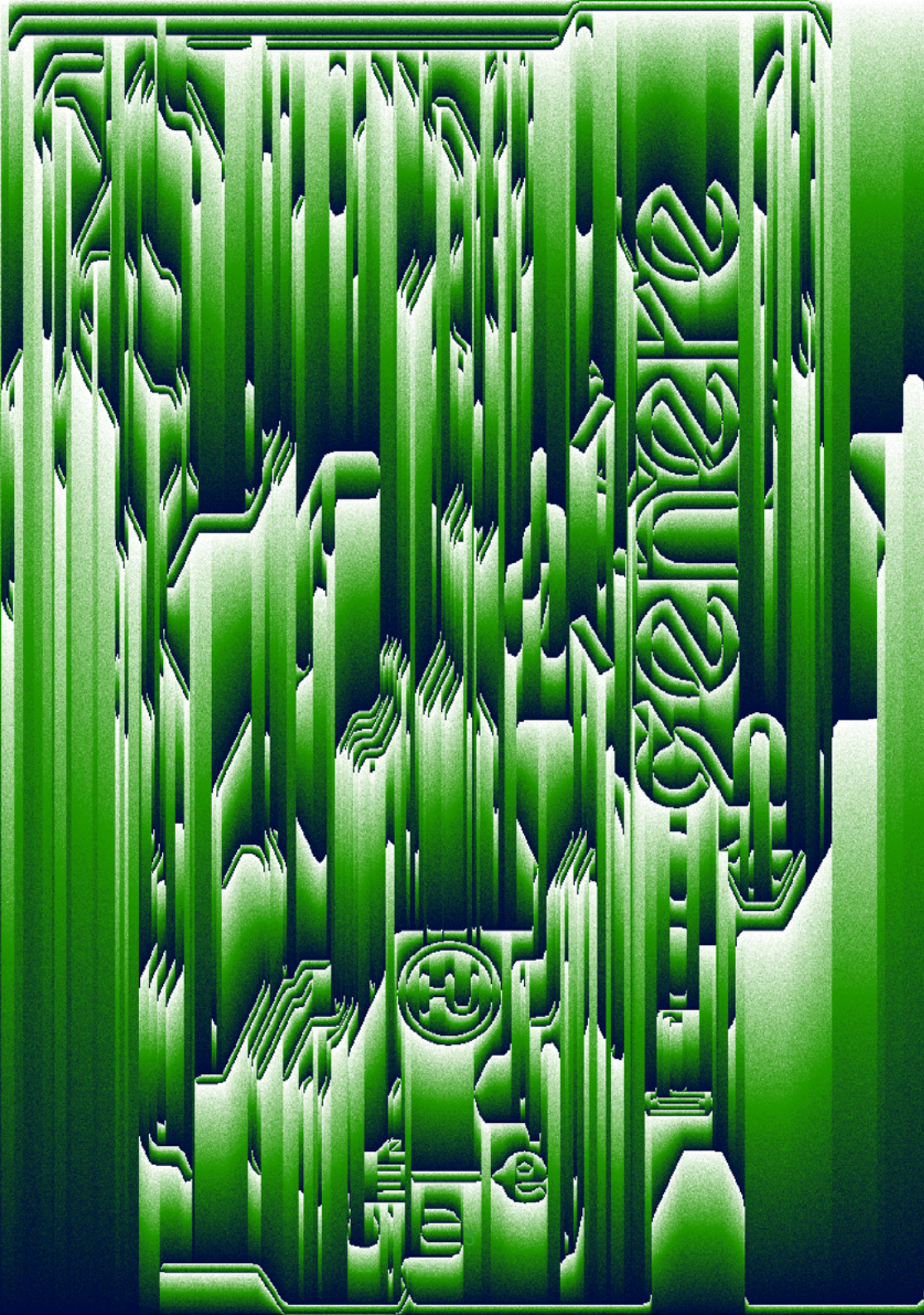


Comment définir la frontière entre génération et création ?

---gènère

gènère



Ce mémoire n'aurait pas pu être exister sans l'inventeur du papier, ni sans Alan Turing, ni sans mes parents.

Un grand merci à eux.

Comment définir la frontière entre génération et création dans le design graphique?

Le fait de générer semble être au cœur des réflexions sur l'avenir de la pratique du designer. Issu du latin *generare*, le verbe *générer* signifie produire, engendrer – souvent de manière automatique, naturelle ou systématique.

Par opposition, le verbe *créer*, issu du latin *creare*, désigne l'acte de faire exister quelque chose qui n'était pas, de donner naissance à une œuvre ou un objet par un effort de l'esprit, impliquant une intention, une subjectivité et une originalité. Créer est l'acte maîtrisé, subjectif, singulier, souvent perçu comme la quintessence du rôle de l'artiste ou du designer.

La tension entre création et génération se situe précisément dans la remise en question de cette singularité et de cette maîtrise.

Allons-nous vers un effacement du designer ?
Est-il celui qui crée l'algorithme ou le système de règles, ou celui qui crée la forme finale ?

Ce mémoire propose une exploration de pratiques graphiques intégrant des processus de génération, où la part de création, présente pour chacune d'elles, interroge cette frontière floue.

	---	introduction
	---	partie 1
		prompter : utiliser l'outil génératif pour créer

		1.1. Une nouvelle façon de créer

		1.2. Boîte noire & Déplacement de la création

		1.3. Biais éthiques et Juridiques
	---	partie 2
		paramétrer : configurer l'outil génératif pour créer

		2.1. Le design génératif

		2.2. L'itération au service de l'exploration

		2.3. La barrière du paramétrique et le risque du trop algorithmique
	---	partie 3
		coder : construire l'outil génératif pour créer

		3.1. Le geste de génération maîtrisée : construire l'outil de son intention

		3.2. Liberté créative et maîtrise technique

		3.3. Libre, bazar & Communs
	---	conclusion
	---	annexe

Prompter : utiliser l'outil génératif pour créer

1.1. Une nouvelle façon de créer

Pour étudier l'utilisation des outils génératifs en design graphique comme nouvelle façon de créer, j'ai interrogé l'IA générative textuelle Le Chat de Mistral AI¹ :

« L'avènement des Intelligences Artificielles génératives ces dernières années a posé, dans la pratique du graphiste, de nombreuses questions et critiques, tant elles opèrent une transformation radicale des étapes de conception.

Une fois pris en main, l'outil permet au designer d'explorer un champ de possibilités visuelles à une vitesse sans précédent, générant rapidement des prototypes ou des pistes esthétiques qui auraient requis un temps de production manuelle considérable. »

¹ Utilisation de l'IA textuelle Le Chat de Mistral AI (modèle Mixtral 8x22B) le 14.10.2025, avec le prompt suivant : « Explique-moi brièvement sur un ton formel et académique, l'impact des Intelligences Artificielles génératives sur la pratique du designer graphique ».

1.2. Boîte noire & Déplacement de la création

Mais au delà d'avoir réponse à tout, l'IA générative se caractérise aussi par son opacité.

Les mécanismes complexes et fondateurs qui les régissent – compilation de grands volumes de données, espace latent², etc... – appuient l'effet boîte noire³. Le designer se retrouve dans une posture nouvelle, celle d'un pilote de système, un *prompt engineer*⁴, un curateur d'options, fournissant une requête textuelle (*input*) sans accès direct ni compréhension intime de la manière dont l'algorithme produit le visuel final (*output*).

Le travail de création se déplace alors de l'exécution à la formulation de l'intention: le prompt⁵, et à la sélection critique des résultats générés.

² L'espace Latent est un espace vectoriel abstrait qui constitue la cartographie sémantique des données d'apprentissage. cf. Glossaire.

³ La boîte noire (en anglais *black box*) est une métaphore spécifique apparue dans le monde de la recherche pour décrire le manque de transparence et l'opacité du fonctionnement interne des modèles d'IA complexes.

Pour une analyse approfondie de cette menace pour le design: Masure, Anthony, « Résister aux boîtes noires: le design face aux intelligences artificielles », publié en décembre 2019.

⁴ L'ingénierie de prompt (de l'anglais *prompt engineering*), est le processus de conception et de structuration d'un prompt, afin qu'il soit efficacement interprété et compris par un modèle d'intelligence artificielle générative et suive l'intention de son auteur.

⁵ Un prompt (en français *requête* ou *invite*) est une description textuelle de la tâche que l'IA doit effectuer.

Le projet « The Next Rembrandt » de VML⁶ est un cas d'étude marquant, où l'algorithme a généré une nouvelle œuvre après analyse des centaines d'œuvres de l'artiste, posant la question de la présence indirecte de ce dernier dans cette synthèse artistique nouvelle, réalisée sans intervention manuelle directe d'un humain, de la génération à la "peinture".



⁶ VML, « The Next Rembrandt », 2016. Ce projet, né d'une collaboration entre l'agence VML et Microsoft, a mobilisé des data-analystes pour traiter des milliers de fragments de l'œuvre de Rembrandt via un algorithme d'apprentissage profond. En identifiant les constantes statistiques de l'artiste, le système a généré une synthèse type, qu'il a ensuite matérialisée par impression 3D en simulant jusqu'à la texture de la peinture à l'huile.

1.3. Biais éthiques et Juridiques

Les questionnements éthiques et juridiques que les modèles d'IA génératives soulèvent résident dans leur construction même. Ils nécessitent l'absorption massive de corpus d'images et d'œuvres existantes – réalisée sans permission explicite des créateurs. Ce *web scraping*⁷ non autorisé d'œuvres protégées pose un risque juridique majeur et remet en question la propriété intellectuelle.

Éthiquement, l'IA reproduit et amplifie les biais algorithmiques présents dans ses données d'entraînement. Sur le plan philosophique, cette captation de la production créative s'inscrit dans un cadre plus large de capitalisme de surveillance⁸ où les données deviennent marchandises, tel que conceptualisé par Shoshana Zuboff⁹.

⁷ Le *web scraping*, ou *moissonnage* en français, est la principale technique de récupération et organisation automatisées des données Web. Elle vise à capter des données, pour les transformer et/ou les réutiliser dans un autre contexte comme dans notre cas l'enrichissement de bases de données pour l'apprentissage profond (*deep learning* cf. Glossaire) pour une intelligence artificielle.

⁸ Zuboff, Shoshana. *L'Âge du capitalisme de surveillance*, éd. trad. (2020, 1ère éd. 2019) décrivant le modèle économique fondé sur l'extraction et l'exploitation des données comportementales humaines (y compris créatives) pour créer des produits de prédiction. Il fournit le cadre éthique pour critiquer les enjeux de la capture des interactions avec les outils d'IA propriétaires.

⁹ Shoshana Zuboff est universitaire, professeure à la Harvard Business School, psycho-sociologue et femme de lettres américaine. Ses travaux portent principalement sur l'économie du numérique.

L'encadrement de ces pratiques est devenu une urgence législative, comme en témoigne le Règlement (UE) 2024/1689 du Parlement européen, connu sous le nom d'« AI Act ¹⁰ », qui cherche à définir les règles de transparence et d'utilisation de l'Intelligence Artificielle. ---gènère

Ces enjeux rappellent que la délégation de tâches aux IA génératives rend le designer complice d'un processus dont les fondations sont contestables.

Face à un système propriétaire dont le fonctionnement et les implications restent hors de portée, une autre voie s'impose : celle de la reprise de maîtrise en cherchant à concevoir le système plutôt qu'à s'y soumettre.

¹⁰ Parlement Européen, Règlement (UE) 2024/1689 du Parlement européen et du Conseil du 13 juin 2024 dit « AI Act », Bruxelles, 2024.

---partie 2

Paramétrer : configurer l'outil génératif pour créer

2.1 Le design génératif

Plutôt que de subir un outil fermé, opaque, et biaisé, le design génératif permet de retrouver la maîtrise par la création de systèmes, en utilisant des logiciels de modélisation paramétrique (TouchDesigner ou Grasshopper)¹¹ ou des environnements et bibliothèques de code (Processing ou p5.js)¹² pour collaborer avec l'algorithme.

Ce mouvement prend une nouvelle urgence face aux systèmes propriétaires. Il s'inscrit dans le sillage de John Maeda qui, dès les années 90, développe *Design by Numbers*¹³, méthode d'apprentissage pour rendre la programmation accessible aux designers. Ses étudiants Benjamin Fry et Casey Reas, prolonge sa démarche en créant Processing, ouvrant la voie à des designers faisant de la configuration de l'outil un geste créatif à part entière.

¹¹ cf. Glossaire pour en savoir plus sur les logiciels de modélisation paramétrique Touch Designer et Grasshopper

¹² cf. Glossaire pour en savoir plus sur les environnements et bibliothèques de code Processing et p5.js.

¹³ Maeda, John. *Design by Numbers*, MIT Press, 1999. Artiste, graphiste et chercheur, il a jeté les bases pédagogiques du *Creative Coding* en développant une méthode et un langage simplifiés afin de rendre le code accessible aux artistes et designers.

L'intention créative se déplace: on ne dessine plus une forme, mais on conserve la maîtrise en définissant les paramètres permettant à une forme inattendue d'émerger, en s'aidant de l'aléatoire, de données internes ou externes, ou encore d'interactions humaines captées.

La création devient la définition des règles du généré.

L'identité visuelle de la Casa da Música à Porto¹⁴ conçue par Stefan Sagmeister illustre cette approche: le logotype est un système algorithmique se déformant et se colorant selon des variables définies. La création réside dans l'architecture du système engendrant ces variations, assurant reconnaissance et nouveauté constantes.

La configuration du système est, en soi, l'œuvre.



¹⁴ Sagmeister, Stefan. Casa da Música, Porto, 2005. L'identité visuelle, développée avec le studio Sagmeister & Walsh, est célèbre pour son usage pionnier du design génératif. L'enjeu était de créer une identité dynamique qui reflète la diversité et la complexité des événements musicaux.

---Comment définir la frontière entre génération et création ?

2.2 L'itération¹⁵ au service de l'exploration

Le paramétrage ouvre un champ d'exploration formelle infini, où l'itération devient la méthode pour découvrir des solutions que le designer n'aurait pas pu concevoir manuellement. Cette démultiplication des possibles permet d'atteindre l'unicité par la variation¹⁶.

Le projet « See before reading » de Susanne Stahl et André Gottschalk¹⁷ exploite cette puissance itérative. Composée de trois niveaux de lecture visuelle (genre, atmosphère, structure narrative), la couverture est générée de manière systémique pour refléter le contenu du livre. Le designer crée ici un outil d'aide à la décision pour l'utilisateur, où la génération de la forme est directement liée à l'information et à l'usage.



¹⁵ L'itération est le processus qui consiste à répéter plusieurs fois un ensemble d'instructions ou d'étapes. Elle permet d'effectuer plusieurs fois la même tâche avec de légères variations ou de modifier un morceau de code jusqu'à ce qu'une condition spécifique soit remplie.

¹⁶ Cette approche s'inscrit dans l'héritage des démarches artistiques qui, avant l'ère numérique, définissait déjà des protocoles stricts pour générer des formes variées, comme les travaux sur l'ordre et le chaos dans l'art systémique.

Des figures comme Josef Albers avec ses séries d'études sur la couleur « Homage to the Square » ou Max Bill et son approche de l'art concret, concevaient déjà la création comme l'application de règles pour explorer les variations formelles.

¹⁷ Stahl, Susanne et Gottschalk, André. See before reading, projet de recherche et concept de design génératif appliqué aux couvertures de livres, 2017. L'intention du dispositif est de transformer la couverture en une interface informative et esthétique, générée selon des paramètres liés au genre, à l'atmosphère et à la structure de l'ouvrage.

Dans le champ de la typographie, les *Variable Fonts*¹⁸, dont les designers typographes configurent précisément les axes de variations, permettent d'itérer de manière fluide entre des milliers de variations intermédiaires. La création réside ici dans la définition des axes autorisant un rendu expressif et adaptatif.

L'itération démontre que le designer ne délègue pas la création à l'outil génératif, mais qu'il codifie la génération pour en faire un moteur d'innovation heuristique par la répétition.

2.3 La barrière du paramétrique et le risque du trop algorithmique

Si la configuration paramétrique confère une maîtrise conceptuelle, elle engendre une nouvelle difficulté que l'on pourrait nommer la barrière du paramétrique. En effet, l'utilisation de logiciels paramétriques exige une expertise technique significative qui représente une vraie barrière à l'apprentissage.

La construction de systèmes par programmation, même aussi visuelle et didactique qu'elle puisse être, introduit une complexité d'architecture qui devient un vrai frein pour le graphiste, déplaçant la création vers la quasi-ingénierie.

¹⁸ Les polices à axes variables (*Variable Fonts* en anglais) sont une technologie introduite en 2016 permettant de définir plusieurs axes de variation (graisse, largeur, inclinaison, etc.) dans un seul fichier de police. L'utilisateur peut interpoler entre ces axes de manière fluide, générant potentiellement des milliers de variations intermédiaires à partir des paramètres définis par le typographe. Des logiciels comme Glyph, ou sa version libre Birdfont, permettent de créer des Variable Fonts.

Un autre risque majeur est la perte d'impact sémantique lorsque le procédé de génération devient trop lisible. Le travail de Refik Anadol¹⁹ dans « Unsupervised » illustre cette tension. Son œuvre, bien que sophistiquée, peut être taxée de trop algorithmique : la fluidité hypnotique des formes masque le fond, l'œuvre est comprise comme générée, ce qui limite la projection sémantique du spectateur, et le résultat est réduit à un rendu qui ne célèbre que sa propre fabrication (autoréférence formelle), la prouesse technique occultant l'intention subjective du designer.

L'enjeu final pour lui est donc de configurer l'outil avec une distance critique : le protocole technique doit rester au service de la création, et non l'inverse.



¹⁹ Refik Anadol. Artiste médiatique turco-américain spécialisé dans l'art algorithmique. Pour son œuvre *Unsupervised* (MoMA, 2022-2023) il a recueilli toutes les œuvres du musée dans une base de données qui a servi à entraîner un modèle d'IA générative sur mesure (*StyleGAN2*). Ce processus lui a permis de générer une cartographie de l'espace latent, explorée via un logiciel personnalisé, le *Latent Space Browser*, puis exposée au public sur des écrans monumentaux.

Coder : construire l'outil génératif pour créer

3.1 Le geste de génération maîtrisée : construire l'outil de son intention

Le codage créatif²⁰ ne consiste pas seulement à manipuler un système, mais à le construire en comprenant ses rouages, garantissant que chaque ligne de code est au service d'une intention artistique ou conceptuelle unique.

L'outil généré devient le prolongement direct du geste conceptuel du designer, qui s'affranchit également des logiques marchandes ou des contraintes logicielles des plateformes propriétaires.

L'exemple de l'artiste pionnière Vera Molnar²¹ est fondateur. À la fin des années 60, elle a utilisé l'ordinateur pour explorer la relation entre ordre et désordre, régulier et aléatoire. En écrivant ses propres programmes pour générer des séries d'images (par exemple, la déformation progressive et aléatoire d'un carré), Molnar a démontré que la création réside dans le choix initial du système de règles et dans la définition des limites de l'aléa. Elle ne paramétrait pas un logiciel existant, elle codait son intention, prouvant que l'ordinateur peut être plus qu'un simple outil de rendu : un collaborateur créatif.

²⁰ Le codage créatif est une discipline de la programmation où le code est utilisé avec comme intention de créer quelque chose d'expressif et esthétique, et qui n'a pas à être fonctionnel. Il peut s'agir de générer des œuvres artistiques, des visuels interactifs, de la vidéo ou des performances sonores.

²¹ Molnar, Vera (1924-2023). Artiste franco-hongroise, considérée comme l'une des pionnières de l'art informatique et de l'art algorithmique. Son travail avec un ordinateur, débuté dès 1968, explore l'esthétique de la génération par des règles (souvent appelées « machines imaginaires »), plaçant le processus et le système au centre de sa démarche créative.

---Mémoire de DN MADŕ Graphisme – École Estienne

3.2 Liberté créative et maîtrise technique

Le codage créatif impose une exigence technique forte, mais l'apprentissage du code, bien qu'ardu, est la condition d'une liberté créative illimitée.

En maîtrisant la syntaxe et la logique de programmation, le designer graphique peut non seulement générer des formes statiques, mais aussi gérer des flux de données complexes, intégrer des interactions en temps réel et inventer des dispositifs visuels dynamiques qui dépassent les conventions graphiques établies. Coder pour créer c'est aussi être autonome, ne plus dépendre des éditeurs de logiciels propriétaires comme Adobe et ses outils verrouillés²².

La frontière entre génération et création se clarifie. La génération est le mécanisme algorithmique, mais la création est l'intention subjective qui décide non seulement du résultat, mais aussi de la manière dont ce mécanisme est construit.

Le designer devient l'auteur complet de son médium, injectant du sens là où la génération ne produirait que de la forme. Le code devient ainsi une matière première malléable permettant une expression illimitée.

²² Depuis le passage de l'achat à vie à l'abonnement obligatoire en 2013, avec le transfert forcé au *Creative Cloud*, le monopolistique éditeur Adobe est vivement critiqué. Accusé de vouloir verrouiller les créatifs, les forçant à stocker leur travail dans un cloud dont l'accès est une location perpétuelle; de ne plus proposer de réelles nouveautés depuis des années sur certains logiciels comme InDesign; ou encore récemment d'entraîner des IA avec les travaux des utilisateurs sans leur consentement clair. cf. les travaux de Cory Doctorow pour la Electronic Frontier Foundation depuis 2013 et son concept d'« *enshittification* » (2023) pour désigner la dégradation délibérée des plateformes au profit de la capture d'une rente.

3.3 Libre, bazar & Communs

Le fait de cr  er son propre outil est intrins  quement li      l'id  ologie de l'Open Source²³, offrant une r  ponse directe aux probl  matiques   thiques et juridiques des syst  mes propri  taires opaques (1.3).

S'opposant au mod  le de d  veloppement commercial, vertical et ferm  , d  sign   par Eric S. Raymond comme le mod  le de la Cath  drale, il adopte le mod  le du Bazar. Ce dernier, chaotique mais efficace, repose sur la d  centralisation de la contribution, la transparence du code et la correction collective par la communaut  .

Cette philosophie garantit l'autonomie du designer et s'oppose au capitalisme de surveillance en assurant que les technologies de g  n ration ne sont pas captur  es par une entit   unique. L'utilisation de licences ouvertes (comme la GNU GPL²⁴) ou le partage de son code sur GitHub ou GitLab²⁵ n'est pas un simple choix technique, mais un geste politique faisant du code un territoire d'  mancipation et d'innovation collective. En contribuant    ces   cosyst  mes, le designer participe activement    la construction d'une bo  te    outils   thique, partag  e et adapt  e    sa pratique.

²³ L'Open Source d  signe un mode de d  veloppement logiciel o   le code source est librement accessible, modifiable et distribu  ble. Cette philosophie, formalis  e dans les ann  es 1990, s'oppose au mod  le propri  taire en favorisant la transparence, la collaboration et l'  mancipation des utilisateurs. C'est la philosophie aujourd'hui port  e par le syst  me d'exploitation Linux, ou les logiciels de cr  ation graphique GIMP & Krita.

²⁴ La licence GNU GPL est une licence de logiciel libre qui garantit aux utilisateurs le droit d'ex  cuter, d'  tudier, de modifier et de partager le code source. cf. Glossaire

²⁵ GitHub et GitLab sont deux plateformes de d  veloppement logiciel qui offrent des services bas  s sur Git, un syst  me de contr  le de version d  centralis  . Ces plateformes permettent aux d  veloppeurs de stocker leur code, de suivre les modifications et, surtout, de collaborer sur des projets.

---conclusion

Compte rendu d'exploration

L'exploration des pratiques graphiques intégrant la génération, de l'Intelligence Artificielle générative au codage créatif, révèle que la frontière entre génération et création n'est pas absolue, mais se définit par le degré de maîtrise et l'intention critique du designer.

Ce mémoire a cartographié cette évolution en trois étapes : celle de l'opacité, où la création est déléguée aux systèmes propriétaires et soumise au risque éthique ; celle de la maîtrise des règles, où le designer devient architecte mais doit contourner l'autoréférence formelle ; et enfin, celle de l'autonomie radicale, où le codage créatif lui permet de bâtir son propre médium.

En définitive, l'importance de la génération est inversement proportionnelle à la connaissance que le graphiste a du système. La création n'est pas l'absence de génération, mais l'acte subjectif qui en choisit les règles, les limites et la finalité. Loin de s'effacer, le designer est invité à gagner en compétence, passant d'un statut d'utilisateur passif à celui d'un "artisan" maître de son propre outil.

---merci de m'avoir lu !

Bibliographie

---ouvrages et textes théoriques

Zuboff, Shoshana, L'Âge du capitalisme de surveillance, Paris, Albin Michel, 2020 (1ère éd. 2019), 704 p.

Shiffman, Daniel, The Nature of Code: Simulating Natural Systems with Processing, San Francisco, The Nature of Code LLC, 2012, 520 p.

Reas, Casey & Fry, Ben, Processing: A Programming Handbook for Visual Designers and Artists, Cambridge (Massachusetts), MIT Press, 2007, 736 p.

Maeda, John, Design By Numbers, Cambridge (Massachusetts), MIT Press, 1999, 256 p.

Raymond, Eric S., La Cathédrale et le Bazar, traduction française par Sébastien Blondeel, 1998 (1ère éd. 1997), Disponible sur : <http://www.catb.org/~esr/writings/cathedral-bazaar/> [consultation le 15.09.25]

---articles, articles de lois & ressources en ligne

Evrard, Olivier, « Design Génératif », Disponible sur : <https://www.olivierevrard.be/master/design-generatif.html> [consultation le 15.10.25]

Masure, Anthony, « Résister aux boîtes noires : le design face aux intelligences artificielles », publié en décembre 2019. Disponible sur : <https://www.anthonymasure.com/articles/2019-12-resister-boites-noires-design-intelligences-artificielles> [consultation le 12.11.25]

Blanc, Julie, et Maudet, Nolwenn, « Code - Design graphique. Dix ans de relations », Dans : Graphisme en France, Création, outils, recherche, n° 28, CNAP, 2022, p. 3-32. Disponible sur : https://www.cnap.fr/sites/default/files/n%C2%B028%20-%20Graphisme%20en%20France%20-%20Cr%C3%A9ation%20%20outils%20%20recherche%20%202022_compressed.pdf [consultation le 10.10.25]

Hübner, Patrik, « Generative Design », Dans : Data-Design Dictionary. Disponible sur : <https://www.patrik-huebner.com/datadesigndictionary/generative-design/> [consultation le 19.10.25]

Parlement Européen, Règlement (UE) 2024/1689 du Parlement européen et du Conseil du 13 juin 2024 dit « AI Act », Bruxelles, 2024, Disponible sur : <https://artificialintelligenceact.eu> [consultation le 14.10.25]

GitHub, Open Source Guides, Disponible sur : <https://opensource.guide/> [consultation le 04.10.25]

---projets et études de cas

VML, « The Next Rembrandt », 2016, Disponible sur : <https://www.vml.com/work/next-rembrandt>
[consultation le 14.10.25]



Sagmeister & Walsh, Identité visuelle Casa da Música, 2007, Disponible sur : <https://sagmeister.com/work/casa-da-musica/> [consultation le 10.10.2025]



---la génératives, design génératif & codage créatif

Stahl, Susanne et Gottschalk, André. See before reading,
 , 2017, Disponible sur : <https://blog.clementbuee.fr/2014/11/28/see-before-reading-generateur-de-couvertures/> [consultation le 28.10.2025]



Anadol, Refik, Unsupervised – Machine Hallucinations, MoMA,
 (19 nov. 2022 – 29 oct.2023), Disponible sur :
<https://refikanadol.com/works/unsupervised/>
 [consultation le 12.10.25]



Glossaire

Algorithme

Suite finie et non ambiguë d'instructions ou de règles permettant de résoudre un problème ou d'obtenir un résultat précis. En art numérique, l'algorithme est la "recette" qui dicte comment les données ou les formes doivent évoluer, se multiplier ou réagir aux interactions de l'utilisateur.

Creative coding (Codage créatif)

Pratique de programmation informatique dont l'objectif premier est la création d'une œuvre expressive ou esthétique plutôt que la simple exécution d'une tâche fonctionnelle. Il s'agit d'utiliser le code comme un pinceau ou un instrument de musique pour produire des visuels génératifs, des installations interactives ou des performances "live".

Deep learning (Apprentissage profond)

Sous-ensemble de l'intelligence artificielle qui utilise des réseaux de neurones artificiels multicouches pour apprendre automatiquement à partir de grandes quantités de données. C'est la technologie derrière la reconnaissance d'image et la génération de contenu complexe. Pour en apprendre plus : <https://developers.google.com/machine-learning/crash-course>

Design génératif

Méthode de conception où le designer n'élabore pas l'objet final lui-même, mais définit un ensemble de règles et de paramètres qui génèrent automatiquement une multitude de formes. C'est une collaboration entre l'humain, qui guide le processus, et la machine, qui explore les possibilités mathématiques.

Espace Latent

Espace vectoriel abstrait qui constitue la cartographie sémantique des données d'apprentissage. Il permet aux réseaux génératifs d'encoder des concepts complexes sous forme de vecteurs. En manipulant ces vecteurs par interpolation, la machine génère des variations infinies : c'est là que l'objet appris est transformé par l'algorithme.

Pour en apprendre plus : <https://www.ibm.com/fr-fr/think/topics/latent-space>

GAN (Réseaux Antagonistes Génératifs)

Architecture d'IA où deux réseaux de neurones s'affrontent : le "générateur" crée des images, tandis que le "discriminateur" tente de déceler si elles sont réelles ou fausses.

Cette compétition permet de produire des images d'un réalisme saisissant.

GNU GPL

Licence de logiciel libre garantissant aux utilisateurs le droit d'exécuter, d'étudier, de modifier et de partager le code source. Son caractère « héréditaire » (copyleft) impose que toute version modifiée reste libre. Pour en apprendre plus :

<https://www.gnu.org/licenses/quick-guide-gplv3.fr.html>

Grasshopper

Outil de programmation visuelle intégré au logiciel Rhino 3D. L'utilisateur connecte des blocs de fonctions pour générer des structures paramétriques et géométriques fluides sans écrire de texte. Pour en apprendre plus : <https://modelab.gitbooks.io/grasshopper-primer/content/>

Intelligence artificielle (IA)

Ensemble de techniques permettant à des machines de simuler l'intelligence humaine. Dans l'art, elle analyse des jeux de données pour générer des contenus originaux en imitant ou transformant les styles appris lors de sa phase d'entraînement.

Intention créative

---gènère

Expression de la vision ou du message de l'artiste. Dans un contexte algorithmique, il s'agit de maintenir la volonté humaine sur la machine en choisissant précisément les paramètres et les limites du système.

p5.js

Bibliothèque JavaScript gratuite et open-source conçue pour rendre le codage créatif accessible sur le web. Elle permet de créer des animations et des interactions complexes directement dans un navigateur. Pour en apprendre plus : <https://p5js.org/>

Processing

Environnement de développement libre fondé sur Java, créé par Benjamin Fry et Casey Reas. Conçu pour les arts visuels, il a joué un rôle majeur dans la démocratisation du codage créatif. Pour en apprendre plus : <https://processing.org/overview>

Prompt (Invite)

Instruction textuelle donnée à une IA générative pour orienter sa production. La qualité du résultat dépend souvent de la précision et de la structure sémantique de ce "prompt".

TouchDesigner

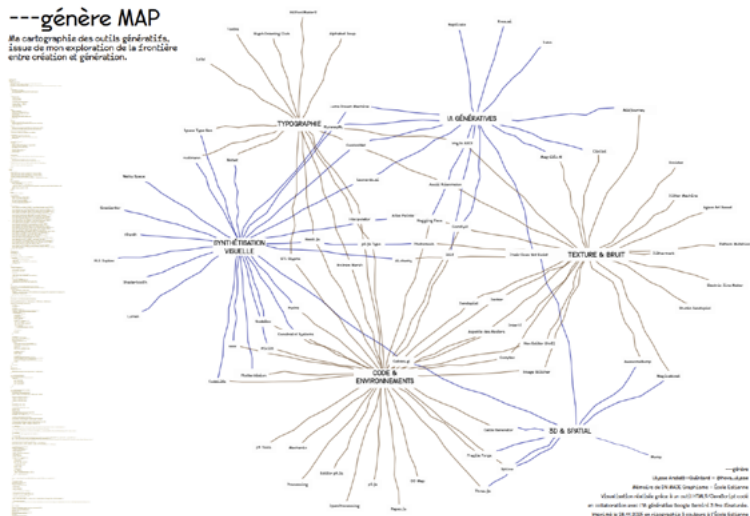
Outil propriétaire de programmation visuelle basé sur des nœuds, spécialisé dans la manipulation de contenu multimédia interactif en temps réel pour le spectacle vivant et les installations. Pour en apprendre plus : <https://derivative.ca/>

---@hova_ulyssz ~~~~ Ulysse Anstett--Quintard

Cartographie des outils génératifs

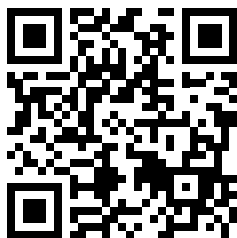
---gènère MAP

Ma cartographie des outils génératifs,
issue de mon exploration de la frontière
entre création et génération.



<https://genere.hovaulysse.com/map>

Ma cartographie des outils génératifs, réalisé à partir de
cette Bibliothèque d'outils issue de mon exploration.
Également imprimée en risographie 3 couleurs.



Bibliothèque d'outils – entre Génération & Création

<http://al.chemy.org/>

Interface expérimentale pour explorer des interactions algorithmiques visuelles et sonores.

<https://alienmelon.itch.io/electric-zine-maker>

Création de fanzines numériques via un outil ludique et interactif.

<https://andrewmarsh.com/software/>

Collection d'outils expérimentaux pour le dessin vectoriel et la génération typographique.

<https://app.spline.design/>

Éditeur 3D en ligne, orienté design interactif et création de scènes génératives.

<https://ascii.alienmelon.com/>

Générateur de visuels ASCII à partir d'images ou de texte.

https://b2renger.github.io/p5js_typo/

Exploration typographique interactive avec p5.js.

<https://bbtgnn.github.io/GTL-web/glyphs>

Outil pour générer et manipuler des glyphes typographiques de façon algorithmique.

<https://bbtgnn.github.io/hofmann-1.0.0/index.html>

Expérience graphique inspirée du Bauhaus et de la typographie expérimentale.

<https://brubsby.itch.io/cozyvec>

Outil vectoriel interactif et ludique pour expérimenter des formes graphiques.

---Comment définir la frontière entre génération et création ?

<https://cellusious.itch.io/alivepaintery>

Simulation de peinture dynamique, où les coups de pinceau génèrent des motifs vivants.

<https://cables.gl/>

Plateforme visuelle par nœuds pour la création interactive et le design génératif en temps réel.

<https://cc-map.net/#/>

Carte interactive pour visualiser et expérimenter des projets de creative coding.

<https://constraint.systems/>

Générateur visuel basé sur des contraintes, idéal pour l'expérimentation graphique algorithmique.

<https://editor.p5js.org/>

IDE en ligne pour écrire et tester du code p5.js, idéal pour le prototypage rapide.

<https://ffd8.github.io/p5-t00ls/>

Collection d'outils interactifs pour la manipulation graphique avec p5.js.

<https://francoisegamma.cat/things/drawif/drawlf.html>

Outil pour expérimenter des motifs et dessins programmatisés simples.

<https://github.com/kmkolasinski/AwesomeBump>

Logiciel open source pour créer des textures et maps 3D à partir d'images.

<https://gradientor.afterimage.cc/>

Générateur de dégradés et de variations colorées pour projets graphiques et visuels.

<https://glyphdrawing.club/>

Plateforme collaborative pour créer et partager des glyphes vectoriels génératifs.

<https://huggingface.co/spaces>

Environnements interactifs hébergeant des modèles IA, utilisables pour générer des visuels ou du code.

<https://hydra.ojack.xyz>

Environnement de live coding visuel pour générer des images et animations en temps réel.

<https://internet-janitor.itch.io/decker>

Outil expérimental pour manipuler du texte et générer des motifs interactifs.

<https://lorenschmidt.itch.io/pattern-mutation-tool>

Générateur de motifs vectoriels mutables et interactifs.

<https://lunarlabs.itch.io/dither-machine>

Générateur d'effets de trame et dithering sur images.

<https://maxbittker.github.io/alphabet-soup/>

Outil interactif pour expérimenter avec des lettres et leur disposition générative.

<https://maxbittker.github.io/broider/>

Générateur graphique inspiré du point de broderie et des textures répétitives.

<https://megabyte-games.itch.io/agave-art-sweet>

Outil ludique de création visuelle et expérimentations pixel/texture.

<https://notwk.london/>

Environnement expérimental pour générer des visuels et typographies interactives.

<https://openprocessing.org/>

Plateforme de partage et découverte de projets p5.js pour le creative coding.

<https://p5js.org/>

Bibliothèque JavaScript pour la création graphique et interactive dans le navigateur.

<https://pentacom.jp/pentacom/bitfontmaker2/>

Générateur de polices bitmap simple et interactif.

<https://pixijs.com/>

Moteur graphique 2D performant pour des expériences interactives et génératives en HTML5.

<https://processing.org/>

Environnement de programmation visuelle pour la création artistique et expérimentale.

<https://raphaelbastide.com/fragile-forge/>

Outil interactif génératif pour expérimenter formes et structures.

<https://replicate.com/>

Plateforme pour tester et exécuter des modèles IA génératifs open source.

<https://runwayml.com/>

Plateforme IA pour la création visuelle et le prototypage génératif.

<https://sandspiel.club/#>

Simulation de sable interactif pour expérimentations visuelles et algorithmiques.

<https://shaderbooth.com/>

Environnement de création de shaders et effets graphiques temps réel.

<https://spacetypegenerator.com/>

Générateur typographique interactif basé sur des formes géométriques

<https://studio.sandspiel.club/>

Version avancée de Sandspiel avec plus de contrôle et de fonctionnalités.

<https://thischairdoesnotexist.com/>

Générateur paramétrable basé sur IA pour créer des chaises.

<https://threejs.org/>

Bibliothèque JavaScript pour créer des scènes 3D interactives dans le navigateur.

<https://www.org/>

Environnement visuel temps réel pour le design génératif et interactif, orienté patching.

<https://walky.space/>

Plateforme de génération visuelle interactive, expérimentale et ludique.

<https://www.aspektedesrasters.de/>

Générateur de motifs vectoriels interactif.

---annexe

Utilisation de l'IA

---tableau consignant mes utilisations de l'IA

date ../../..	outils + version	prompt ou résumé	partie impactée	recul & retour
14/10/2026	Mistral AI Le Chat (modèle Mixtral 8x22B light en local sur Ollama)	« Explique-moi brièvement sur un ton formel et académique, l'impact des Intelligences Artificielles génératives sur la pratique du designer graphique »	Toute la Partie 1.1,	Le but était d'utiliser l'IA une bonne et seule fois dans mon écrit de façon créative en lui demandant de parler d'elle même pour rentrer dans le sujet
18/12/2025	Gemini Pro client web	« Aides moi à traduire ce fichier .md »	Trad. du fichier README. md	Bonne traduction mais modifs perso après en +
02/10/2025	Claude Sonnet 3.5	Relecture et aller retour pour correction de code	Code ayant servi à créer le principe de compo avec le cadre en contour général	Un peu long pour obtenir ce que je veux, mais j'ai pu aller au bout de mon idée en comblant mes lacunes en python et contenu du résultat
23/10/2025	Gemini Pro client web	Formatage des liens et ressources pour respecter la nomenclature.	Annexe (Biblio + Biblio d'outils)	Assez efficace, aucune erreur après vérif'

Ressources Graphiques utilisées

Typographie

L'entièreté de l'ouvrage utilise la typographie Heal The Web A.

Extrait traduit du README.md contenu dans le fichier source de la typographie :

« Une typographie pour soigner le web.

Le projet est né d'une ambition simple mais forte : "Heal the web" (Guérir le Web). Cette police de caractères open source et expérimentale a été conçue pour être accessible à tous. Initialement développée par le studio Rainbow Unicorn pour le rapport 2020 de Mozilla sur la santé d'Internet (Internet Health Report), son dessin évoque des circuits et des réseaux. Les formes géométriques s'y assemblent tel un puzzle abstrait.

Tout en préservant son caractère incisif d'origine, l'objectif a évolué vers la création d'une famille complète, offrant une meilleure lisibilité tout en permettant des expérimentations typographiques complexes. Les deux versions (A & B) intègrent des collections de glyphes alternatifs issus des différentes étapes du processus de création. Ces variantes sont facilement accessibles via les jeux stylistiques OpenType (Stylistic Sets).

La police Heal the Web est distribuée sous licence MPL-2.0 (Mozilla Public License 2.0).

Design : Rainbow Unicorn & Jakub Kanior.

Production : Jakub Kanior »

---Comment définir la frontière entre génération et création ?

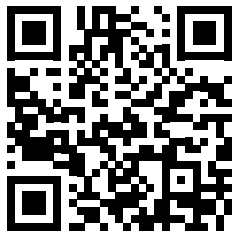
Comment définir la frontière entre génération et création?

Face à l'avènement des intelligences artificielles génératives, ce mémoire interroge la définition de la frontière entre le processus de génération automatique et l'acte de création dans le design graphique. À travers une exploration comparative des pratiques – du "prompting" sur des systèmes propriétaires opaques au "creative coding" en open source – et l'analyse de cas d'étude, cette recherche examine le déplacement du rôle du designer.

L'étude démontre que la délégation totale à la "boîte noire" de l'IA menace l'intégrité éthique et créative, tandis que la maîtrise des règles paramétriques et du code permet de transformer l'algorithme en collaborateur. Il apparaît que la frontière entre génération et création n'est pas technique mais intentionnelle : elle dépend aussi du degré de maîtrise de l'outil par le graphiste. En perspective, ce travail invite le designer à dépasser le statut d'utilisateur passif pour devenir l'artisan de ses propres systèmes.

Mots-clefs

Intelligence artificielle, Design génératif, Creative coding, Algorithme, Intention créative, Open source, Outils, Code



Cet abstract a été généré par l'intelligence artificielle, s'il ne vous plaît pas, vous pouvez générer le vôtre, mais aussi lire le mémoire en version numérique ou explorer ma carte des outils génératifs sur le site :

<https://genere.hovaulysse.com>

