

**Quels futurs pour  
le glitch art ?  
*Quête de sens  
nouveaux à l'aube  
d'une cristallisation***

Philippe Mion  
Mémoire du DNSEP design sonore  
2022\_2023

## Sommaire

	Introduction	7
Partie 1 →	GLITCH	13
1_1	Définitions - glitch et art	16
1_2	Origines du glitch art	17
1_3	Une histoire de la musique glitch	24
1_4	Russolo et Cage comme précurseurs du glitch	26
1_5	Les traces de glitch dans d'autres formes d'art	28
1_6	L'esthétique du son glitch	33
Partie 2 →	FUTURS	37
2_1	Théorie de l'information	40
2_2	Archéologie des média	43
2_3	Témoignage - La place de l'intelligence artificielle dans l'esthétique de l'erreur	45
2_4	Entretien avec le collectif anthropie	49
2_5	Quête de sens pour un renouveau conceptuel	63
2_6	Le post-digital	74
	Conclusion	83
	Bibliographie	85
	Webographie	87
	Remerciements	89

Entretien avec le collectif anthropie

## 2\_4: Entretien avec le collectif anthropie

Salut Anthropie, vous êtes qui et vous faites quoi?

Salut Philippe, anthropie, qu'on préfère écrire sans majuscule, est un collectif d'écriture basé en Suisse romande. On insiste toujours sur l'écriture, parce que c'est notre point de départ. anthropie est composée d'entre 5 et 7 personnes, selon les projets, qui écrivent plus ou moins collectivement des textes (*Début*, en 2018, *Dio* en 2020 et *Extinction Piscine* en 2022).

Ces textes sont des objets hybrides, qui empruntent à la science-fiction, à la fable, au manifeste politique et un peu à la poésie contemporaine. Leur forme est prévue dès le départ pour s'adapter aux contraintes de la vidéo: chaque ligne du texte doit former une unité syntaxique et être assez courte pour tenir intégralement dans une *frame* au format 16/9ème. Ça a l'air compliqué dit comme ça, mais en pratique ça donne juste des textes en vers libres, contraints à une narration épurée et où le style doit rester simple, travaillé avec une esthétique proche de la punchline ou du slogan.

Une fois le texte terminé, on le partage en libre accès sur [www.anthropie.art](http://www.anthropie.art), on le publie parfois en version papier et surtout on commence à l'adapter. L'idée d'origine c'était de faire de la « littérature hors du livre » comme on dit maintenant, dans une version un peu pirate, un peu politisée, un peu machinique. On découpe et on remonte des extraits de nos textes dans différents médiums, en vidéo, en stickers, en affiches, en algorithmes, en *bots* twitter, en virus informatiques, etc. On s'est vite intéressé.es aux arts vivants qui sont devenus depuis l'autre colonne vertébrale du collectif. Durant la séquence 2020-2021, on a travaillé à faire du texte de *Dio* une performance, qui mêle lecture live, modifications de voix, musique et

projection du texte-comme-vidéo. On l'a positionnée dans le champ culturel comme un « spectacle », accueilli dans des théâtres contemporains, mais on a exploré aussi des formes de théâtre pirate, dans une veine DIY.

Depuis quelques mois, on travaille sur *Extinction Piscine*, un nouveau texte à paraître, qui va donner lieu à une version adaptée en « story » sur Instagram et à un autre « spectacle », tout ça éclaté sur le cours 2023/2024. En parallèle, on bosse aussi sur quelque chose d'un peu différent qui t'intéressera peut-être (qu'on appelle pour l'instant la *Théorie du bug*, mais qui sera probablement renommé en cours de route): un « manuel » ouvert, en ligne, dans lequel on recense et on explique nos différents protocoles de glitch (des tutos tout ce qu'il y a de plus concret, mais avec un peu de théorie politique dedans), parce que c'est effectivement au centre de la partie « visuelle » de notre travail.

C'est quoi votre définition du glitch ?

Difficile à dire, mais on se lance. Pour nous, le glitch c'est à la fois une apparence (un résultat visuel) et un ensemble de protocoles. C'est le résultat enregistré ou converti d'opérations réalisées par une machine ou un logiciel qui n'a pas été prévu pour ça. C'est demander à un programme d'aller contre la logique de son propre code pour produire des effets que les programmeur·ices n'avaient pas prévus.

Il y a des pensées qui datent l'apparition du glitch comme protocole artistique avant les technologies numériques contemporaines (ça a été dit à propos des dada, ou même de la valorisation artistique des « erreurs » dans les sculptures antiques ou la peinture du XIXe). En ce qui nous concerne, on le connaît et on l'entend surtout comme une manière d'envisager les machines, plus précisément

les machines numériques (qui procèdent des données encodées en binaire). Cela dit, le glitch audiovisuel a une histoire analogique, qui commence quelque part dans le XXème siècle sur le plan visuel (avec des artistes comme Nam Jun Paik, des projets comme le *Digital TV Dinner* ou n'importe quel enfant curieux.se qui approche un aimant d'une télévision cathodique), mais surtout sonore (l'histoire de la noise est très liée à l'idée de détourner l'usage originel d'instruments ou de programmes).

Côté digital, le glitch est souvent défini autour de la notion de *bug* ou d'erreur. C'est vrai d'une certaine manière, mais c'est surtout l'occasion de questionner ce qu'est une erreur machinique. Comme plein d'autres praticien·nes du *glitch art*, on passe une grande partie de notre temps sur des programmes d'encodage ou de traitement image/vidéo et on se demande comment déstabiliser les fichiers. Mais lorsque tu demandes à un script écrit en python de réorganiser les métadonnées d'un fichier vidéo AVI, est-ce que tu génères une erreur ? Quand tu supprimes des morceaux de données d'un fichier vidéo ou que tu l'encodes comme un son, est-ce que tu génères une erreur ? À partir du moment où il est *informatiquement possible* de réaliser ces opérations, c'est que ton code n'est pas absurde. Simplement, tu l'as réécrit ou tu l'as exécuté d'une manière que n'avaient pas prévu les personnes qui l'ont conçu. C'est une erreur relativement à une norme d'usage sociale, technique et esthétique, pas par rapport à la machine en elle-même. Si tu décides de peindre avec un pistolet à eau, ou une brosse à chiotte, est-ce que tu provoques une erreur du pistolet à eau ou de la brosse à chiotte ? C'est une erreur par rapport à une certaine norme sociale, technique et esthétique selon laquelle la peinture doit aller sur un pinceau et les brosses servent à laver les chiottes.

On sait toutes et tous combien cette norme est dépassée dans la peinture contemporaine, le *glitch art* c'est un peu le même geste, mais dans les arts visuels numériques.

Un dernier aspect important de la définition, à nos yeux, c'est la différence entre le glitch non-simulé et le glitch simulé. Le glitch simulé est généré par des programmes, des filtres ou des scripts qui ajoutent sur un fichier des *effets glitch*. C'est très à la mode sur Insta ou Tiktok. Contrairement à ce qui est en jeu dans le glitch d'encodage ou de programme, ces effets font exactement ce pourquoi ils sont prévus (décaler les plages RVB pour citer un exemple connu). Dans le monde (restreint et un peu puriste) du *glitch art*, ces filtres sont parfois dénigrés parce qu'ils ne relèveraient pas du «vrai» glitch. Même si, dans l'anthropie, on les utilise assez peu, parce que leurs résultats sont beaucoup moins surprenants, ils restent très intéressants en termes de codes culturels. Les glitches que tu vois dans *Matrix* ne résultent pas d'«erreurs», ils ont été patiemment dessinés sur des softs d'effets spéciaux, ça ne les empêche pas d'être esthétiquement et politiquement passionnants. Comment expliquer que le glitch simulé, le glitch comme imagerie, soit si populaire dans les montages sur Youtube ou Instagram ? Sans avoir de réponse claire, on le précise par autocritique : définir le glitch comme un détournement du code, c'est passer à côté de 99% de ses usages populaires.

Pourquoi avoir choisi le glitch pour illustrer les travaux de votre collectif ?

On a découvert le *glitch art* il y a plusieurs années, et, pour anticiper sur la suite, on y est entré.es par une porte politique. Certain.es d'entre nous étaient actif.ves dans la sphère hack, au sens large, vers laquelle on s'est tourné.es par intérêt pour les actions directes numériques qu'elle permet (avec plus ou moins d'efficacité, c'est un autre

sujet). En explorant ce que faisaient diverses communautés, comme *AnonOps* ou *Cult of the Dead Cow*, tu te retrouves vite à métaboliser un nouvel univers, avec des codes, des normes et des tentatives formelles. Sans trop savoir par où on a commencé, on s'est intéressé.es à une certaine nébuleuse des arts numériques plus ou moins pirates (ASCII Art, html-art, code-art, etc.) parce que ces communautés la valorisaient. Il y a un lien évident entre hack et curiosité : pour passer des heures à bidouiller du code ou des configurations réseaux, il faut aimer résoudre des problèmes, des casse-têtes plus ou moins mathématiques. Parfois tu as un objectif clair («cracker» quelque chose), parfois, pour être honnête, c'est juste une manière drôle de capter comment une technologie fonctionne. Il y a quelque chose de ludique et de didactique au hack : démonter et rebrancher pour comprendre.

Sur ce plan, le glitch art est très proche des pratiques hack. Il faut d'abord comprendre comment les différents encodages vidéo fonctionnent, comment ils traduisent l'image en données binaires, puis bidouiller à peu près logiquement ces encodages pour générer des résultats inattendus. Au départ, tu n'as pas d'idée claire de ce que tu veux produire, il y a une part de libre-arbitre laissé à la machine. Tu modifies les données jusqu'à trouver le point d'équilibre, presque impossible à théoriser, où le fichier est corrompu mais encore lisible par certains programmes (et tu te débrouilles comme tu peux pour le convertir en une vidéo de qualité, ce qu'en apprenti.es boulanger.ès on appelle «*baker*»).

On a commencé à vraiment s'y mettre quand on préparait la version performée de *Dio*, pour une raison idéologique et pour une raison pratique. Idéologiquement, le texte de *Dio* aborde beaucoup le rôle organique de la technologie, à la fois comme extension (la pensée cyborg qui s'intéresse aux prolongements

techniques des corps humains) mais aussi comme métaphore du corps social (la société est composée d'organes reliés entre eux par des appareillages technologiques). *Dio* questionne le rôle politique des images, des médias et des langages informatiques, alors on voulait l'axer visuellement sur des images très standardisées (presque exclusivement du *stock footage* de visages, de paysages, de modélisations 3D), mais qui dysfonctionnent. C'est un effet idéologique du glitch : dans un monde où le perfectionnement technique cherche à rendre l'image toujours plus immersive, toujours plus transparente, le glitch fait soudainement réapparaître le medium (on développe cette idée à la réponse suivante).

Pratiquement, on avait près de 50 minutes de texte à monter sur de la vidéo, en quelques mois. C'est un autre aspect intéressant du *glitch art* : on peut automatiser beaucoup de protocoles et tayloriser l'abstraction. Laisser faire la machine, ça peut vouloir dire télécharger *Avatar* en 4K, transcoder le fichier dans un format absurde (par exemple un encodage de communication satellite), mélanger aléatoirement les *frames* avec un script, forcer VLC à lire le résultat et obtenir, en vingt bonnes minutes de travail, un film d'art abstrait qui dure plus de deux heures. On exagère, parce qu'il faut choisir les plans, coder des dysfonctionnements qui conservent la lisibilité du fichier, monter, étalonner, et bien sûr tout ça prend en réalité un temps fou, mais l'idée générale nous plaît. Il y a quelque chose d'à la fois humble et bordélique à laisser une certaine liberté créatrice à la machine. Et d'ailleurs on blague sur le taylorisme, mais dans le glitch, contrairement à ce qu'il se passe dans une usine, automatisation ne signifie pas reproductibilité, c'est même justement le contraire. La plupart des protocoles de glitch (non-simulé) génèrent des résultats strictement uniques, impossibles à reproduire. Prendre un fichier vidéo, le transcoder en onde sonore, puis le retranscoder en vidéo produira, à

chaque itération, une vidéo *glitchée* différente. Le glitch est une usine très efficace, mais dont la chaîne de production relève d'un système chaotique où les mêmes causes ne produisent jamais les mêmes effets.

Sur le site de la maison d'édition Abrüpt vous êtes qualifiés de «collectif d'écriture audiovisuelle», comment abordez-vous la création sonore et graphique?

On a pas trouvé de meilleure formule pour décrire ce qu'on faisait. Comment on appelle un collectif qui fait de la littérature de cette manière-là ? Peut-être qu'on ne l'appelle pas, que c'est pas pour rien qu'on ne trouve pas le terme exact et qu'on est voués à disparaître parce que ce qu'on fait n'a aucun sens. La création graphique, on l'a déjà évoquée, mais on pourrait ajouter qu'on trouve intéressant de travailler essentiellement sur des images qui existent déjà. C'est un autre aspect du glitch : aucun protocole ne peut inventer de la donnée. Tous les protocoles de notre connaissance ont besoin d'un fichier préexistant pour fonctionner. Derrière le glitch, il y a toujours l'idée d'une altération, et c'est aussi sa puissance critique : digérer les images qui existent déjà.

Pour la création sonore, on a essayé plusieurs choses, en travaillant souvent avec des samples. Ce qu'on peut dire d'intéressant, c'est que la quasi-totalité des protocoles de glitch vidéo, appliqués à des fichiers .mp4, génèrent aussi des glitches sonores. La plupart des encodages vidéo contemporains considèrent l'image et le son comme un seul flux de données, et il est presque impossible de faire dysfonctionner l'un sans affecter l'autre – ce qui est plutôt une bonne nouvelle pour nous. Pour reprendre l'exemple précédent, notre *Avatar 2.0*, converti en mp3, c'est aussi un album noise abstrait de 2h. Il nous est arrivé plusieurs fois de peaufiner le glitch d'une vidéo, puis d'enfiler des écouteurs et de réaliser que nos protocoles avaient

engendré un résultat sonore intéressant, puis de le découper, le sampler et l'utiliser pour autre chose.

Trouvez vous une dimension politique au glitch art ?

Vaste question. Est-ce que les formes portent en elles-mêmes des idéologies ? Pas sûr. Cela dit, la principale dimension politique du glitch, à nos yeux, c'est la fêlure de l'écran. On peut développer ce qu'on disait à propos de Dio. Un *medium*, comme une image ou une vidéo, c'est un artefact enté sur un processus : disparaître pour créer l'illusion de l'immédiateté. Cette immédiateté est factice, parce qu'elle repose sur un pliage de l'espace-temps – et le glitch est un révélateur efficace de cette facticité. Quand tu passes un coup de téléphone, la technologie plie l'espace : la personne avec qui tu parles est *immédiatement* présente à ta subjectivité. Tu peux complètement oublier que vous êtes sujets d'une médiation et passer trois heures à lui parler comme si elle était avec toi. C'est tout à fait pareil avec une vidéo, qui plie aussi l'espace pour te faire voir une plage des Maldives *comme* si tu y étais. Forcément, on améliore la technologie téléphonique pour que le son des voix soit toujours plus net, toujours plus agréable. La courbe de l'évolution technologique est presque toujours orientée vers une augmentation de l'immédiateté : plus un *medium* s'améliore, plus il cherche à disparaître efficacement.

Mais s'il commence à y avoir du bruit dans la communication téléphonique, ou que ta connexion lâche, alors d'un coup le médium réapparaît. Si la plage des Maldives est mal encodée, tu te rappelles brutalement que tu es en train de regarder un écran dans ta chambre. C'est une puissance critique du glitch : briser l'envoûtement.

Alors, dis comme ça, c'est encore trop réactionnaire pour nous. Bien sûr que certaines images, et certaines

pratiques de l'écran en général, jouent une partition aliénante, au service de dominations identifiées. Mais utiliser simplement le glitch comme un pur outil technocritique, c'est réducteur. Si tu veux juste dénoncer le rôle aliénant des appareils, autant débrancher la prise ou filmer le public en récitant du Marx. Ce qui nous intéresse, c'est de chercher des dysfonctions capables de susciter des émotions esthétiques et politiques, de travailler patiemment la plage des Maldives pour la faire glitcher de manière spectaculaire, abstraite, surprenante, fictionnelle. La plage des Maldives glitchée va produire un nouvel envoûtement, le même que n'importe quelle peinture abstraite, mais un envoûtement conscient du *medium*, un *envoûtement critique*. La relation entre *medium* et immédiateté est transfigurée : le *medium* n'est plus invisible, on le révèle volontairement, on le transforme malgré lui en une toile où se dessine une abstraction envoûtante et il n'opère plus une médiation de l'espace-temps. Lui qui cherchait à s'effacer est devenu *ce qui est immédiatement présent*, comme n'importe quelle toile dans un musée. C'est d'ailleurs quelque chose qui nous a particulièrement stimulé dans les arts vivants : pendant la performance de Dio, le public était collectivement co-présent.es avec une performeuse qui portait le texte, mais aussi avec un écran devenu toile sur lequel défilaient 45 minutes d'abstraction.

Pour vous c'est quoi le glitch de demain ? Comment se distingue-t-il du glitch d'hier et d'aujourd'hui ?

Difficile à prévoir, mais le métavers de Zuck a l'air de bien glitcher. Ce qui est sûr, c'est que le glitch suivra une trajectoire parallèle à l'histoire de la technologie future. Aucune technologie n'est à l'abri d'un bug, c'est ontologique à tout appareil. Il y aura toujours des curieux.ses pour se demander comment forcer une nouvelle invention à aller contre son usage programmatique. Vu le vecteur



de l'évolution, obsédé par l'immersion, un certain totalitarisme sensoriel (la VR, la 3D, etc.), l'avenir du glitch sera probablement lié à des dysfonctions proprioceptives. Explorer les enjeux esthétiques et politiques du bug en réalité virtuelle, c'est l'une des tâches de la prochaine génération de glitch kids.

Ça c'est pour l'optimisme, mais il y a aussi surtout des sources d'inquiétudes, notamment la financiarisation et la spécialisation du numérique. Tout le monde sait que la technologie numérique est au XXIème siècle ce que l'industrie métallurgique était au XXème siècle : la source de gigantesques conglomérats capitalistes qui se partagent le continent internet comme autrefois les continents physiques. Cette évolution implique que les codes, les fichiers et les programmes sont toujours plus fermés, sécurisés et propriétaires. Le glitch s'épanouit dans un monde ouvert, où la lecture est libre et où l'information est à disposition de tous. Pour faire dysfonctionner un fichier avec complexité et douceur, il faut avoir un accès entier à ses entrailles. Plus le fichier sera propriétaire et secret, plus il faudra le casser brutalement, ce qui n'a rien d'absurde, mais il serait dommage que ça devienne la norme. À cela il faut ajouter l'évolution des hégémoniques smartphones : des ordinateurs fermés, standardisés et où l'on ne peut même plus coder (sans les avoir « crackés » au préalable, ce qui est lourd de signification). Et y ajouter aussi la spécialisation des tâches dans le numérique, inhérente à la complexification, qui oblige à avoir des connaissances toujours plus poussées pour comprendre un champ technique toujours plus restreint. D'ailleurs, beaucoup des protocoles qu'on utilise reposent sur de vieux logiciels, qui datent de l'époque où les données étaient bordéliques, constellées de trous et de vides juridiques. Il n'est pas rare qu'on doive remonter dans les archives pour télécharger des versions datant des années 2000. Audacity est un bon exemple : c'est un vieux logiciel de traitement de son, que

plus personne n'utilise, mais, comme beaucoup de logiciels de son temps, son interpréteur n'est pas verrouillé. Il peut transcoder absolument n'importe quel fichier en une onde sonore, ce qui n'a aucun sens dans la plupart des cas, mais il le peut. C'est une vieille manière de penser les programmes : si c'est *techniquement possible* de transcoder n'importe quoi en son, alors on laisse cette fonctionnalité, elle servira peut-être un jour à quelqu'un. Cet état d'esprit a quasiment disparu : la plupart des logiciels contemporains sont verrouillés, standardisés, pour éviter le plus possible que les utilisateur.ices fassent des erreurs, des *glitches*.

Récemment, on a mis à un jour l'un de nos macs, et il n'a plus supporté un logiciel de 2006 qu'on utilise beaucoup pour les visuels d'un projet de spectacle. Pour pouvoir continuer à faire du théâtre *contemporain*, il nous a fallu ouvrir le corps de l'ordinateur et *reboot* la machine dans une version antérieure. On sait pas lire dans les entrailles, mais c'est sans doute de mauvais augure.