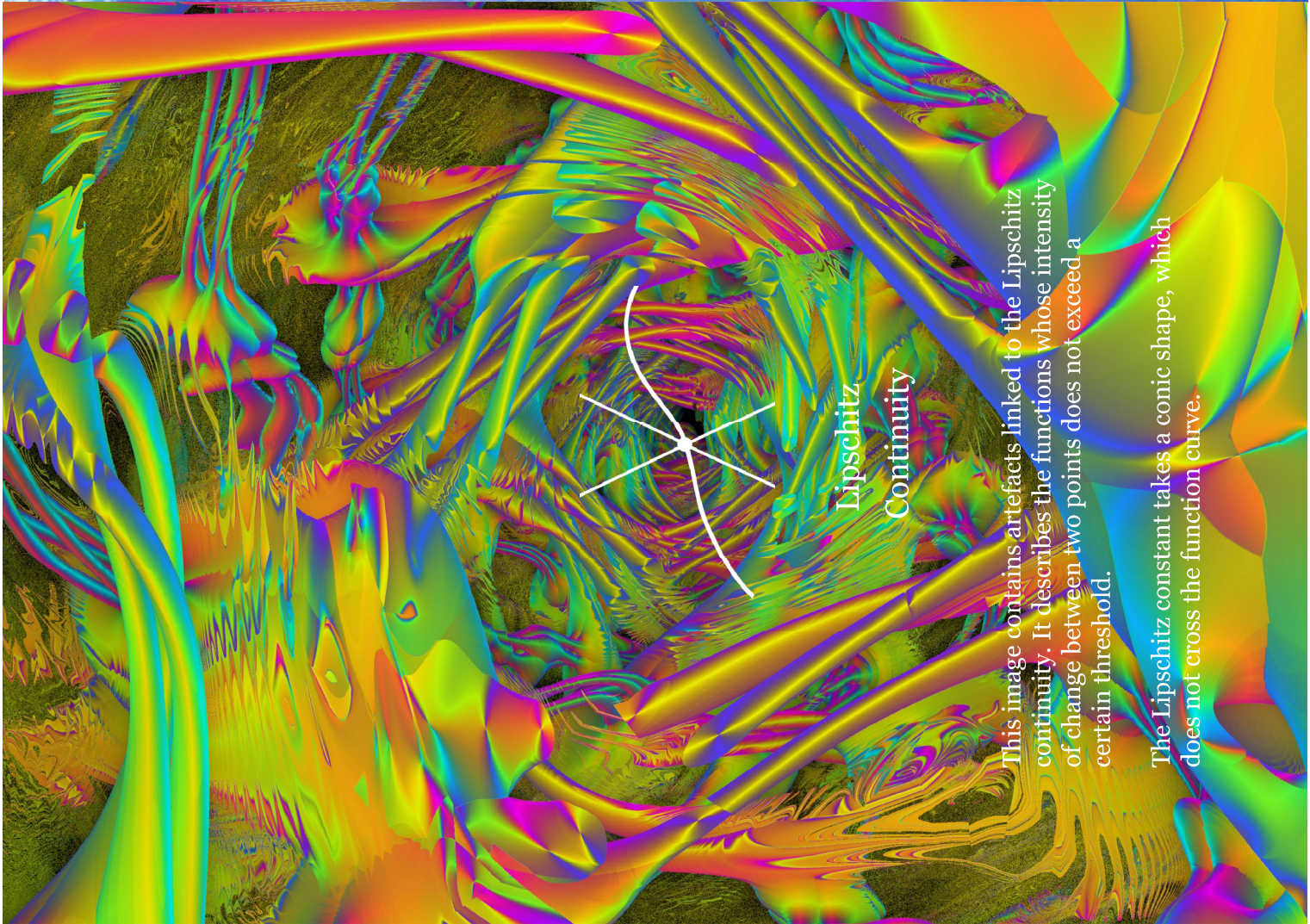
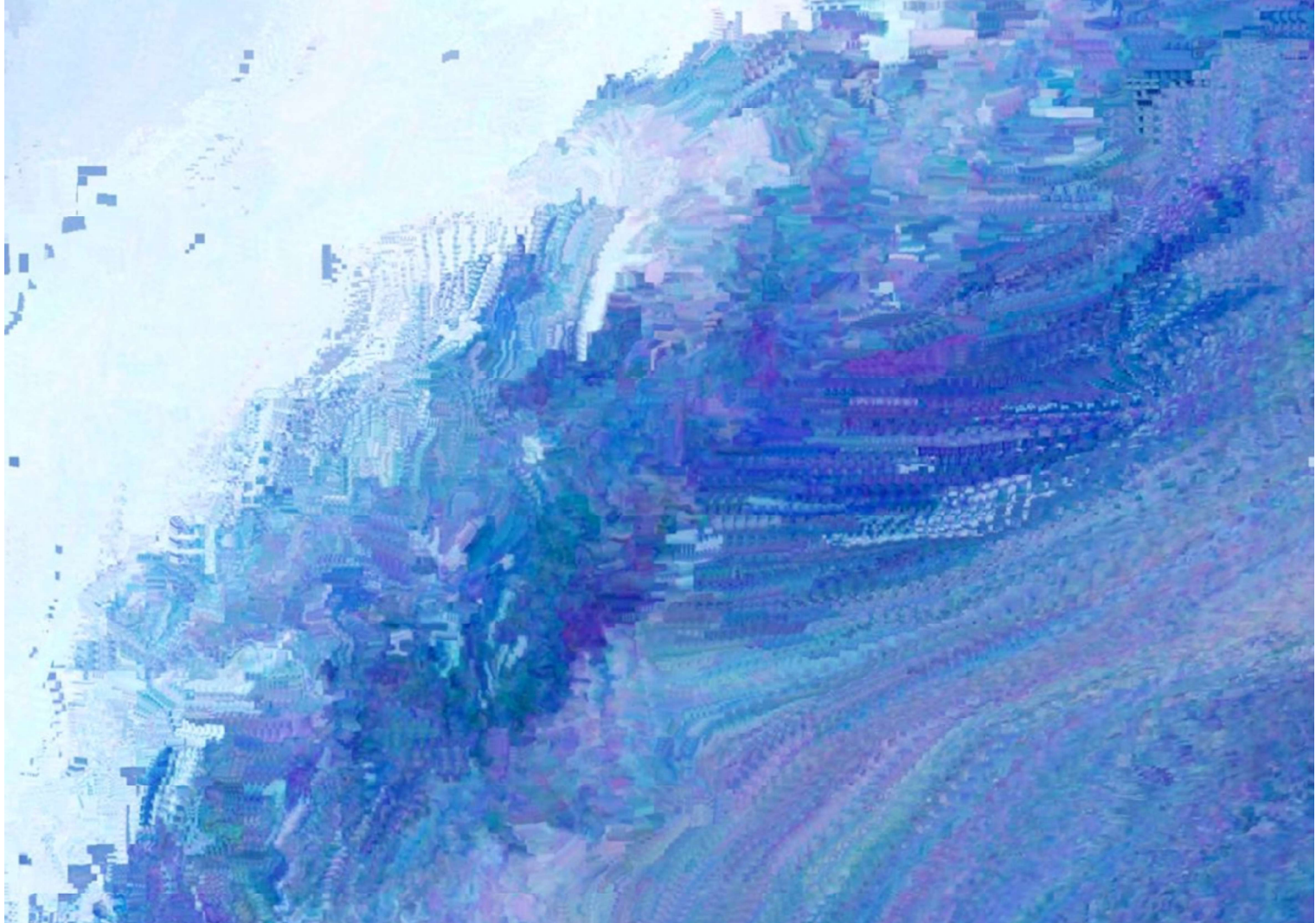
**Audio Video Interleave index breaker**

[github.com/itskaspasr/tomato/blob/master/tomato.py](https://github.com/itskaspasr/tomato/blob/master/tomato.py)

These images are the results of a raymarching algorithm whose signed distance function ventures beyond Lipschitz continuity. The visual effects produced are like fragmented optical layers of a distorted lens.

**Léon Denise**





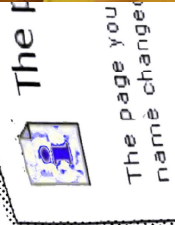
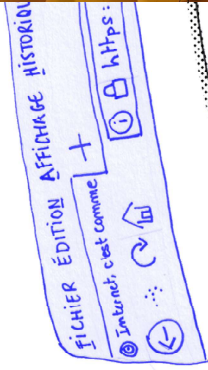
## Lipschitz Continuity

This image contains artefacts linked to the Lipschitz continuity. It describes the functions whose intensity of change between two points does not exceed a certain threshold.

The Lipschitz constant takes a conic shape, which does not cross the function curve.

# I LS PENS AIENT QU'INT

Code	Message	Signification
400	Bad Request	La syntaxe de la requête est erronée.
401	Unauthorized	Une authentification est nécessaire pour accéder à la ressource.
402	Payment Required	Paiement requis pour accéder à la ressource.
403	Forbidden	Le serveur a compris la requête mais refuse de l'accepter car elle n'est pas autorisée.
404	Not Found	Ressource non trouvée.
405	Method Not Allowed	Méthode de requête non autorisée.
406	Not Acceptable	Le serveur ne peut générer une réponse acceptable pour le navigateur.
407	Proxy Authentication Required	Accès à la ressource via un proxy nécessitant une authentification.
408	Request Time-out	Le serveur a attendu trop longtemps une réponse de la part du client.
409	Conflict	La requête ne peut être traitée car elle est en conflit avec une autre requête.
410	Gone	La ressource n'a jamais existé.
411	Length Required	La longueur de la requête n'est pas spécifiée.
412	Precondition Failed	Les conditions de la requête ne sont pas satisfaites.
413	Request Entity Too Large	La taille de la requête est trop grande.
414	Request-URI Too Long	L'URI de la requête est trop long.
415	Unsupported Media Type	Le format de la requête n'est pas supporté.
416	Requested range unsatisfiable	Le serveur ne peut satisfaire la requête de la part du client.
417	Expectation failed	Le serveur ne peut satisfaire la requête de la part du client.
418	I'm a teapot	Le serveur ne peut satisfaire la requête de la part du client.
421	Bad mapping / Misdirected Request	La requête a été redirigée vers une autre adresse.
422	Unprocessable entity	Le serveur ne peut traiter la requête de la part du client.
423	Locked	Le serveur ne peut traiter la requête de la part du client.
424	Method failure	Le serveur ne peut traiter la requête de la part du client.
425	Unordered Collection	Le serveur ne peut traiter la requête de la part du client.
426	Upgrade Required	Le serveur ne peut traiter la requête de la part du client.
428	Precondition Required	Le serveur ne peut traiter la requête de la part du client.
429	Too Many Requests	Le serveur ne peut traiter la requête de la part du client.
431	Request Header Fields Too Large	Le serveur ne peut traiter la requête de la part du client.
444	Retry With	Le serveur ne peut traiter la requête de la part du client.
450	Blocked by Windows-Parental Controls	Le serveur ne peut traiter la requête de la part du client.

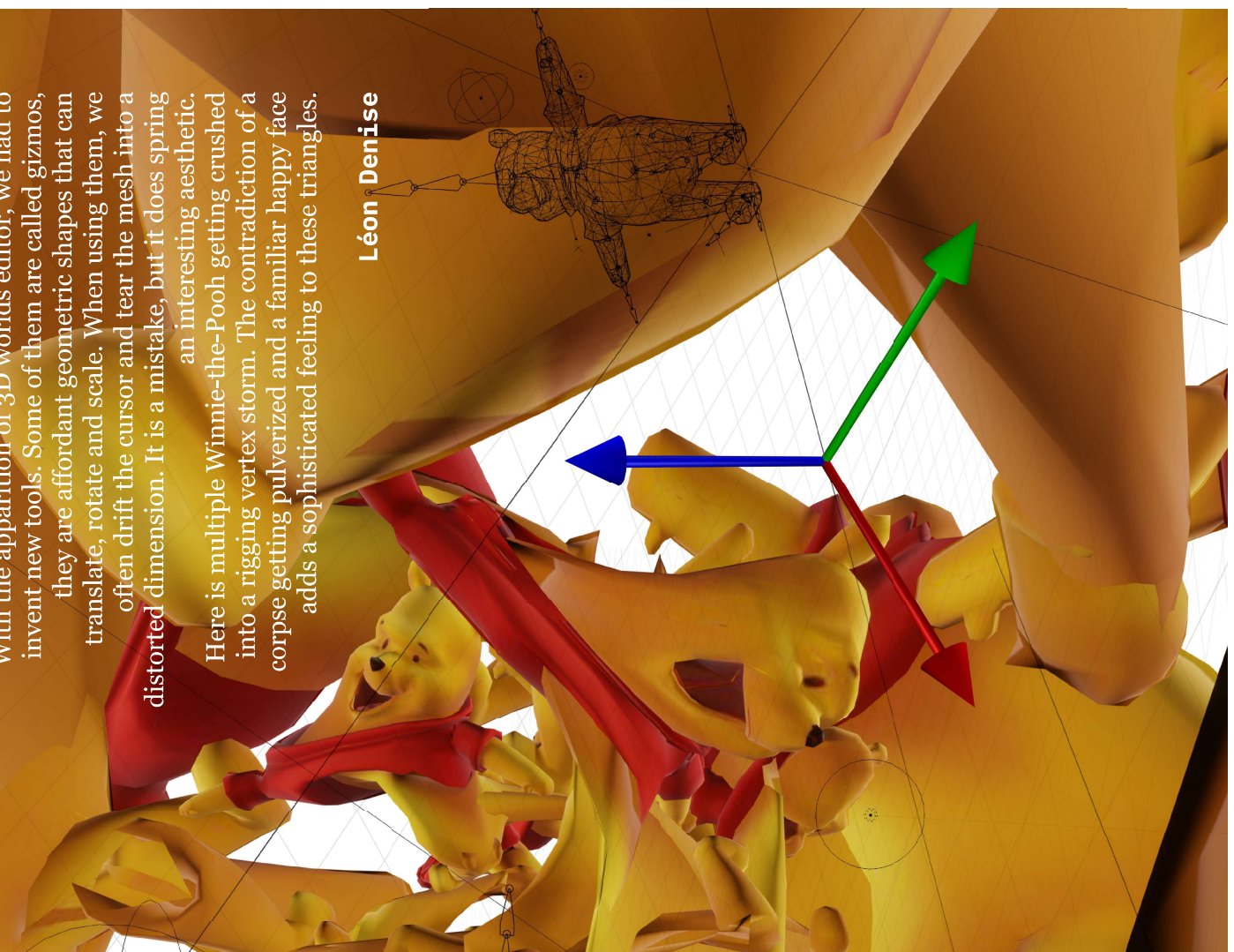


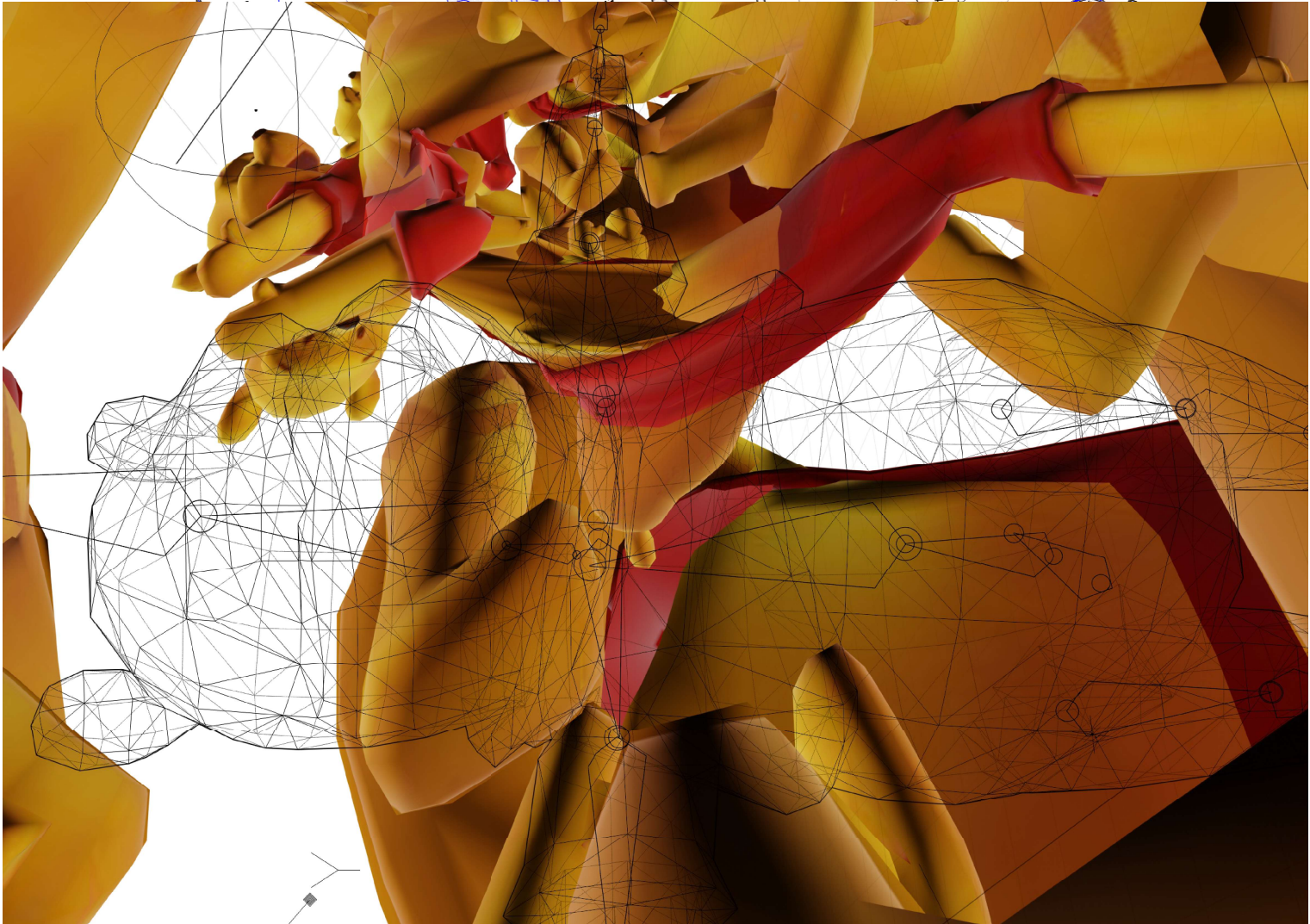
# IMPRI MEZ AVANT Q

With the apparition of 3D worlds editor, we had to invent new tools. Some of them are called gizmos, they are affordant geometric shapes that can translate, rotate and scale. When using them, we often drift the cursor and tear the mesh into a distorted dimension. It is a mistake, but it does spring an interesting aesthetic.

Here is multiple Winnie-the-Pooh getting crushed into a rigging vertex storm. The contradiction of a corpse getting pulverized and a familiar happy face adds a sophisticated feeling to these triangles.

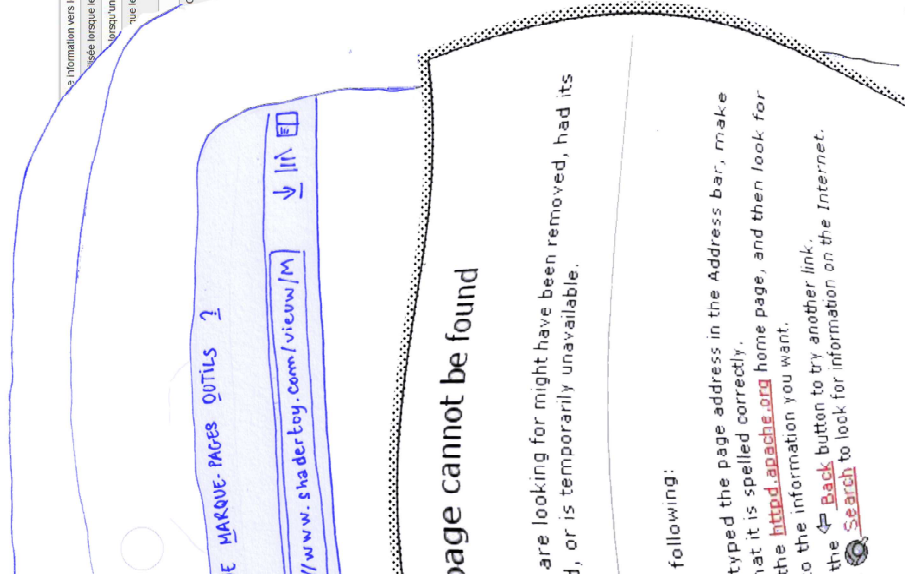
Léon Denise





# INTERNET ETAIT IMMORTEL

La information vers le client et a fermé a connexion  
 eue lorsque le client a fourni un certificat invalide  
 lorsqu'un certificat client requis n'est pas fourni.  
 que le client envoie une requête HTTP vers le port 443  
 Cette erreur se produit quand le traitement est trop long côté  
 ir a reçu une réponse invalide depuis le serveur distant.  
 sinaire écarté.  
 selon  
 straire la collection.  
 5642 %  
 se quida  
 ressources HTTP s'abolissent.  
 Utilisé par les portails captils pour renvoyer les clients  
 ligne retourne un résultat inprévu.  
 n ces échec de résolution de nom de serveur  
 sçu de réponse HTTP avant l'expiration du délai de  
 2  
 25 après que la connexion WAN ait été établie  
 envé. Elle signiait "les requêtes suivantes



# INTERNET NE S'EFFACE

## BUFFER CLEARING

CameraClearFlags.Nothing  
Other Versions - Leave feedback  
Description :  
Don't clear anything.

This will leave colors and depth buffer from the previous frame or whatever was displayed before. See Also: Camera.clearFlags property, camera component.

Did you find this page useful? Please give it a rating: \*\*\*\*\*

Report a problem on this page

-----  
Question from armoredpokey · 03/10/12 at 16:18 · camerabackgroundstandalone:

Screen background not clearing in standalone/web player build.

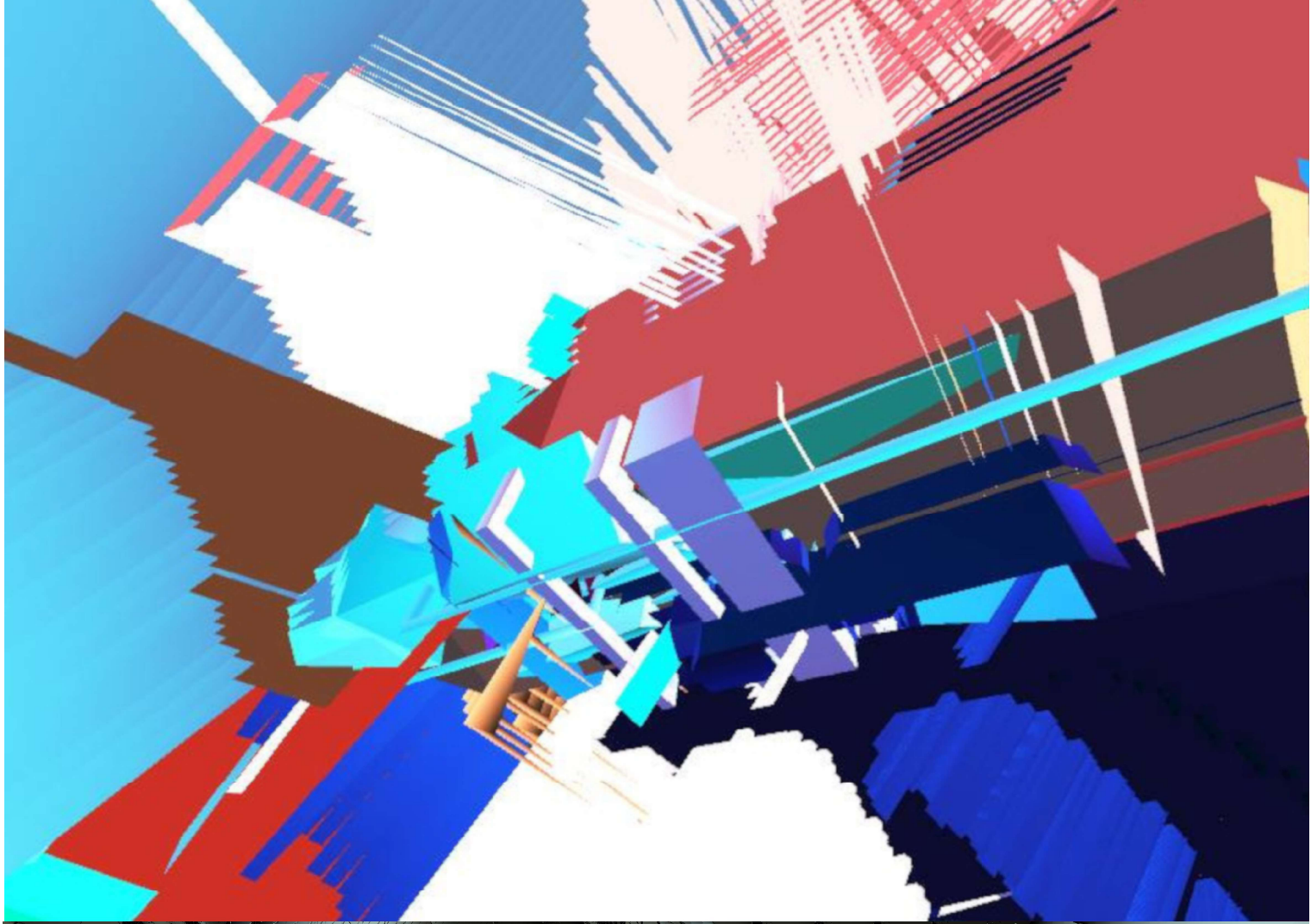
I have a camera object whose position changes based on player input. In the Unity editor, this works great - there is a small screen that moves around on a black background and generally looks awesome.

The only problem is when I export to web player or standalone, for some reason the background doesn't clear after my camera's position changes. Instead I get a smear of camera scene that follows my screen.

Any ideas as to how to fix this? Ideally I'd like to just set some flag that causes the background to clear to black and then have my new camera scene rendered on top of it, every frame. Would adding another camera layered underneath it work, and if so, what's the best way to get it to render black?

**Benjamin Vedrenne**  
glkitty.com





When you work with numbers in the digital realm, computers have to approximate rational numbers. The precision of this approximation starts to fade away whenever you are using huge numbers. In a 3D space, if you navigate for a long time in the same direction, you will approach an unstable state called floating jittering space. It breaks the continuity and produce aliasing.

This image is an example of how jittering can produce interesting shattered geometry.

**Léon Denise**



This image is produced by a filter inspired by datamashing. An activity that consists into modifying compressed datas.

A digital video holds a complete list of colours every seconds. Between those keyframes there is relative images.

Lists of pixels that are not colours, but ponderation vectors, describing how to update colour relative to previous images.

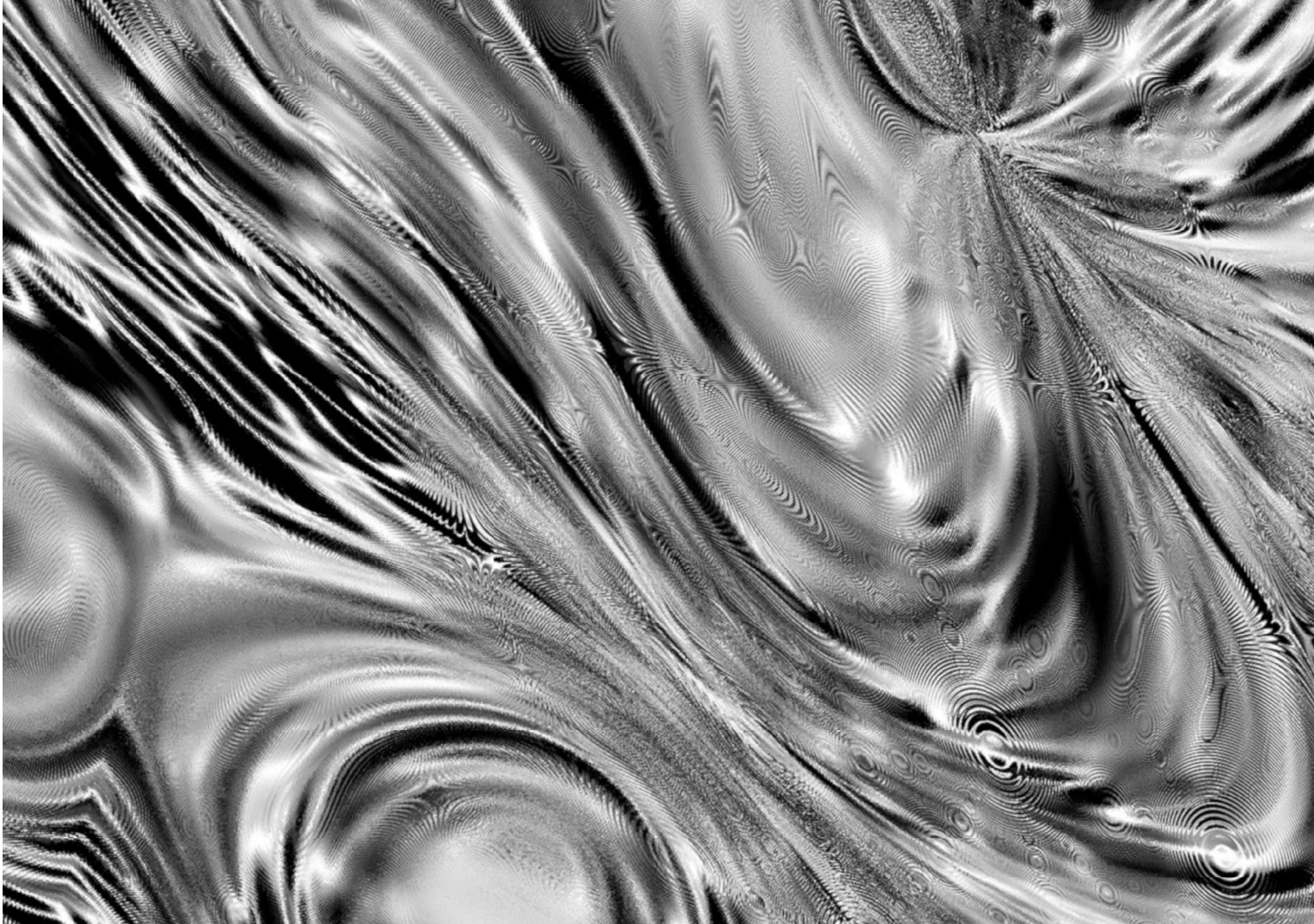
When we modify the sequence of key frames and relative images, the video player improvise and produce an intriguing image. Beyond the aesthetic, what pleases is also that form reveals function. The way the compression algorithm is thought generate different graphic styles.

Thus we can imagine an artistic way of compressing datas to produce willingly a particular visual effect.

The background image is the result of an interpretation of data compression. Each pixel is updated if the distance between his colour and video's pixel colour is superior to a certain threshold.

The pixel is translated from his own colour. The gray value of a colour can transform into an angle, and then into a direction. The pixel drifts as it changes colour. As a painting, pixels flow along the screen.

**Léon Denise**  
[github.com/leon196/OpticalFlowExample](https://github.com/leon196/OpticalFlowExample)



"Le paysage, ineffablement assoupi, avait cette moire magnifique que font sur les prairies et sur les rivières les déplacements de l'ombre et de la clarté."

"The landscape, calm and peaceful, displayed the enchanting hazy effects of light and shade over the fields and river"  
(Victor Hugo, Quatre-vingt-treize, 1874)

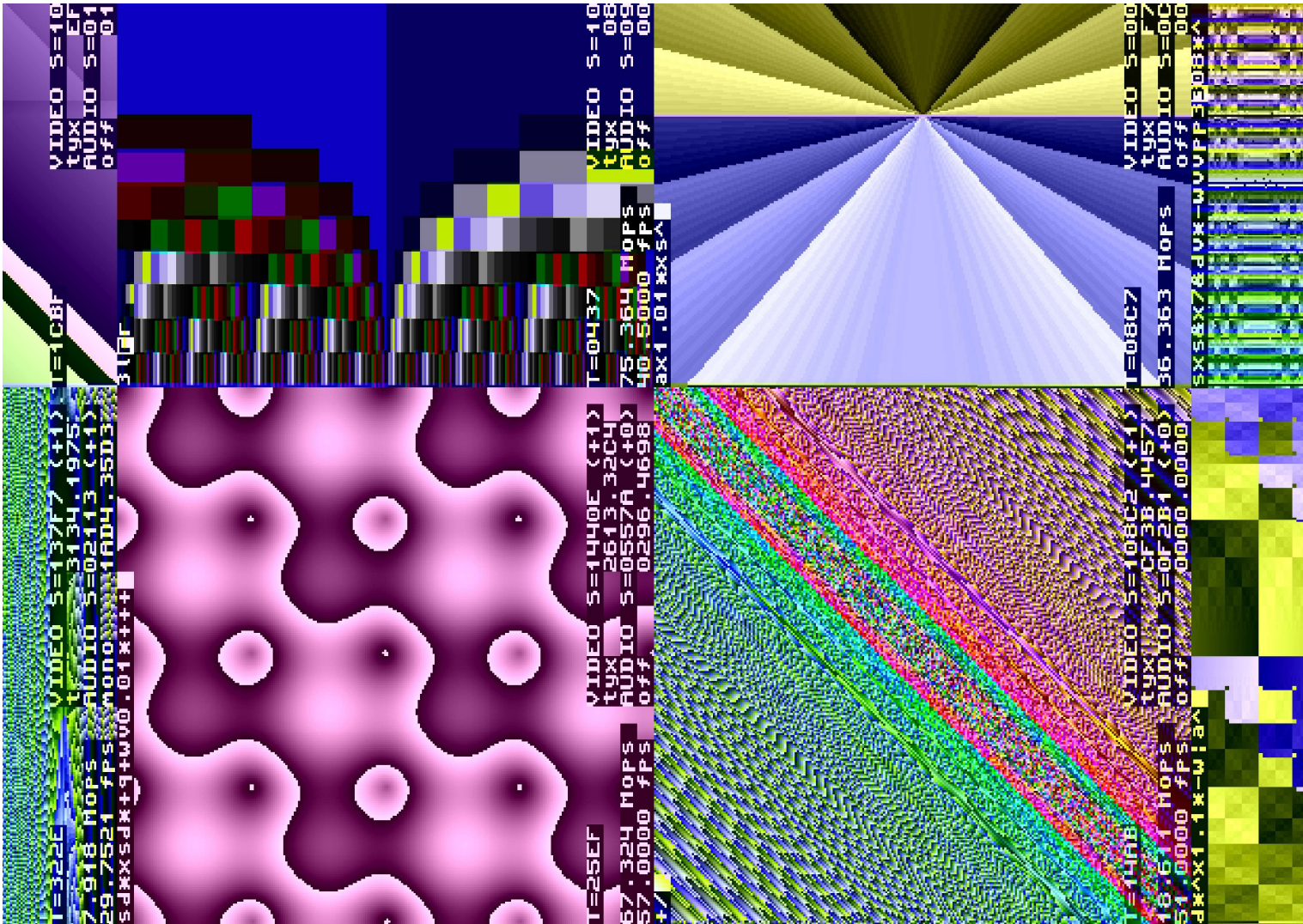
"Moiré" is an optical phenomenon, whose description does not reach a consensus: sometimes as "undulating reflection", sometimes as "interference patterns between networks", or even as "alternance of ripples, of mat and glossy welts." They can occur as wind flows on a landscape, or on the surface of a quiet lake. The term was first used to describe a kind of shimmering fabric, on which concentric reflections can be seen, changing with light.

Starting with the technological era, the term adopted a new usage, although it still describes the same optical effect. Indeed, digital information is made of samples, cut into little bits which are then stored, transmitted, filtered, copied, distorted... These bits are pixels in the case of images, or voltage curves for sounds. Thus, the sampling resolution becomes a critical topic: how many pixels are needed to accurately reproduce a picture? And if some elements are smaller than a single pixel, what is their fate? For instance, a grid, or a frame, where each gap is smaller than a pixel, could not be reliably depicted. Gaps may appear on a pixel out of two, of nine, twelve, twenty, depending on the scale ratio between the offset and the pixel. At this moment, an effect of stripes or contrasted curves appears. For digital photographers, or 3D artists, to name a few, this effect is often unintended, unintentional, and disturbing. The high contrast of moiré takes precedence over the original intent of the artist, and the latter aims at getting rid of the artefact.

Through its high contrast, its unexpected shapes, apparently without any physical basis, and above all its spontaneous emergence, which is difficult to control, moiré continues to fascinate.

Benjamin Vedrenne  
glkitty.com









VIDEO S=10004 (+1) T=1035 VIDEO S=08584 (+F)  
 One of my professors recounted his first <sup>intro</sup> years as a researcher. At that time, programs were written to punch cards, and executed at a snail's pace. Therefore he could only test his program twice a day, at best. To be efficient, one had to be cautious and exigent.

|||||

Nowadays, millions of lines of code are compiled in a few seconds. Capitalism market drives uncountable students in a digital eldorado where companies survive through delivering half-baked products. Quality is not the priority, and we are affected, from personal (work conditions, self-expression) to universal levels (planned obsolescence, global warming).

|||||

I found an interest in non-conventional ways of programming, because it forces me to slow down and thoughtfully think about my code. This requires discipline, patience, and <sup>EO</sup> factually puts me in a zen state. I may type very slowly, but each letter would be the result of a clear understanding of my footsteps.

IBNIZ is such a framework, which generates visuals and sounds from a very concise code. Each character has a meaning. They actually instruct the processor to hold values in memory, and do some mathematical operations with them. The result is a colorful, fractal-oriented animation, and a somewhat experimental music.

In every tile shown on this page, the source code is written in the upper left corner. Have a look, and try for yourself.

Jonathan Giroux (+1)  
 youtu.be/vgcQ6k1rEFA  
 IBNIZ by Viznut 592 a7#1% .01%+s  
 pelulamu.net/ibniz/

T=031C  
 VIDEO S=003EE (+1)  
 TYP CC33.FF35  
 AUDIO S=0F804 (+0)  
 off 30D0.BAEF

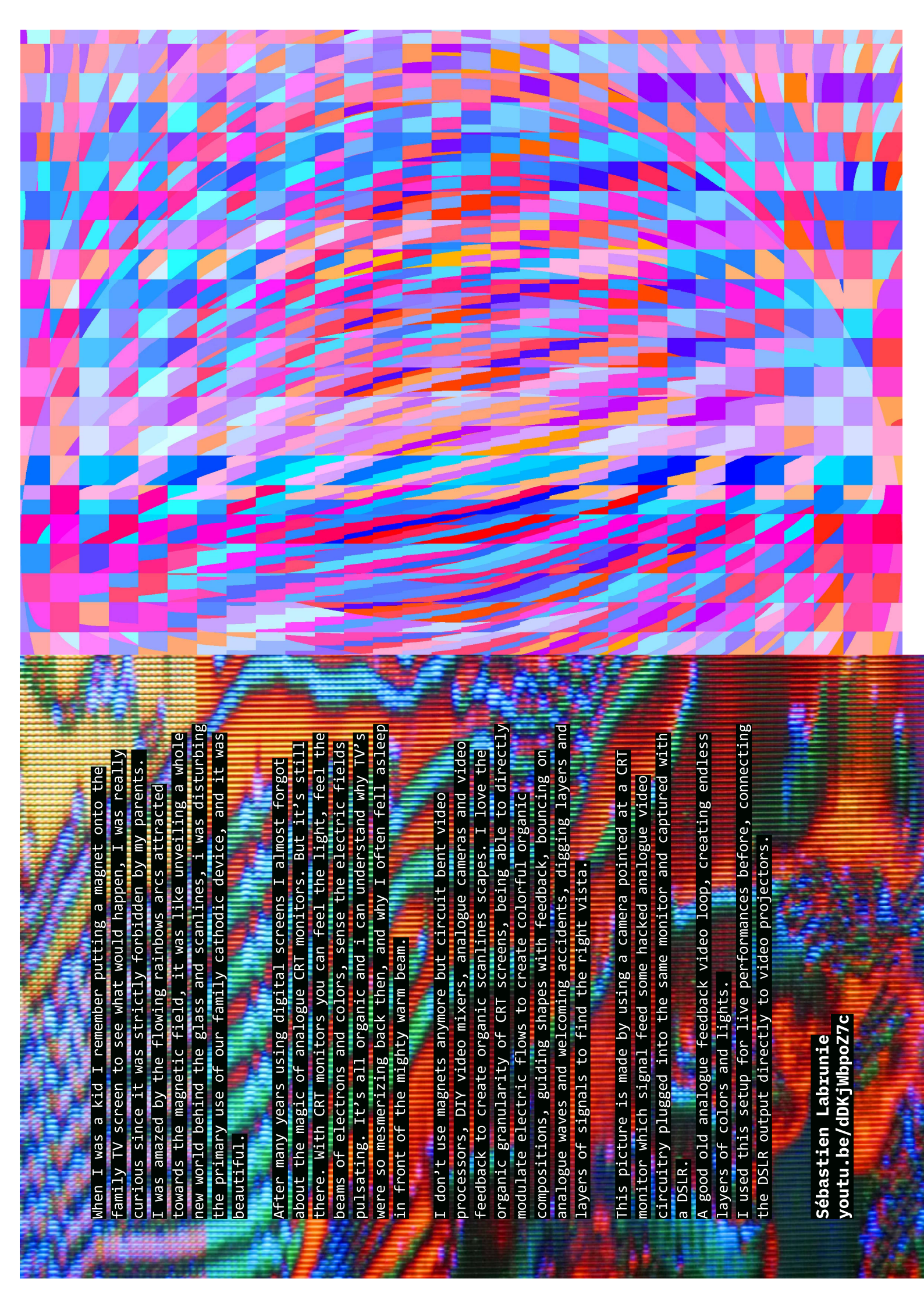
18000  
 12 FOR HELP

```
void main () {
  vec2 uv = gl_FragCoord.xy/resolution.xy;
  vec4 video = texture2D(videoMap, uv);
  vec2 unit = 1./resolution.xy;

  float a = luminance(video.rgb) * TAU;
  uv += vec2(cos(a),sin(a)) * unit;
  a = luminance(texture2D(backbuffer, uv).rgb) * TAU;
  uv += vec2(cos(a),sin(a)) * unit;
  uv -= normalize(p) * unit;
  vec4 frame = texture2D(backbuffer, uv);

  float should = colorDistance(frame, video);
  should = smoothstep(.3, .4, should);
  gl_FragColor = mix(frame, video, should - .01);
}
```





When I was a kid I remember putting a magnet onto the family TV screen to see what would happen, I was really curious since it was strictly forbidden by my parents. I was amazed by the flowing rainbows arcs attracted towards the magnetic field, it was like unveiling a whole new world behind the glass and scanlines, i was disturbing the primary use of our family cathodic device, and it was beautiful.

After many years using digital screens I almost forgot about the magic of analogue CRT monitors. But it's still there. With CRT monitors you can feel the light, feel the beams of electrons and colors, sense the electric fields pulsating. It's all organic and i can understand why TV's were so mesmerizing back then, and why I often fell asleep in front of the mighty warm beam.

I don't use magnets anymore but circuit bent video processors, DIY video mixers, analogue cameras and video feedback to create organic scanlines scapes. I love the organic granularity of CRT screens, being able to directly modulate electric flows to create colorful organic compositions, guiding shapes with feedback, bouncing on analogue waves and welcoming accidents, digging layers and layers of signals to find the right vista.

This picture is made by using a camera pointed at a CRT monitor which signal feed some hacked analogue video circuitry plugged into the same monitor and captured with a DSLR.

A good old analogue feedback video loop, creating endless layers of colors and lights. I used this setup for live performances before, connecting the DSLR output directly to video projectors.

**Sébastien Labrunie**  
**[youtu.be/dDKjmbpoZ7c](https://youtu.be/dDKjmbpoZ7c)**

```

float rand (float x)
{return fract(sin(x)*256.43);}

float HexDist (vec2 uv) {
uv = abs(uv);
return max(dot(uv, normalize(hr)), uv.x);
}

vec4 HexGrid (vec2 uv) {
uv *= detail;
vec2 ga = mod (uv, hr)-hr*0.5;
vec2 gb = mod (uv-hr*0.5, hr)-hr*0.5;
vec2 guv = dot(ga,ga)<dot(gb,gb) ? ga : gb;
guv = fract(guv);
vec2 id = uv-guv;
guv.y = .5-HexDist(guv);
return vec4(guv.x, guv.y, id.x, id.y);
}

void mainImage( out vec4 fragColor, in vec2
fragCoord )
{
vec2 uv = 2.*(fragCoord.xy /
iResolution.xy)-1.;
uv.x *= iResolution.x / iResolution.y;

uv.x -= 0.25;
uv = mix(uv, floor(uv*15.)/10.,
length(uv));
float hx = HexGrid(uv).x;
uv.y += fract(uv.x*5.);
vec4 hc = HexGrid(uv);
vec3 col = mix(vec3(rand(hc.z)*0.8,
rand(hc.w), 1.),
vec3(1., rand(hc.z)*0.6,
rand(hc.w)),
step(0.05, hc.y+sin(time)));
col *=
smoothstep(0.05,0.4, hx+sin(time*2.));
col = pow(col, vec3(0.6));
fragColor = vec4(col, 1.0);
}

```

**Flopine**

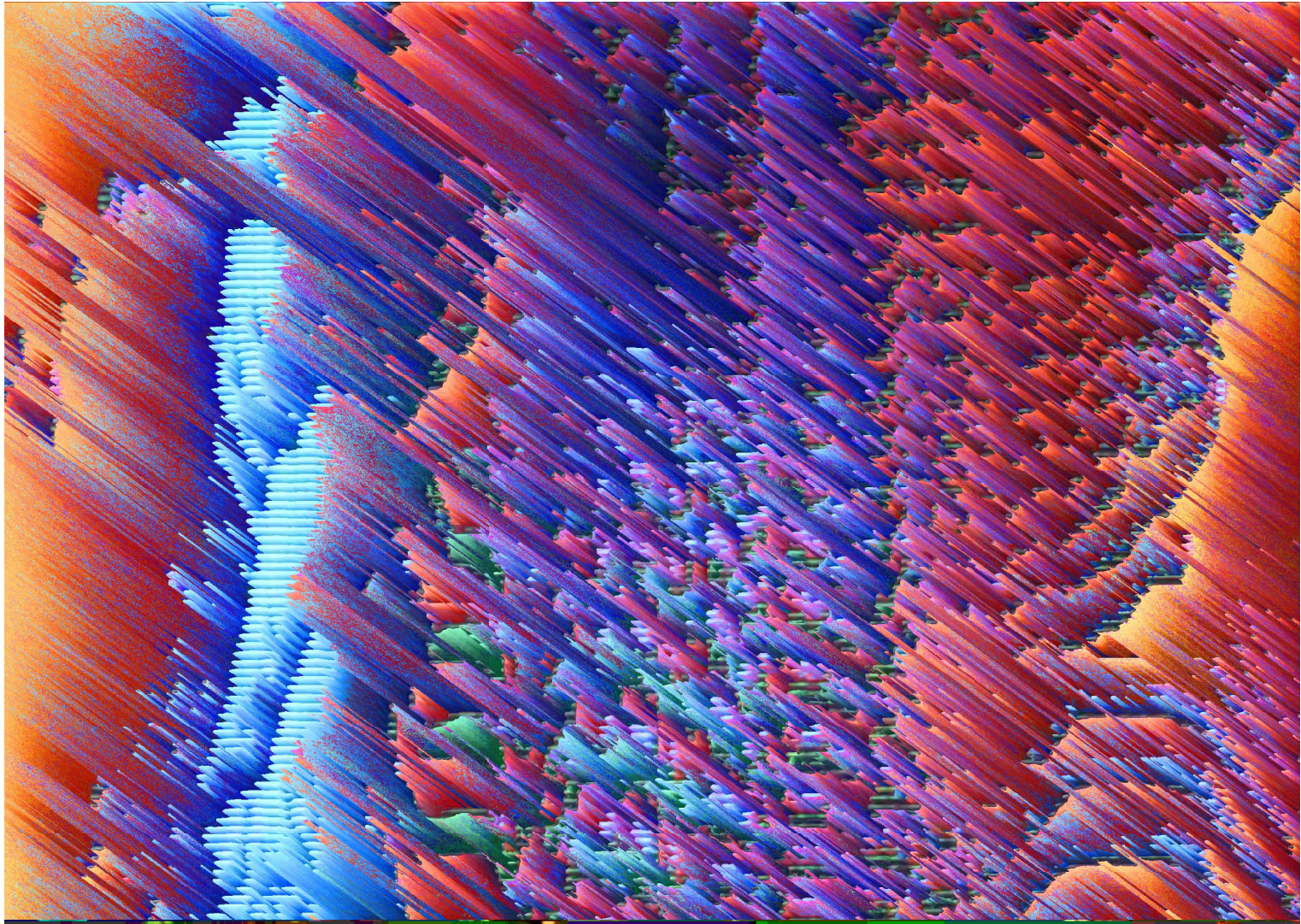
[shadertoy.com/view/wdSSMw](http://shadertoy.com/view/wdSSMw)





d' \_O= xàTVO0ãD0+e&>| N'' @A0~!  
S' fCØ//r.ÆA3\*Z&skAãadvø«T'sUB'²D5÷-1'  
c \$ZT³...4zà«U),µPLE0°,A  
=Pwdr' /x.ú | m?%@Zi·0»Eàu0,,%iAatVPæ~ eapq  
iPl32xãG1=ú±0E' :VèB,7&p İlmİ --{fU36i ¥Yããt, İ  
&YS' »0ÈE]E%YfEi}µwı÷\$V'' /;²Yİf→eàoİB,~ãh;ØQ b,±mQZİ2]  
\*JÉDü <ãHİç, »«}ëA  
N' \*bfãE0J,SU' Sidh 'NãZëü4E' cı÷·G}J«-Cıp' /B\*seh-  
ËYzi' EEU#>İ\Zd'p''²P[INyOsøAŞ±™, ù  
ed\$ cõİİz, sYI, ÆgOYİcB•JÉYããEY@ògn~leIEùLZK;Øq.ı00İ  
ç;ıUA@-òS' WA' Sx' T'KZY'ù«ıİD, çİ@Ø4, ù#àà. ~<ã8#YEN\_ãh  
'AEãµ-ge0\* ¥-eN-ÆU0Wz  
-Uxı0gmF0eòZİT0S'«0' JxpSTö|' 6İµ0REtZG0ú~we0f  
»Q0t, jk-8e0fçB), 6y...0c#è, q' à Z' C-,™.ç?  
hİ}}/çvqøİã#|çtçİc  
üüTX,ÆİBü0Jü0T0n'ACu' yëw'İ=--  
İU'xæxµMaU' f'ra'ıİç;ç'İP, İ'sıfıJàiYpı/Y'Yëã'}C}  
Sç0Aw'xUı'ı'8&ıüwye ^0ü3İA'ı'ëëJw...ò'ç'ç'pıYüzy  
RYI'ã'Üsöa ^,;0=ıWQŞİ'Êe, ^Êf-90ãç  
ı'ı'ı'ı'ı'ã0İ'v™g'ÄxvPK9çYfEİ-?ë, @K¥fEİ"ãbu«Bè0''3fb  
x' óZ••, ÊepCà=0By4k'3EãEuaıçıYı'ÊU7·ø U'·öüıP@  
)p]3İ]  
çıµ9zu0ü. ·Dàİs,»0ãŞYzç' pH; =tİP¥ıys-ü... ç8zç6' -\$.³A0  
'--00COX-°y'ZçS0-™OHäİEçZKYİ' İİ4g-40Aè&ãA\_İAb]Ø'  
t4e&Ş, ZrãŞ  
Oİã°ıNv'  
ZXdYİ·òbŞU'H' hA± 8ŞD Ga(YüsB>ds İüex'w,½'Ä6'  
b00c0İ,ZİzCçD' 0'şYçü.~çıy1;f™Yİİbb, Hã=»İ'şY  
Çyã ZşpE-»=ıYkİfãaaeöñİMçY  
ÜFN'™cããñöã--  
İİı, w=İ'zEİ\ ZçöPıYı, ggpıµyãmİñüJ, °cã#ı, p' <#3 OEŞf™0  
Xhf HM ØãWfö±) %DBföx' wA=-ãGDNA³ TÖYkVç@AEİE \_ı: @YAD  
İ.»,ZöptÉfE=ëA+, İ~ãŞU't8.0g'+7ıYÖ]wã0 ZÜİ'  
<w\*ÄZÜİø?ıkç-òx|ıxı-0İİİç' m.İü³, İëãıü™g~eGñçøã5ıYıV<üP  
p08ı,öY-Aİëë"ñYEWZÇEİTİSÜYŞDµ-çøµã áırd} Hk  
d''sp³öüEİç)çöPıP |uç |öçPpèİ^àooüİİn  
Yı0A...zçç6 İçıuü  
>ocçã-ıİßİü™KÜ'u'ÜZx0ëıKı>öZæy»|b' aöYp-ãİx4ı'Y

Sébastien Labrunie  
youtu.be/dDKjmbpoz7c



```
// VCR style inspired from https://www.shadertoy.com/  
view/ldjGzV  
  
#define t iTime  
#define rot(a) mat2(C=cos(a),S=sin(a),-S,C)  
float C,S;  
  
void mainImage( out vec4 fragColor, in vec2 fragCoord )  
{  
    vec2 uv = fragCoord.xy / iResolution.xy;  
    float hs=step(0.3,sin(t+2.+4.*sin(t*9.))  
        *(sin(uv.y*10.+t*5.))+sin(t)*sin(t*20.)*.5)*.05;  
    uv.x+=hs;  
    float vs = step(0.8,sin(t+2.*sin(t*4.)))  
        *(sin(t)*sin(t*20.))+cos(t)*sin(t*200.)*.1);  
    uv.y=fract(uv.y+vs);  
    vec3 c=texture(iChannel0,uv).rgb;  
    c.xy*=rot(iTime*.3);  
    c.yz*=rot(iTime*.5);  
    c.xy*=rot(iTime*.7);  
    c.yz*=rot(iTime*.9);  
    fragColor = vec4(floor(5.*abs(c)).*2,1,0);  
}
```

**Jonathan Giroux**  
[shadertoy.com/view/4sffzS](https://www.shadertoy.com/view/4sffzS)

From broken technology to affective artifacts, Impression Numérique is an extension to the Impressionist art movement of the 19th century. The works in this series aim to challenge our perspective on glitches & all that is considered as faults in today's new media standards.

The development of an impressionist glitch art form is a reaction to the constant development of "newer" and "better" technologies.

We often think of digital art as being cold and rarely affective, but we never give it the opportunity to express itself for what it is. We spend our time contributing to commercially pushed HD standards, hiding flaws, rejecting low quality, and upscaling our media using tools that are inherent to these systems.

Using random popular culture movies as a source, I disrupt compression algorithms and corrupt their video files to create stills where the pixels are set free. While colors and motion take liberated paths, the spectator is left with the image of what happens when binary data becomes a malleable entity with its own means of expression.

Here, the focus was set on the fleeting visuals produced by broken files desperately trying to be displayed correctly in VLC. The resulting images were then reinterpreted and rescaled through deep learning, thus creating imaginary details.

Being ultimately dependent on system flaws, these artworks do not only exist in the glitch momentum, but are also active participants in [today's fight against hard cut technologies] & net existentialist's theory - that digital art cannot live on without digital error.

**Kaspar Ravel**

**[kaspar.wtf/project/impressionsnumeriques](http://kaspar.wtf/project/impressionsnumeriques)**

De l'ère du programme,  
accueillons la déesse

Perfection impatiente,  
omnisciente on la sait

Vêtue de quelques bits  
elle enchante la détresse

Impalpable, inodore,  
sa texture disparaît.

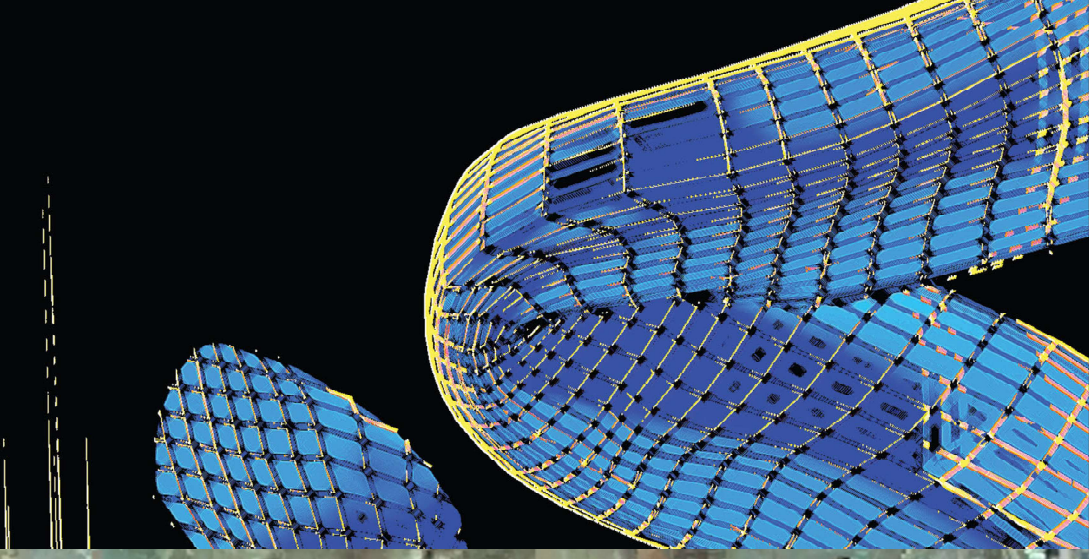
Visages inexpressifs,  
paquets d'individus

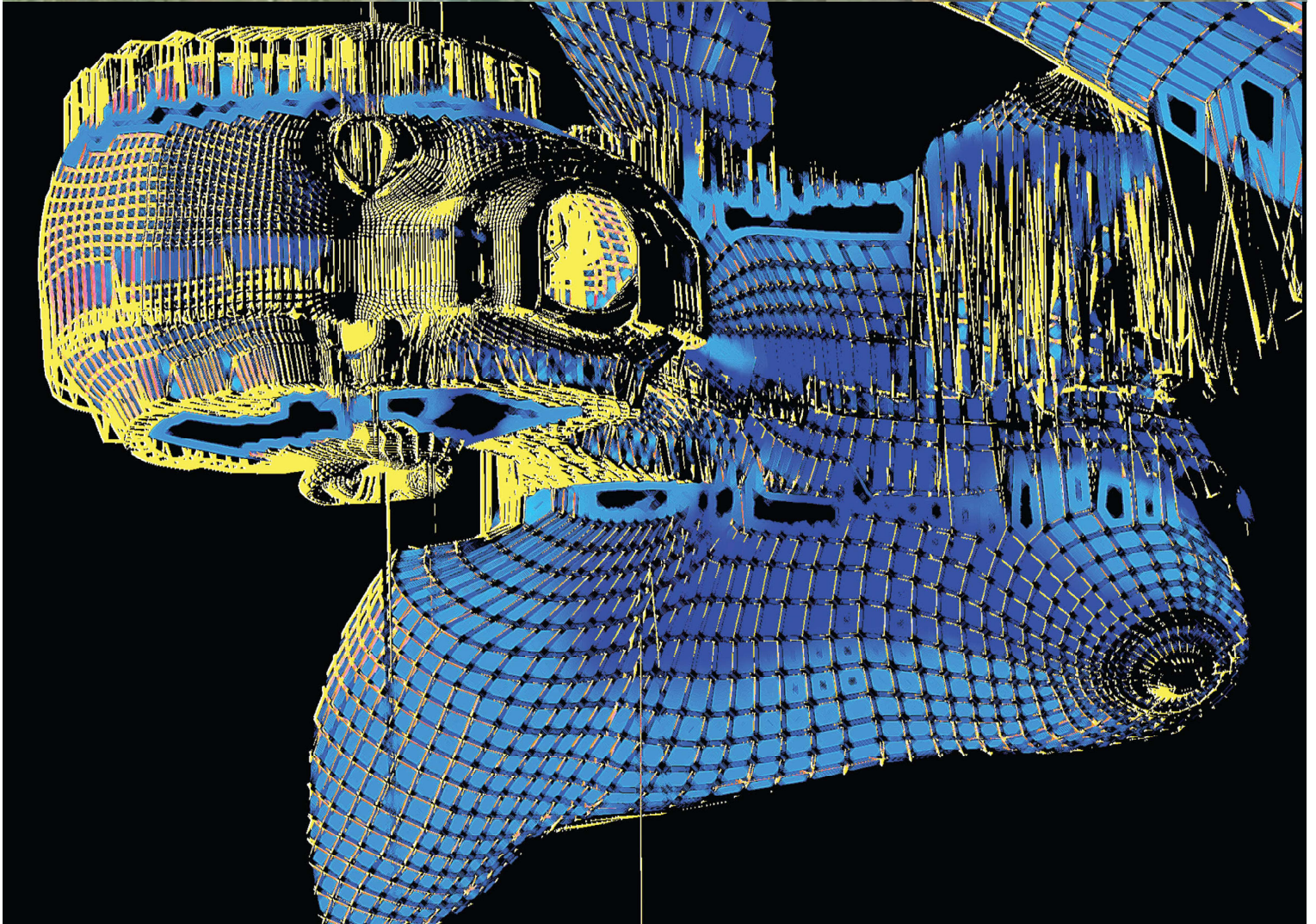
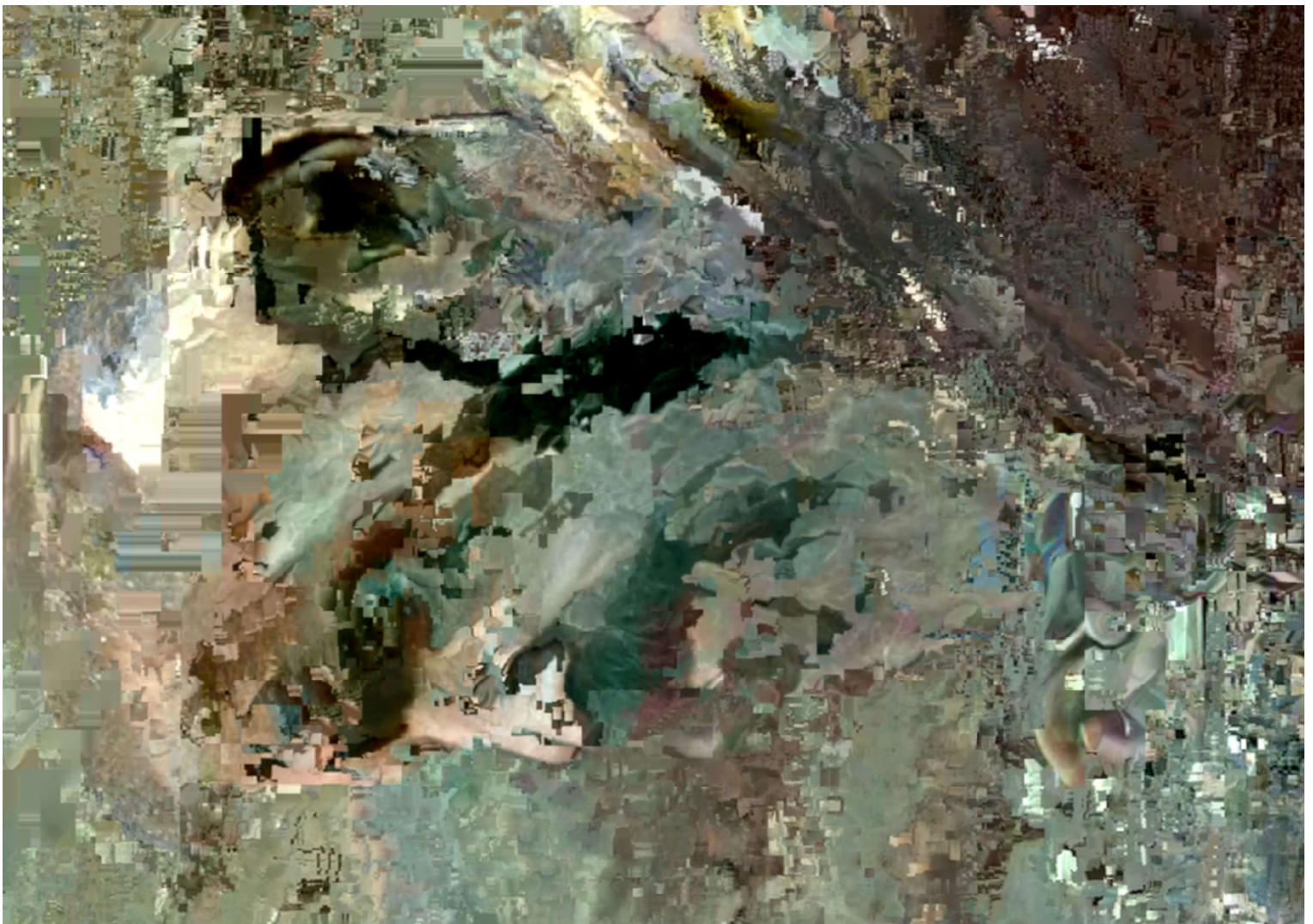
L'hagard voit défilier  
machines tissant leur toile

Grillages de pixels,  
sémantiques disparues

De l'erreur et de l'âme  
on a tombé le voile.

**Clara R/**







UNINITIALIZED  
REFERENCE MEMBER  
IN STRUCT

LINE  
35  
C048

THE VARIABLE "IT" WAS  
DECLARED BUT  
NEVER REEVALUATED

56G52T4+ - 02  
F A U T

ERROR: 'PNGLBINDTEXTUREPROC'  
DID YOU MEAN: 'PNGLBINDTEXTURES PROC'

We expected;  
before;

UNDEFINED  
IS NOT  
A FUNCTION!

A GREAT PART OF THE  
PROGRAMMER'S WORK  
IS TO FIND THE LINE THAT  
CAUSES AN ERROR. THE  
BUG HUNT CAN BE EXTREMELY  
LONG AND TEDIOUS DEPENDING  
ON THE DEBUGMENT TOOLS  
AND CAN TAKE DAYS.

THE COMPUTER SENDS US  
AN OBSCURE MESSAGE ABOUT  
THE NATURE OF THE BUG. THIS  
INFORMATION IS OFTEN RED, LIKE  
A BLOOD STAIN IN A CRIME SCENE.  
AND AS A DETECTIVE, WE SEARCH  
FOR A HUNT. THE COMPUTER IS AT THE  
SAME TIME: WITNESS, ACCUSED,  
SNITCH AND VICTIM. AND THE  
GUILTY WILL ALWAYS BE THE  
H U M A N ...!

YOU ARE MISSING  
STATEMENT  
NON-VOID  
RECIPE  
TARGET  
FOR  
ALL  
TAILED