Introduction	4
D'un rapport direct à la nature Exemples de projets Alex Dragulescu : organisation de réseaux sociaux Luna Maurer : développement de champignons	4
Contexte et définitions	8
Le code en tant que matériel du designer Quel approfondissement des connaissances de l'outil? Confrontation avec la programmation	9
La nature, un concept mouvant Rupture entre nature et artefact Approche mécaniste de la nature La nature aujourd'hui, point de vue de l'écologie	10
Point de vue scientifique Épistémologie et interprétation de la vie artificielle	13
Le modèle dans les sciences Représentation des phénomènes Simulation d'un modèle Une application : le biomimétisme	14
Propriétés des algorithmes numériques	18
Un outil de composition algorithmique Une machine à simuler De la théorie à la pratique des algorithmes Réappropriation des modèles scientifiques	19
Versant concret du numérique Matériel logique Manipulation du matériel numérique	23
Structures de programmation Répétition : la boucle La récursivité Évolution génétique	25
Vers une complexité naturelle	32
Des algorithmes naturels ? Quelle liberté pour les productions humaines ? Des artefacts à l'état sauvage Glissements du champ d'action de l'auteur	33 33



Introduction

D'un rapport direct à la nature

J'ai un besoin profond d'être régulièrement en contact avec la nature en dehors de mes activités, en dehors de chez moi, en dehors de la ville. En raison de mon rythme de vie citadin, j'initie ces moments trop rarement. Mais je sais que quand ils arrivent, je les délecte avec passion. Lors de ces explorations, mes sens sont en éveil et ma concentration entière. C'est à chaque fois un plaisir simple et un état des plus positifs qui m'anime à chaque excursion en forêt ou à chaque regard sur les étoiles d'ordinaire invisibles dans les nuits urbaines.

Depuis le début de mes études, je m'interroge sur l'importance de ces moments dans ma production. Est-ce qu'à travers mes travaux je cherche à transmettre ce que je ressens dans la nature? Est-ce transmissible ou est-ce seulement une expérience à vivre qui ne peut rester que personnelle? Intuitivement je pense que ces sensations si uniques ne peuvent être vécues qu'en étant dans la nature. Cependant, les expériences sensorielles fortes forment l'œil et certains de leurs aspects transparaissent plus ou moins directement dans les productions. Au cours de l'histoire, il est fréquent de voir que les artistes et les designers se sont inspirés de la nature.

L'un des domaines que j'ai le plus exploré et qui me tient à cœur en ce moment, à savoir l'informatique, me paraît à première vue particulièrement distant de ce que l'on pourrait entendre par nature. Pour les productions visuelles, il est commun de voir l'ordinateur comme l'outil des formes plates, lisses et virtuelles à l'opposé de cet aspect si concret, profond, varié et texturé que l'on retrouve dans les matières du vivant ou du minéral. Le propre des visuels numériques est associé à la régularité des formes, la monotonie des aplats colorés et la simplicité des textures quand on l'oppose à la richesse des médiums matériels. Que ce soit par la génération de formes calculées ou par la numérisation d'éléments analogiques, il semble y avoir un manque de finesse dans l'aspect sensible des productions informatiques.

Y aurait-il pourtant un rapport subtil entre numérique et analogique permettant de dépasser cette opposition?

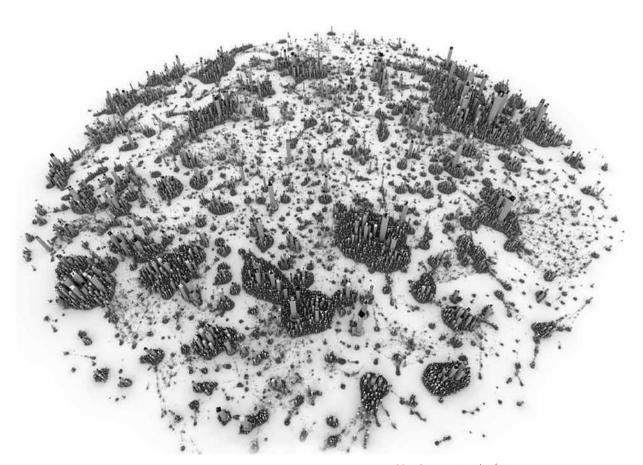
Exemples de projets

Alex Dragulescu : organisation de réseaux sociaux

La data-visualisation est une branche du design graphique largement liée à la programmation. Le traitement visuel des données nécessite parfois des agencements spatiaux spécifiques qui sortent du repaire orthonormé habituel à deux dimensions, base de ces fameux «diagrammes en bâtons». Certains designers vont utiliser des systèmes moins évidents qui serviront souvent à montrer certaines particularités des données tout en offrant une lecture globale dans une forme générale et unifiée du visuel.

Alex Dragulescu a agencé dans sa série <u>Ekisto</u> les comptes d'utilisateurs de communautés en ligne sur une même surface. Sa démarche surprend par le résultat organique qui n'est que produit par le choix de l'algorithme qui a servi à la disposition des comptes d'utilisateurs alors que ces derniers ne sont représentés que par des blocs gris qui font plutôt référence aux immeubles d'une ville. Mise à part leur échelle, les unités d'informations ont toutes la même apparence qui est loin des formes rappelant la nature.

Ce travail fait émerger des types d'organisation différents de manière automatique, c'est à dire sans modification supplémentaire ou intervention manuelle d'Alex Dragulescu, selon chaque ensemble de données qui correspond aux différents réseaux communautaires. L'aspect auto-organisationnel des réseaux sociaux est révélé de manière visuelle.



Vue des comptes du réseau communautaire StackOverflow en juillet 2013 d'Alex Dragulescu.





Blue Fungus, Luna Maurer, Stedelijk Museum Amsterdam, 2009





Luna Maurer: Développement de «champignons»

Dans ses projets Fungus et plus largement dans sa démarche, Luna Maurer utilise des algorithmes dans la plupart de ses projets sans pour autant avoir nécessairement recours au calcul par ordinateur. Elle propose dans Blue Fungus un visuel génératif évoluant sur la durée de l'exposition au Stedelijk Museum d'Amsterdam. Luna a mis à disposition de chaque visiteur une feuille comprenant quatre auto-collants bleus et quatre règles simples tel que:

> Placez votre auto-collant à coté d'un auto-collant qui est déjà collé. La distance entre les deux ne doit pas être supérieure à la taille d'un seul auto-collant.
>
> ▼

La simplicité apparente du processus et de l'action de coller Traduction personnelle. un auto-collant n'a pas empêché l'émergence d'une certaine complexité. Les visiteurs n'avaient finalement que le choix de la position, et encore, les quatre règles conditionnaient l'emplacement possible des auto-collants.

Blue Fungus, Luna Maurer, 2009

Ce projet ainsi que les réalisations du groupe Conditional Design dont Luna Maurer fait partie, mettent en valeur les réactions humaines spontanées et individuelles dans un dispositif strict qui semble automatique. Chaque visiteur a transmis sa propre envie dans la liberté restreinte que lui laissait le dispositif, tandis que l'aspect algorithmique du cadre concrétisé par les quatre indications apporte une cohésion de l'ensemble. Cette procédure systématique a le rôle moteur qui induit une dynamique dans le temps et autorise un déploiement des formes cohérent.

Les caractéristiques que je retrouve dans ces travaux sont communs au vivant puisque les formes semblent se générer et se déployer dans l'espace et dans le temps. Cela est régulièrement mis en avant à travers certains titres que donne Conditional Design à leurs projets tel que Primordial Soup, Cellular Relationships ou Flatland Family Trees.

Ainsi quelle est la part de naturel dans ces dispositifs, reste-t-elle de l'ordre de l'évocation? Provient-elle de facteurs extérieurs ou auraitelle un rapport étroit avec l'utilisation d'algorithmes?



Le code en tant que materiel du designer

Ouel approfondissement des connaissances de l'outil?

Dans son mémoire Outils numériques et design graphique, Kévin Donnot a développé les inconvénients propres à l'utilisation de certains logiciels. Ainsi les logiciels et les productions qui en découlent se cantonnent souvent à mimer des principes analogiques. Les interfaces utilisateurs de création graphique les plus utilisés sont «WYSIWYG» et répondent aux logiques marketing de l'époque industrielle qui tendent à la standardisation des logiciels et à des limitations qui peuvent êtres évitées dans le numérique. Des solutions plus appropriées per-

mettent d'exploiter les caractéristiques du numérique comme la plupart des idées émanant de l'open source et du logiciel libre. Ce mimétisme d'anciennes techniques est tout à fait logique dans la continuité des évolutions humaines, mais ces redondances sousexploitent les capacités des technologies. On peut comparer ce cas à la longue période ou la typographie imprimée a conservé un maximum de particularités de l'écriture manuscrite mal-

gré l'invention de Gutenberg.

Douglas Rushkoff déplore aussi cet encombrement dont se chargent «les utilisateurs ou pire, les utilisés des programmes ». Mais bien qu'étant un fervent défenseur des logiciels libres, il avance une solution plus radicale. Il propose dans le chapitre final de son livre Program or be Programmed de passer de l'utilisation de logiciels à la pratique de la programmation pour ne pas déléguer aveuglément la conception algorithmique. Cette remarque me parait être une nécessité pour la pratique du design assisté par ordinateur. Autant pour réaliser une tâche pour laquelle il y a une unique solution toute tracée l'utilisation d'un logiciel est logique car il s'agit de rester dans les rails, autant elle est moins appropriée lorsque l'on fait du design où il est convenu de chercher sa solution dans un champ des possibles large mais maîtrisé (dont on connaît les limites). Le designer se doit de sortir d'une fixité des usages afin d'adapter au mieux ses outils. La connaissance de langages de programmation permet au moins de comprendre les logiciels utilisés, mais surtout de se les réapproprier.

Cependant il existe toute sorte de logiciels et de langages de programmation qui laissent plus ou moins de liberté d'action. La question est plutôt de savoir quel niveau de maîtrise de l'outil nous voulons atteindre et pourquoi?

Confrontation avec la programmation

Ma pratique est ancrée dans la programmation numérique et cette utilisation fréquente de langages informatiques a évidemment un impact sur la production mais aussi sur ma vison des choses. Cette pratique revêt une grande importance dans cette réflexion car elle permet d'être plus proche des algorithmes exécutés par ordinateur, le code informatique étant précisément le langage qui les décrit.

Pour que le numérique et donc la programmation sous-jacente

Kévin Donnot, Outils numériques et design graphique, 2010, EESAB Rennes dont un extrait est republié dans Graphisme en France, code<> outils <> design, 2012, CNAP

Une interface WYSIWYG, What You See Is What You Get, permet de composer visuellement le résultat voulu qui apparaît directement à l'écran.

Douglas Rushkoff, Program or be Programmed: Ten Commands for a Digital Age, 2010, OR Books

soient perçus comme un médium à part entière et non un moyen parmi d'autres, il faudrait dégager quelques propriétés propres à la programmation.

Il arrive que je me fasse des sessions de programmation sans avoir de projet spécifique à réaliser. Je me lance pour plusieurs heures en partant de quelques bouts de codes provenant souvent de la dernière session, d'une collection de données, dataset, trouvée sur le web ou d'une idée visuelle sans application préconcue. Le début de la session est plutôt intellectuel et théorique : je mets en place les principes qui génèrent des formes, puis je les optimise et j'y relie des contrôleurs qui me permettent de maîtriser les variations en temps réel. De manière très progressive, les moments plus cérébraux se font de plus en plus brefs pour laisser place à l'exploration sensible du résultat émergent, souvent visuel, parfois sonore. Là, je navigue jusqu'à me perdre dans les formes qui semblent varier à l'infini tout en restant en équilibre entre chaos et organisation stricte. Ici le contact avec ces découvertes est direct et fluide, il s'agit bien d'expérimentation, d'expériences concrètes.

Une fois loin du domaine de la réflexion, le mouvement que l'on fait ressemble vite à une chute à travers les phénomènes : on n'a plus d'emprise sur les choses et elles s'écoulent naturellement. Je ne peux m'empêcher de faire le lien entre ces situations et mon expérience avec la nature. Ma position réceptive me mène dans les deux cas à cet état d'observation et de concentration face à des sensations riches et foisonnantes. Cependant d'un côté les formes proviennent d'une longue chaîne de sciences et de technologies humaines alors qu'à l'inverse, la nature précède l'Homme. Willém Flusser place le design, pris au sens large, dans une opposition similaire:

> Le levier [...] a pour but de vaincre la pesanteur par la ruse, de duper les lois de la nature et, précisément par l'exploitation stratégique et rusée d'une de ces lois, de nous libérer de notre condition naturelle. [...] Tel est le design, le dessein fondateur de toute culture, de toutes civilisation: tromper la nature au moyen de la technique, surpasser le naturel par l'artificiel. \"

Le naturel est le point d'appui initial ou la base pour les fondations de toutes les constructions humaines. Mais avec tant de couches de techniques, d'objets automatisés interagissant entre eux et concentrés dans nos ordinateurs, il me semble qu'on retrouve ainsi certaines qualités naturelles. Dans quelle mesure pourrait-on dompter ces luxuriances formelles? Comment pourrais-je transformer ces expériences de programmation plus ou moins passives en acte conscient et productif de design?

Vilém Flusser, Petite philosophie du design, Circé, 2002

La nature, un concept mouvant

La nature est considérée dans le langage courant comme étant tout ce qui ne provient pas de l'Homme. Elle est là de manière dite naturelle, elle n'est pas créée par l'humain. Il s'agit des trois règnes: animal, végétal et minéral. Cette vision qui oppose la nature et l'artificiel semble être la plus commune, ce qui implique que sans réflexion approfondie, on s'y réfère par défaut. Pourtant on constatera que cette appréciation simpliste ne suffit pas pour classer les choses aui nous entourent.

Rupture entre nature et artefact

La pensée d'Aristote avance un argument qui permet de Aristote, Physique, différencier le naturel de l'artificiel. Il montre qu'un II, 1, 192b 8-31 élément naturel possède «en soi-même un principe de mouvement » qu'on percoit par la croissance. l'altération ou le déplacement. Alors que l'artefact «ne possède aucune tendance naturelle au changement». Les plantes poussent et les animaux se meuvent, à l'inverse, une table est inaltérable, c'est son constituant, le bois, qui lui est naturel et sujet au vieillissement.

Approche mécaniste de la nature

Cependant, ce point de vue a ses limites car les frontières entre nature et artefact sont de plus en plus brouillées. Lorsque l'Homme cultive un champ, les plantes de ce champ poussent par la conséquence de l'action humaine. Sont-elles plutôt du domaine de la nature ou de l'artefact? Et le principe premier d'une automobile n'est-il pas justement qu'elle initie un mouvement, par elle-même? L'avancée des productions humaines semble s'approprier les attributs de la nature puis les brouiller sinon les devancer.

La philosophie moderne remet en cause cette distinction entre nature et artefact. Ainsi, Descartes pense la nature comme mécanique.

La nature a le même type de structure mécanique que l'artificiel. Cette vision du réel est dite mécaniste. En niant la limite il intègre les deux termes l'un dans l'autre :

Les principes de la philosophie, quatrième partie, article 403

Je ne reconnais aucune différence entre les machines que font les artisans et les divers corps que la nature seule compose, sinon que les effets des machines ne dépendent que de l'agencement de certains tuyaux, ou ressorts, ou autres instruments, qui, devant avoir quelque proportion avec les mains de ceux qui les font, sont toujours si grands que leurs figures et mouvements se peuvent voir, au lieu que les tuyaux ou ressorts qui causent les effets des corps naturels sont ordinairement trop petits pour être aperçus de nos sens. Et il est certain que toutes les règles des mécaniques appartiennent à la physique, en sorte que toutes les choses qui sont artificielles, sont avec cela naturelles.

Il est intéressant de constater que les principes unificateurs sont les lois de la physique. Qu'un objet soit conçu par l'Homme ou non, ce

qui est sûr, est qu'il obéit dans les deux cas à ces lois dites universelles. La définition de la nature est donc élargie à tout objet existant dans notre univers physique comme l'écrit en 1783 Kant dans les Prolégomènes à toute métaphysique future. La nature devient équivalente à toute chose physique, à l'ensemble du réel matériel. La pensée

Principe d'invariance des lois de la physique. Elles sont applicables aux objets quel que soit leur position dans l'espace ou le temps. La symétrie aujourd'hui, Jean-Marc Lévy-Leblond, 1989, Point Sciences, pp. 55 - 69

mécaniste est de moins en moins étonnante avec les avancées technologiques qui soulignent la faiblesse de la frontière entre nature et artifice. Les progrès génétiques permettent de manipuler le vivant et nos appareils réagissent en toute autonomie aux stimuli de leur environnement grâce à des capteurs en tout genre.

La nature aujourd'hui: point de vue de l'écologie

L'opposition entre nature et artefact, reste pourtant communément admise. Cela revêt depuis la seconde moitié du XX^e siècle une certaine importance liée à la naissance de l'idéologie écologique. Ces mouvements se dressent contre les pouvoirs économico-politiques de la société industrielle dominés par la logique cartésienne.

Un grand nombre d'outils et de technologies sont en inadéquation avec la nature primitive et la modifie profondément. Aujourd'hui, très peu de lieux sur terre n'ont pas été atteints par les productions humaines qui ont changé jusqu'à la composition de l'atmosphère. Une partie de la communauté scientifique propose de différencier d'ailleurs une nouvelle période géologique depuis la révolution industrielle: l'anthropocène. Ce terme met en évidence l'humanité comme la force majeure d'influence sur la géologie terrestre.

Dans ce contexte, l'écologie apporte des idées qui vont à l'encontre de la vision mécaniste de la nature. Certains mouvements vont attribuer aux éléments naturels des valeurs autres que celle de matière première à disposition des humains. ▶ Ces pensées prennent en compte les intérêts humains sur le long terme en incluant l'Homme dans des écosystèmes plus larges.

Frédéric Dufoing, L'écologie radicale, Infolio, 2012, pp. 56-57

Un mouvement de pensée récent, la dark ecology à accepte Timothy Morton, Ecology Without entièrement le constat de cette nature qui est irrémédia-Nature, Massachusetts: Harvard University Press, 2007 blement mutée par les humains. Et certains chercheurs prônent un point de vue en rupture avec l'écologie classique en évitant le combat pour la sauvegarde ou même le retour à un état de nature qui ferait partie du passé. Ils s'intéressent davantage aux rapports entre pensée écologique et pensée technologique. Ils vont tout autant s'intéresser aux questions soulevées par la technologie dans l'environnement que mettre en place une réflexion sur l'écologie des technologies. La fondation Next Nature

Je mets évidemment de côté les penseurs qui y ont un intérêt commercial.

Next nature est la nature créé par l'Homme. Cela peut sembler être une contradiction mais ce n'est pas le cas. Nos artefacts culturels sont devenu si complèxes et autonomes qu'ils ont un fonctionnement plus similaire à des organismes ou des écosystèmes qu'à des objets innanimés."

Dans cette thèse extrême soutenue par la dark ecology on retrouve le retrait de la frontière entre naturel et artificiel. Malgré le fait qu'elle tende à effacer l'opposition entre les deux parties, cette idéologie va dans le sens inverse de la pensée moderne cartésienne. En effet au lieu d'une intégration de la nature à notre monde mécanique artificiel on discerne ici une pénétration de nos artefacts dans les sphères biologiques de la nature.

définit l'environnement humain par:

Traduction personnelle

Next Nature est une plate-forme de publication, d'événements et de recherche en collaboration avec le département de design industriel de l'université des technologies d'Eindhoven basée à Amsterdam

Pour jauger de manière plus fluide l'appartenance au premier cas ou au second en conservant les difficultés de catégorisation, j'utiliserai les substantifs naturalité et artificialité. Cela servira à décrire de manière plus juste les degrés d'intervention ou de non-intervention de l'Homme tout en gardant leur caractères hybrides.

L'Homme occidental, dans sa démarche de rationalisation, recherche des moyens de transcription pour comprendre et décrire le fonctionnement de la nature. Galilée parlait des mathématiques comme d'un «langage décrivant la nature» en 1623 dans son Opus sur les comètes. Les sciences jouent dans un premier temps le rôle d'un outil qui permet aux humains de percevoir les fondements qui sous-tendent la nature. Ainsi les lois physiques à l'aide des mathématigues décrivent les phénomènes qui régissent notre monde physique, les forces de la nature.

Épistémologie et interprétation de la vie artificielle

Les scientifiques cherchent à formuler ces lois qui semblent être à l'origine de la nature, alors que leur compréhension totale est aujourd'hui impossible. Il faut garder à l'esprit que cette approche n'en est qu'une parmi d'autres mais que c'est elle qui fait partie des fondements de notre société occidentale. En effet, les sciences n'admettent que ce qui est rationnel et objectif. Et ce postulat pourrait occulter une partie des obiets d'étude. Cependant nous sommes entourés d'objets techniques et c'est ce domaine rationnel qui est le mieux exploré dans notre société.

Les éléments considérés comme vivants appartiennent initialement au domaine du naturel. Même au delà d'une vision segmentée entre nature et artefact, la vie reste un cas extrême du versant naturel, elle garde une forte naturalité. Cependant le point de vue mécaniste permet de penser, avec l'évolution des techniques actuelles, que l'on peut fabriquer du naturel, voire du vivant.

La vie artificielle est un champ de recherche récent entre informatique et biologie qui tend à synthétiser certaines caractéristiques du vivant. Ainsi certains chercheurs ont développé une interprétation définie plus tard comme forte de la vie artificielle comme le décrit l'un des fondateurs de ce champ de recherche, Christopher Langton:

> Le but ultime de la vie artificielle serait de créer la «vie» dans un autre substrat, idéalement un substrat virtuel où l'essence de la vie aurait été abstraite des détails de sa mise en œuvre dans quelque substrat que ce soit. Nous aimerions construire des modèles qui sont si semblables au vivant qu'ils cesseraient d'être des simulations de la vie pour en devenir des exemples.
>
> ▼

Alors que l'interprétation faible de la vie artificielle conserve Christopher Langton, Studying Artificial Life with Cellular une limite infranchissable entre vie biologique et vie arti-Automata, Physica D, 1986, p. 147 ficielle, son interprétation forte soulève l'hypothèse du vivant comme une question d'organisation et de structure et non de substance. Ainsi pour Christopher Langton le vivant n'est pas restreint aux éléments naturels mais pourrait théoriquement être recréé par l'Homme.

Même si l'ancien fantasme que portait l'humanité pour la création du vivant soulève aujourd'hui un certain nombre de questions au regard des avancées technologiques actuelles, je me contente ici de rechercher certains aspects isolés de nos artefacts ayant une certaine naturalité.

Le modèle dans les sciences

en le simplifiant pour faciliter sa compréhension.

Représentation des phénomènes

Pour mieux comprendre sur quoi se basent ces champs d'exploration à la croisée des sciences et de l'ingénierie qui disent pouvoir créer de la naturalité, nous allons approfondir la notion de modèle. Les sciences utilisent largement des modèles et depuis longtemps, l'astronomie et les mathématiques antiques en faisait déjà usage. Le modèle scientifique est un système, une structure théorique qui permet de représenter l'objet d'étude des méthodes plus rigoureuse

Ptolémée à notamment approfondis cette notion avec

Dans un premier temps la notion scientifique de modèle se doit de correspondre à la réalité des phénomènes. Le modèle décrit les objets d'étude et les représente. Ainsi une représentation d'un bal-Ion peut être un rond tracé sur une feuille, son modèle est alors le cercle. Le modèle permet d'isoler et d'extraire une ou plusieurs caractéristiques du réel comme le fait la représentation pour transmettre un sens. Dans notre exemple, c'est le pour-Du grec ancien skhēma. tour du ballon qui est schématisé bà travers le cercle alors que d'autres détails de l'objet sont délaissés. En sciences cette réduction des aspects du réel a une importance primordiale au delà du fait qu'elle est la source de la différence entre la représentation et l'objet représenté. Elle permet de clarifier la

nature complexe des choses pour révéler leurs principes

de rendre lisible et compréhensible les fonctionnements

«manière d'être», «forme»

Réductionnisme en épistémologie (philosophie fondamentaux. D'une certaine manière, c'est un moyen des sciences). Cela ce fait selon des règles précises.

Simulation d'un modèle

de notre environnement.

Une simulation est la mise en application d'un modèle qui lui est une structure générique, théorique. Dans ce sens, on simule un phénomène grâce à son modèle et le résultat en est la simulation. Le modèle possède des entrées par lesquelles les données d'une simulation vont influencer son comportement. Et il a également des sorties qui donnent le résultat de la simulation. Ce résultat est l'objet étudié dans la situation correspondant aux entrées.

Considérons le modèle de l'addition, il est constitué d'une suite de lois logiques qui à partir de deux nombres en donne un troisième qui en sera la somme. Une simulation possible de ce modèle serait de mettre en application l'addition de 3 et 2. Les deux entrées de la simulation sont 3 et 2 et la sortie est 5. Pour prolonger l'exemple, imaginons que je fasse l'action bien réelle dans l'espace physique de déposer 2 pommes dans un panier en contenant déjà 3. La simulation d'addition que l'on a fait plus tôt est une façon de représenter l'aspect du réel qu'est la quantité de pommes qu'il y a dans mon panier.

L'addition est une chose tellement habituelle, qu'elle est ancrée dans notre manière de percevoir le monde. On pourrait être tenté de dire que l'addition est naturelle ou au moins, qu'elle est incluse dans le réel qui nous environne, dans la nature au sens le plus large et donc qu'elle est contenue dans l'expérience physique de l'ajout de pommes dans le panier. Or, ce n'est pas le cas, c'est une opération humaine d'abstraction.

Les quelques premiers mois où je me suis sérieusement mis aux langages de programmation, lorsque je n'avais pas l'esprit très concentré, je me suis surpris à plusieurs reprises à imaginer ou peut être percevoir les structures sous-jacentes aux motifs graphiques que je croisais au quotidien. Au contact de ces stimuli visuels, nervures du bois, pavages, ondes dans l'eau, maillages de tissus, mon esprit se mettait en marche et essayait de décrypter les organisations, apparemment sans démarche consciente de ma part. Quelles logiques il y a-t-il là dessous? Quelles équations ou plutôt lignes de programmation seraient cachées dans cette forme que potentiellement je pourrais reproduire dans mon ordinateur. Je fus étonné qu'une pratique influe à tel point sur ma perception.

Une application: le biomimétisme

Pour revenir sur un cas plus concret de transfert de caractère naturel qui exploite largement nos connaissances scientifiques, on peut considérer une technique du domaine de l'ingénierie: le biomimétisme. Il s'agit d'une transmission de potentiels techniques présents dans la nature pour des utilisations dans les sciences ou les technologies. Ce procédé met en avant un mode d'influence de

la nature sur les productions humaines. Le biomimétisme est un mot récent be mais il s'inscrit à la suite d'une longue histoire de l'utilisation de processus naturels via l'observation de l'environnement. Dès le 4e siècle avant J.-C., Aristote observait les incroyables capacités d'adhésion des geckos sur les parois verticales et Léonard de Vinci au 15e siècle cherchait à comprendre le fonctionnement des ailes d'oiseaux et de coléoptères pour réaliser les plans de sa machine volante.

Aujourd'hui, le biomimétisme passe par l'observation et l'étude du fonctionnement d'un aspect précis de la nature, puis par une démarche de modélisation. Janine Benyus, scientifique qui a popularisé le biomimétisme, voit la nature comme « modèle, mesure et mentor » dans le but d'en extraire le potentiel

Janine Benyus <u>Biomimicry :</u> Innovation Inspired by Nature, 1997

technique par un modèle réutilisable dans l'industrie.

Ainsi, ce type d'inscription de naturalité dans nos artefacts est différent d'une interprétation subjective de la nature ou d'une inspiration formelle plus superficielle.

Appliquée au vivant, l'idée du mimétisme est logique. En effet les structures des organismes sont optimisées constamment depuis près de 3,8 milliards d'années par les mutations et les croisements génétiques combinés à la sélection naturelle. Le fonctionnement des formes de la nature renferme une économie de matière et d'énergie très raffinée. Et nous pouvons comprendre et simuler ces principes dans la mesure où nous vivons dans des milieux similaires avec des lois physico-chi-

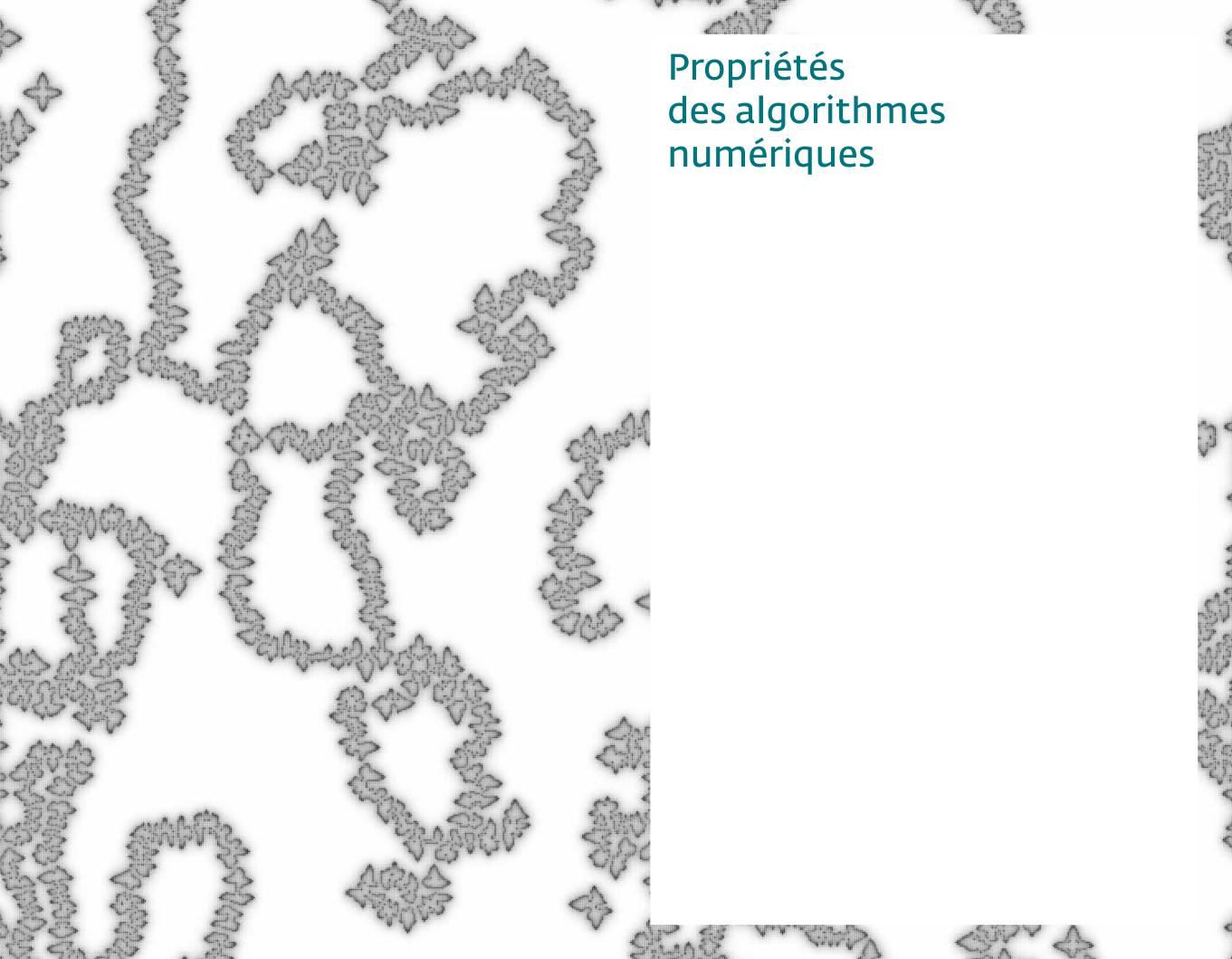
migues semblables.

Le biomimétisme est une pratique qui contribue à l'innovation technologique. Mais n'existe-t-il pas d'autres domaines qui mériteraient de s'inspirer des structures de la nature? Dans les champs du design, c'est l'architecture qui semble avoir le plus intégré ces pratiques. Ce n'est pas si étonnant au regard de sa

ces pratiques. Ce n'est pas si étonnant au regard de sa proximité avec l'ingénierie.

Je pense notamment aux batiments du FRAC Centre réalisé par Jacob et Mac Farlane ou de la fondation Louis Vuitton par Frank Gehry.

On pourrait imaginer toutes sortes de modalités des plus directes aux plus lointaines pour réutiliser ces principes décelés dans la nature. Cela va du prélèvement de données brutes à l'adaptation de systèmes plus ou moins fidèles au réel, pour des finalités techniques comme esthétiques et sémantiques.



Un outil de composition algorithmique

Une machine à simuler

Le biomimétisme a vu le jour si tardivement parce qu'il lui manquait un élément essentiel : la généralisation de l'ordinateur. L'informatique permet un basculement majeur dans la simulation de notre environnement qui modifie aussi notre perception. Aujourd'hui, il devient très aisé de simuler des modèles. Jusque là, les sciences se sont occupé de théoriser des modèles. Alors que depuis les premiers ordinateurs on peut opérer des simulations dont le résultat sera directement perceptible par nos sens ou du moins par la vue et l'ouïe dans un premier temps.

Les algorithmes qui sont contenus dans les langages de programmation sont des modèles numériques. Ils sont matérialisés par une liste de procédures et leur spécificité est d'accepter pour entrées et sorties uniquement des valeurs chiffrées. La programmation est donc la partie concrète qui décrit les algorithmes, eux-mêmes constituant la partie abstraite, théorique. Pour représenter les autres types de modèles, non computationels, il existe les langages de programmation dits théoriques.

En 1936, Alan Turing donne pour la première fois une définition précise de l'algorithme en posant les bases théoriques de l'informatique à travers un modèle qui décrit le fonctionnement d'un Alan Turing, On Computable ordinateur bien avant l'invention de ce dernier. ▶ Le propre Numbers, with an Application to the de cette machine de Turing dite universelle, est que ce Entscheidungsproblem, 1936, pp. 230-265 modèle, selon ses entrées peut simuler tous les algorithmes possibles. C'est à dire qu'on peut décrire dans une de ses entrées, la manière dont il va transformer les autres données

Le langage de programmation, qui permet de transmettre des instructions à un ordinateur est composé d'algorithmes qui sont compréhensibles par l'Homme et par la machine. Grâce à une série d'instructions dont la grammaire rigoureuse ne permet qu'une unique interprétation, on peut décrire un système de manière efficace et succincte.

entrantes en données sortantes.

De la théorie à la pratique des algorithmes

Au lieu de simulation d'un modèle numérique, pour l'informatique, on parlera plus largement d'exécution de programme. Ce glissement de langage est évocateur du changement qu'apporte l'informatique. D'un côté la modélisation qui va de paire avec la simulation, signifie le passage d'une chose tangible à sa représentation via une réduction, autrement dit à la réalisation de sa maquette. De l'autre, l'exécution d'un programme renvoie plutôt à la construction, la mise en œuvre d'une chose à partir de sa description, d'un plan préétabli. Cela vient de l'attribution de nouveaux buts au modèle et à la simulation avec l'arrivée de l'informatique. Le modèle sert comme outil de dissection d'objets de la nature pour faciliter sa compréhension et la simulation est un moyen de vérifier la cohérence du modèle ou un moyen de prédiction des comportements des phénomènes étudiés. Alors qu'avec l'informatique, le modèle devient le moyen de réaliser directement des objets artificiels grâce

aux périphériques de sortie. La sortie, l'output du modèle devient l'imprimante dans un sens large, qui ancre les simulacres dans le réel autrement dit qui les matérialise. D'abord par des chiffres, puis des textes, de l'image, de la vidéo, des objets en volume ...

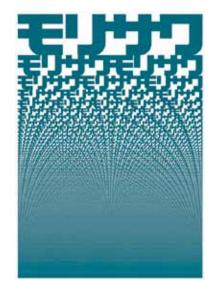
Le scientifique Gary William Flake pointe cette évolution en annoncant que «la simulation devient une forme d'expérimentation dans un univers de théories » et dans le même sens, Vilèm Flusser met Gary William Flake, The Computational en balance formes idéales et matière concrète:

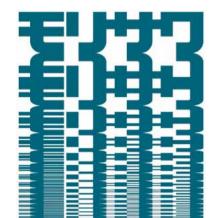
Beauty of Nature, 1998, MIT Press

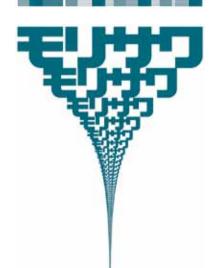
20

Il s'aaissait de donner forme à une matière donnée pour la faire apparaître: maintenant il s'aait plutôt de remplir de matière le torrent débordant des formes jaillies de notre vision théorique et de nos appareils, pour les "matérialiser". [...] Jadis, on formalisait le monde donné; maintenant, on

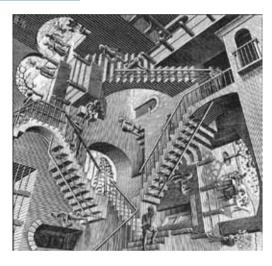
Ces nouveaux mondes ont comme principale base les Vilém Flusser. Petite philosophie sciences. Toutes ces théories scientifiques qui décrivent du design, 2002, Circé, p.92 le réel qui nous entoure constituent la charnière avec leur réutilisation pour la conception d'objets.



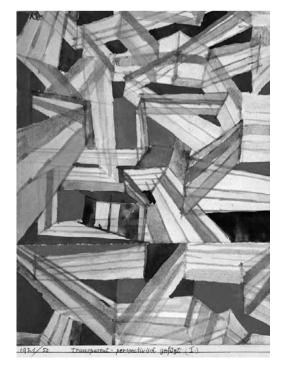




Maurits Cornelis Escher, La Maison aux escaliers, 1951



Paul Klee, <u>Transparent - Travail</u> de la perspective, 1921



Réappropriation des modèles scientifiques

Il est naturel de dire que les objets technologiques s'appuient sur les sciences, mais dans ce mouvement, les productions de design graphique peuvent tout autant être issues de ce potentiel formel. L'image suit le même chemin. On a besoin d'une prise de recul sur notre environnement pour le cerner puis pour s'en détacher afin de concevoir des choses différentes et cela s'est fait par la connaissance théorique même si elle va de paire avec l'approche empirique. Il a fallu s'intéresser aux savoirs issus de la géométrie pour utiliser la perspective au service d'un réalisme en peinture à la Renaissance puis pour le dépasser dans la période de l'art moderne.

Dans les deux premiers exemples à gauche, ce qui fonde la perspective est modifié et révèle cette capacité nouvelle de la représentation. Le réalisme de la gravure d'Escher n'est plus qu'un effet trompeur au service de son travail de reconfiguration de l'espace et Paul Klee s'en détache complètement pour mettre à nu la perspective.

Cependant ces deux images gravées et peintes, naissent d'un processus qui relève encore de l'ordre de la composition de matière pour finalement arriver à des formes, si l'on reprend les mots de Wilém Flusser. Les images de synthèse, quant à elles, sont des compositions de formes théoriques qui sont par la suite matérialisées lors de l'exécution. Cela est possible grâce aux langages de programmation qui permettent de décrire précisément ces formes.

Ici, John Maeda a réalisé ces visuels grâce à un langage de programmation, il a écrit de trois manières différentes l'algorithme qui produira la répétition toujours plus réduite du titre du magazine. Ainsi se crée une impression de perspective car un élément plus petit paraît plus loin, mais c'est le lecteur qui perçoit cet effet de distance par son expérience personnelle de l'espace. La programmation auraitelle en commun avec notre environnement naturel cette possibilité de systématisation?

Le designer qui emprunte cette voie se retrouve à manipuler des rapports quantifiables entre les éléments graphiques. Il compose les règles selon lesquelles vont s'agencer les éléments visuels de manière systématique. Cette possibilité donne une nouvelle importance aux théories qui proposent une mesure systématique des effets physiologiques définissant en partie le sens des signes visuels. Cette sémiologie graphique permet de faire le lien entre le

code qui décrit les attributs des formes et le sens transmis au lecteur. Comme l'ont exprimé Vassily Kandinsky dans ses écrits pour l'art abstrait et Jacques Bertin pour le design d'informations.

Vassily Kandinsky, <u>Point et ligne</u> <u>sur plan</u>, 1926, Gallimard

Le calcul automatisé combiné avec l'idée d'universalité d'Alan

Turing permettant une grande adaptation, offre un autre

statut à la simulation. Mais si l'on peut tout faire avec l'informa-

tique à condition que les données soient numériques, quelle est sa spécificité qui en ferait autre chose qu'une machine qui centralise ce que l'on faisait déjà sans elle?

Versant concret du numérique

Materiel logique

Lors d'un workshop à la Gaîté Lyrique mené par l'artiste Martin Howse, il nous a amenés à ouvrir les cadres classiques de la programmation en la pratiquant dans un environnement totalement différent. Au cours du workshop chacun a fabriqué avec des moyens rudimentaires des portes logiques, base de l'électronique numérique et donc de l'informatique. L'idée étant de remplacer les courants électriques porteurs d'informations par des flux d'eau. On a remplacé les circuits imprimés par des gobelets et des pailles. On remplissait ou non les entrées de chaque porte logique et en fonction de son type, il en sortait un résultat encodé de la même manière: un flux d'eau correspondait à *vrai* et une sortie sans eau à *faux* comme le montre l'illustration ci-dessous. Il suffit ensuite d'assembler ces portes logiques en connectant leurs sorties avec les entrées des suivantes pour complexifier les opérations.

Cette approche concrète de la base logique de la programmation m'a fait réaliser la nature élémentaire matérielle sous-jacente au code et la puissance numérique de l'informatique non pas par sa complexité mais par sa puissance de calcul, autrement dit sa capacité de traitement d'une quantité massive d'informations avec une extrême rapidité qui se compte en milliards d'opérations par seconde pour les processeurs actuels.

Manipulation du matériel numérique

Ce constat de la quantité d'informations élémentaires amène à comparer l'informatique et la terre glaise. Cette terre est composée d'un nombre inconcevable de grains, comme des grains de sable et tellement fins qu'une fois isolés ils sont presque imperceptibles au point que l'on a tendance à les oublier. L'avantage de leur taille est la possibilité d'en rassembler des quantités astronomiques entre nos deux mains pour ensuite les manipuler. À notre échelle ces rassemblements de grains peuvent configurer une infinité de formes lorsqu'on les manipule.

Il en est de même pour le numérique: une image sur un écran est composée de milliers d'unités d'information basique, les couleur des pixels qui peuvent être configurées pour révéler des formes. À un niveau inférieur, tout type de fichiers numériques que nous utilisons sont des assemblages d'unités d'informations élémentaires rassemblées: pixels, vecteurs, caractères, etc. Et au plus bas niveau, utile pour le processeur, l'information est encodée en binaire, ces fameux o et 1.

Cependant, il existe une différence fondamentale dans la façon avec laquelle on manipule ces deux matériaux. La maniabilité de la terre qui passe principalement par la force de la main n'est qu'une première phase pour la production d'objets numériques. Modeler une forme par la programmation n'est pas une chose si spontanée et habituelle pour la plupart d'entre nous car elle transite par une forme sémantique. L'écriture du code passe par le geste de la main puis par le signe graphique, le mot, ou à un plus bas niveau, la lettre. L'agencement de forces appliquées à la matière a pour but premier l'agencement des signes et non la qualité de ces gestes (pression, angle, vitesse ...).

Mais ce n'est pas tout, le troisième type d'intermédiaire dans la production d'objets numériques est la machine. Contrairement à l'écriture d'un texte littéraire, la production de signes ne constitue pas l'objet final. Ces signes forment des instructions dont le but premier et d'être interprétées par la machine et non par un lecteur humain. Il faut encore exécuter le programme. La machine, plus exactement l'ordinateur, va agencer la matière automatiquement selon les instructions du code. Les grains ainsi configurés, vont figurer des formes avec des trames de pixels LCD, de points d'encre ou de mailles de tissu.

Une fois le code écrit, l'objet se forme automatiquement, sans intervention humaine. On ne compose que les conditions initiales, les lois de la nature de l'objet par lesquelles la matière va s'agencer. Le produit final s'éloigne de la main. Malgré la nécessité de l'apprentissage de langages nouveaux, les répercussions de cette distance est une maîtrise plus rigoureuse des grains car on est forcé de décrire précisément la manière dont ils sont gérés individuellement. Cette maîtrise est aussi plus globale dans la mesure où on ne les décrit pas un par un, l'un après l'autre, mais avec des principes de programmation qui permettent d'écrire un minimum de code pour configurer un maximum de grains tout en restant précis. Nous allons en aborder quelques-uns dans le prochain chapitre.

Deux exemples d'exécution d'une porte logique AND réalisées lors du workshop avec Martin Howse.





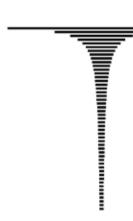
1ET 1 donne 1

o ET 1 donne o

Structures de programmation

Répétitions: la boucle

La boucle offre la possibilité de répéter une suite d'instructions un certain nombre de fois sans réécrire ces lignes. L'exemple ci-contre illustre une boucle qui va produire le dessin d'un rectangle un certain nombre de fois.



À gauche, on a le programme qui décrit l'algorithme, il constitue l'entrée du modèle de Turing qu'est l'ordinateur. Lorsqu'on

exécute le programme, on fait une simulation du modèle qui va produire en sortie l'élément de droite.

Boucle'for' qui affiche un rectangle 'on à chaque itération tant que'i' est èle inférieur à 50 avec Processing.

La boucle est très répandue dans la programmation car elle permet d'appliquer des instructions non pas à un élément unique, mais à une série d'éléments. Ce principe implique une efficience accrue car avec une ligne de code supplémentaire on dessine un grand nombre de rectangles qui auraient très bien pu être décrits sans boucle avec 50 fonctions rect() à la suite. Cette économie impose par la même occasion une systématisation des éléments pris dans la boucle, ils doivent tous suivre la même logique.

La boucle sert de base à d'autres principes tel que la récursivité.

La récursivité

Un autre principe, la récursivité, permet d'atteindre une certaine complexité avec relativement peu d'instructions. En programmation la fonction récursive est le fait qu'un groupe d'instructions, une fonction, possède une de ses instructions qui fait appel à ce même groupe, comme le montre l'exemple ci-contre. Ainsi lorsque l'on exécute ce type de fonction, les premières lignes sont calculées jusqu'à la ligne appelant la fonction qui va exécuter à nouveau les lignes jusqu'au rappel de la fonction et ainsi de suite. Ces propriétés font que la programmation constitue un matériau adéquat pour créer des modèles similaires au monde naturel car on y retrouve des principes identiques.

Comme pour la boucle, un algorithme récursif nécessite une condition de sortie sinon la fonction se répéterait à l'infini. Ce type d'algorithme est utilisé par exemple pour explorer une arborescence de dossiers. Chaque dossier en contient d'autres, ils sont imbriqués et on ne sait pas à l'avance jusqu'à quel niveau. La fonction va donc se répéter pour chaque niveau à partir du dossier racine et la condition

Exemple simple de L-système ayant pour règles de transformation entre chaque itération :

0 deviens **1**[**0**]**0** et **1** deviens **11**

Pour passer à la forme graphique, appliquer

- 0: dessiner un segment de ligne (terminaison ou «feuille»)
- 1: dessiner un segment de ligne
- [: retenir la position et l'angle actuel, tourner à gauche de 45°
-] : restituer la position et l'angle retenue, tourner à droite de 45°

Axiome

0

Première itération :

1[0]0

Seconde itération :

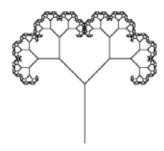
11[1[0]0]1[0]0

Troisième itération :

1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

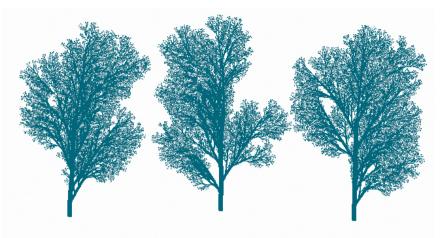
Quatrième itération :

11111111[1111[11[1] 0]0]1[0]0]11[1[0]0] 1[0]0]1111[11[1[0] 0]1[0]0]11[1[0]0]0

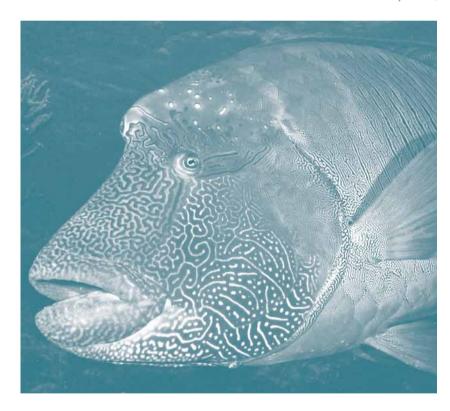


Dixième itération (visuel réduit 15 fois)

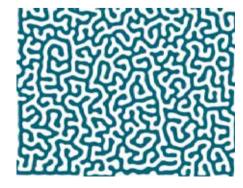
Exemple de L-système incluant une certaine part de hasard.



Humphead Maori Wrasse Andrew J. Green / Reef life Survey



Deux simulations du modèle de réaction-diffusion avec des paramètres différents.





de sortie est le moment où la fonction se trouvera face à un dossier qui ne contient pas de dossier.

Beaucoup de formes naturelles présentent une notion de récursivité. La disposition des branches d'arbre partant du tronc forme un motif qui semble se répéter jusqu'aux branches les plus fines et il en est de même avec les nervures des feuilles, certains camouflages d'animaux, les montagnes et les rivières à différentes échelles. La division cellulaire présente encore plus nettement la procédure de la récursivité par l'aptitude d'auto-réplication de chaque cellule. Par exemple, un L-système est un type d'algorithme récursif développé par le biologiste Aristid Lindenmayer, il permet de simuler le développement de plantes et de certaines bactéries. Les L-systèmes peuvent générer une grande variété de formes complexes grâce à une même structure de règles relativement simples.

La récursivité évoque inévitablement cette notion de « mouvement en soi» dont parlait Aristote. Le mouvement en mécanique étant basé sur la conservation d'un objet pour différentes positions dans l'espace et dans le temps, une fonction récursive semble bien posséder le moyen de se répliquer de manière similaire. De plus il y a une transmission de l'évolution de la forme entre chaque itération mais aussi une transmission de la position et de l'angle entre chaque répétition de ce «Y» dans la forme visuelle du premier exemple de L-système.

Pour ce qui est du vivant, Alan Turing a décrit en 1952 les relations entre des processus physico-chimiques et des formes du vivant dans l'article «The Chemical Basis of Morphogenesis». Alan Turing montre que des réactions physico-chimiques peuvent former une grande variété de motifs que l'on retrouve dans le vivant à partir d'état uniforme. Ainsi opère la morphogenèse qui est la mise en place des caractéristiques d'un être vivant lors de son développement à partir de la cellule-œuf qui contient l'ADN.

Évolution génétique

Plus récemment, avec la compréhension progressive de l'ADN on parvient à faire le lien entre ce code génétique et les formes visibles du vivant. Cependant la recherche sur les fonctions et la nature de l'ADN est loin d'être aboutie. Je ne peux pas en saisir les avancées mais il est certain que l'ADN ne peut être rapproché facilement de la simple programmation dans la mesure où une part essentielle de l'ADN reste inconnue dans ses effets. Eependant leur

point commun indéniable est que les deux sont des codes, l'ADN non codant, autrefois ils servent donc tous deux au support et à la transmission appelée «ADN poubelle». d'informations.

Cette part est nommée

Une espèce reste stable par la transmission de son patrimoine génétique lors de la reproduction d'une génération à l'autre. Il y a de ce fait une continuité dans l'évolution des formes du vivant. Cependant c'est l'évolution génétique, les changements lors de la transmission des gènes par des mutations du code génétique qui permettent à la faune et la flore de se complexifier et de se diversifier. Le champ de la vie artificielle a fait naître des algorithmes qui miment ces cycles de mutations/sélections du génome appelés algorithmes évolutionnaires. Ces algorithmes d'évolution génétique ont déjà fait leurs preuves dans le milieu de l'ingénierie. Ils consistent à tester des solutions générées automatiquement selon des conditions définies dans l'algorithme puis à garder les meilleures et de

recommencer le processus avec des variations des meilleures solutions précédentes.

Pour ce qui est du design graphique, certains designers s'en sont approchés même s'ils restent des cas isolés. Dès les années 60, Karl Gerstner, s'appuyant sur des approches scientifiques et ayant accès à l'informatique, a mis au point un moyen de calculer des solutions graphiques en partant de tables qui listent les éléments et leurs caractéristiques possibles d'un design. Il a mis en place des algorithmes pour épuiser les formes en croisant toutes les possibilités puis en y prélevant automatiquement les plus adéquates. Ici Karl Gerstner s'appuie non seulement sur la puissance de calcul informatique mais adapte également son processus décisionnel à cet outil informatique.

Table de discrétisation des éléments d'un design. Karl Gerstner, Designing 30

				grammes, Lars Mülle	
a Basis					
1. Components	11. Word	12. Abbreviation	13. Word group	14. Combined	
2. Typeface	21. Sans-serif	22. Roman	23. German	24. Some other	25. Combined
3. Technique	31. Written	32. Drawn	33. Composed	34. Some other	35. Combined
Colour	1		1		
1. Shade	11. Light	12. Medium	13. Dark	14. Combined	
2. Value	21. Chromatic	22. Achromatic	23. Mixed	24. Combined	
Appearance					
1. Size	11. Small	12. Medium	13. Large	14. Combined	
2. Proportion	21. Narrow	22. Usual	23. Broad	24. Combined	
3. Boldness	31. Lean	32. Normal	33. Fat	34. Combined	
4. Inclination	41. Upright	42, Oblique	43. Combined		
Expression					
1. Reading direction	11. From left to right	12. From top to bottom	13. From bottom to top	14. Otherwise	15. Combined
2. Spacing	21. Narrow	22. Normal	23. Wide	24. Combined	
3. Form	31. Unmodified	32. Mutilated	33. Projected	34. Something else	35. Combined
4. Design	41, Unmodified	42. Something	43. Something	44. Something	45. Combined

replaced

omitted

added



Des algorithme naturels?

Le chercheur Bernard Chazelle propose l'association algorithmes naturels et la définit par la négative : «Ce sont les algorithmes qui n'ont pas été créés par l'Homme ». Ce type d'algorithme serait-ils complètement dépourvu d'artificialité? Non, une nuance d'artificialité persiste : les algorithmes sont des concepts humains. C'est l'Homme qui perçoit des algorithmes dans la nature. On retombe sur le même problème soulevé par les modèles, ce sont des fabrications abstraites de la réflexion humaine décrivant des processus qui eux peuvent être naturels.

- La forme typique du pelage du zèbre est inscrite en code génétique dans une séquence de la chaîne de son ADN. Est-ce pour nous, observateurs, un algorithme naturel encodé dans l'ADN de l'espèce et ses simulations sur tel ou tel zèbre en particulier?
- Pour Bernard Chazelle, le terme correspond à la réalité biologique, les algorithmes naturels seraient l'exact reflet de ce qui se passe dans la nature. On peut dire qu'un algorithme naturel est exactement à la frontière de la définition d'un modèle car il n'est pas réduit à la description de quelques facettes sélectionnées mais qu'il comporte toutes les caractéristiques de l'objet étudié. En cela il est beaucoup plus complexe, chaotique, pas forcément dans sa formulation potentielle, mais dans sa simulation et donc il n'est pas calculable avec les ordinateur actuels.
- La notion d'algorithme naturel est extrême, à la limite de la contradiction, elle rappelle le vieux rêve de décrire le monde par une formule mathématique. De tels algorithmes sont peut-être utiles pour les sciences mais ils ont peu de chances d'être exportés en l'état dans d'autres champs.
- En laissant de côté la question soulevée par l'interprétation forte de la vie artificielle quand Christopher Langton évoque l'aspect interchangeable du support organique de la vie, les algorithmes naturels de Bernard Chazelle paraissent inconcevables du fait de leur complexité, notamment dans les interactions qu'ils entretiennent entre eux. Comment peut-on en isoler un sans ignorer ses relations avec les autres? À moins qu'il n'y ait qu'un unique algorithme naturel, peut-on réduire la complexité naturelle en procédant par segmentation en plusieurs descriptions autonomes ?

Quelle part de liberté pour les productions humaines ?

Des artefacts à l'état sauvage

Pour en revenir aux artefacts, l'aspect sauvage de la nature pourrait être assimilé à la face autonome de la technologie, dans le sens ou le relève Kevin Slavin dans sa conférence TED. Le sauvage est contraire à la nature de l'Homme, ce qui est sauvage est perçu comme dangereux pour les humains. Cet aspect primaire a une connotation négative. La non maîtrise repousse l'Homme alors même qu'il cherche à automatiser les processus; on cherche à avoir le résultat avec un minimum d'effort. On veut donc

une maîtrise *magique* qui souvent se trouve être une délégation de l'acte. Mais à qui ou à quoi le délègue-t-on ?

Dans l'affaire du krach des échanges boursiers à haute fréquence dont parle Kevin Slavin, les algorithmes de trading ont eu des conséquences imprévisibles de par leur nombre et leur rapidité. Par le biais de techniques humaines que l'on croyait maîtrisées l'homme a créé un système complexe et autonome pour parvenir à ses fins. Les décisions que les humains prenaient jusque là se retrouvent prises par des algorithmes. Finalement l'Homme devrait toujours en être maître dans la mesure où il crée ces lignes de codes qui régissent les réactions des algorithmes. Et les programmes produisent bien des réactions et non des décisions dans la mesure où ils n'ont pas de libre arbitre. Cependant notre manque de lucidité sur les répercussions de l'automatisation des tâches nous surprend souvent. Des choses imprédictibles émergent de nos systèmes rationnels.

Pour parer à ce problème de l'imprédictabilité Stephan Wolfram évoque en 2001 une approche scientifique basée sur l'expérimentation et non la démonstration dans l'étude des systèmes complexes°. Avant l'informatique les sciences se basaient principalement sur la démonstration pour résoudre les questions alors qu'aujourd'hui on peut grâce aux puissances de calcul auxquelles on a accès, simplement calculer les questions avec des algorithmes pour approcher ou atteindre les réponses. Cependant comme le montre les recherches sur le chaos, les calculs révèlent des faiblesses. Dans un système simple où l'on connaît les règles qui en sous-tendent l'évolution, les prédictions que l'on fait de l'état du système après un temps donné s'avèrent parfois inutiles, ce cas est connu sous le nom de l'effet papillon. La plus infime variation des conditions initiales apporte, après un certain nombre d'itérations, des changements radicaux. Même l'approximation dans ce cas là est inutile et les calculs ne nous apportent pas forcément d'informations fiables sur les tendances générales du système.

Les algorithmes que l'on utilise quotidiennement, pour peu qu'on en connaisse la description exacte, ne sont pas aussi mécaniques que ce que l'on pourrait supposer. De notre point de vue ils peuvent produire des résultats inattendus car nous ne pouvons pas être conscients de l'ensemble de leurs interactions internes en tant que systèmes complexes et externes, avec leur environnement.

Glissements du champ d'action de l'auteur

La série des <u>Fungus</u> de Luna Maurer révèle l'aspect autonome des formes créées par les interactions entre des algorithmes et un système complexe. Ce système, les visiteurs et le lieu d'expositon, dans lequel les algorithmes, les règles, sont exécutés constitue l'environnement dans lequel ils se déploient. On remarque un premier basculement radical dans la position de l'auteur du projet par rapport à une production d'art ou de design plus classique. La forme finale n'est pas un choix maîtrisé de Luna Maurer.

Dans l'art, les prémices de cette attitude remontent au mouvement Dada comme l'a mis en perspective Thomas Dreher. En 1921, Tristan Zara publie Pour faire un poème

dadaïste qui propose de découper des mots d'un journal, de les mélanger dans un sac puis de les piocher pour former un poème. Ici la partie autonome du processus qui se détache de la main de

Thomas Dreher, Conceptual Art and Software Art: Notations,
Algorithms and Codes, 2007

l'auteur est le hasard. Dans les années 70, les <u>Wall Drawings</u> de Sol LeWitt révèle une division entre la conception et la réalisation d'une œuvre. Sol LeWitt s'arrête à l'énonciation des instructions qui formeront, après leur exécution, une peinture murale. L'attention est portée sur la création du modèle qui porte en lui le potentiel de la future peinture.

Casey Reas présente en 2004 <u>{Software}</u> Structures qui est une traduction de <u>Wall Drawings</u> en langage de programmation. Il expose les instructions verbales, le code source et le programme qui s'exécute. En passant de la notation verbale au code informatique, Casey Reas rend le texte potentiellement actif. Dans les deux cas, les textes sont descriptifs mais dans le second les instructions sont exécutables par ordinateur. Et comme l'écrit Thomas Dreher, ce changement augmente l'attention donnée à la forme. La conception, les lignes de programmation, se rapprochent davantage du percept, images générées. Cela peut s'expliquer par le fait que les langages de programmation ne peuvent être interprétés que d'une seule manière, mais c'est principalement par le fait que l'exécution informatique est automatique.

Pour Sol LeWitt, l'idée de l'œuvre prime sur le résultat. Sa position d'auteur se situe dans la conception et non dans la réalisation qui devient superficielle. Alors qu'avec la programmation, l'auteur s'occupe davantage de la réalisation formelle. Cela constitue un second glissement ou un léger contre-balancement de son champ d'action entre structure et forme. Je qualifie ce second glissement opéré par l'utilisation de la programmation de léger car si le premier mouvement initié par des productions axées sur le concept, comme celles de Sol LeWitt, a certainement permis de déléguer une part d'autonomie à la forme produite, le second ne la lui retire pas vraiment. En effet la phase de réalisation se rapproche de l'auteur et de ses instructions sans pour autant devenir l'œuvre de l'auteur car c'est toujours l'ordinateur qui calcule les formes.

Casey Reas dévoile avec {Software} Structures en s'appuyant sur le travail de Sol LeWitt des caractéristiques similaires et opposées entre la programmation et la notation verbale. La similitude est cette prise de distance avec le produit final, la forme, grâce à la prise en compte de l'aspect conceptuel en tant qu'instructions. Et l'opposition entre la notation informatique et verbale est cette part activable du code informatique qui rapproche le programme et l'auteur du produit. En cela la programation contient un paradoxe au regard bilatéral qui oppose les notions de concept et de forme.

Pour approcher un travail qui interroge directement le design, l'installation d'affiches génératives <u>Posterwall</u> est un exemple frappant par sa vision radicale du sujet. Le dispositif du studio hollandais Lust projette des affiches aux formes variées conçues automatiquement à partir d'événements de l'actualité provenant de sites web. Le studio décrit leur intention:

Ce projet tente de donner un aperçu de la direction que le design graphique pourrait prendre dans le futur, tout en posant la question: avons-nous encore besoin de designers graphiques?

La question est légitime au regard du fossé qui se creuse entre l'auteur et la réalisation. À travers sa question, Lust considère implicitement que le designer graphique s'occupe plus de créer des produits que des programmes capables de les produire. Cependant, comme le souligne Stdin dans leur article, le champ des

possibles des affiches générées par l'installation est loin

Alexandre Leray et Stéphanie Vilayphiou, Écrire le design. Vers une culture du code, Back Cover #4, B42, 2011, p. 40 d'être infini, il reste restreint dans les possibilités programmées par les membres de Lust, en cela elles restent designées par Lust. Ce sont toujours eux les auteurs de ces affiches en prenant même en compte ce glissement de la position de l'auteur par rapport à l'objet final qui s'applique également au design graphique.

Stdin ajoute avec défi:

De plus, le code [...] n'a pas été remanié depuis la publication du projet. Si ce programme reflète les tendances futures du design graphique, comment se met-il à jour?

Mais comme on l'a vu avec les algorithmes d'évolution génétique, ce n'est peut-être qu'une question de temps avant que ce type d'algorithme soit investi par les designers graphiques selon les évolutions futures du métier.

PosterWall, Lust, Walker Art Center, 2011





Conclusion 38

La nature précède l'humanité, en cela elle n'est que chaos sauvage, un déluge de formes. Au fil de l'histoire, les sciences ont emmagasiné un lourd bagage théorique par la volonté de comprendre la nature, de la rationaliser.

Les modèles scientifiques sont toujours plus englobants et l'homme met en place des techniques toujours plus efficaces pour la maîtriser. Au point qu'aujourd'hui il n'est plus question pour l'Homme de sa perte dans la nature mais de perte de la nature. Il est donc légitime de se demander quelles traces de la nature y aurait-il au sein de nos techniques?

Les techniques numériques actuelles permettent de simuler les modèles scientifiques et de les manipuler avec aisance pour former des expériences sensibles. Le designer se retrouve en prise avec des outils aux paramètres très vastes: les variations de formes sous-tendues par les algorithmes qu'il a à sa disposition semblent désormais quasi infinies.

Une partie des productions humaines et en premier lieu le software semble se diriger vers une autonomie de plus en plus certaine vis à vis de l'Homme. Les formes émergent des programmes pour s'imprimer dans l'espace. Avec les avancées en vie artificielle, on a la possibilité de leur donner un pouvoir d'adaptation, de réplication voire d'évolution. Jean-Jacques Hublin voit dans ces avancées actuelles la continuité progressive de l'externalisation des processus biologiques de l'Homme, chose qui s'ancre dans l'évolution des techniques et des cultures depuis les pré-

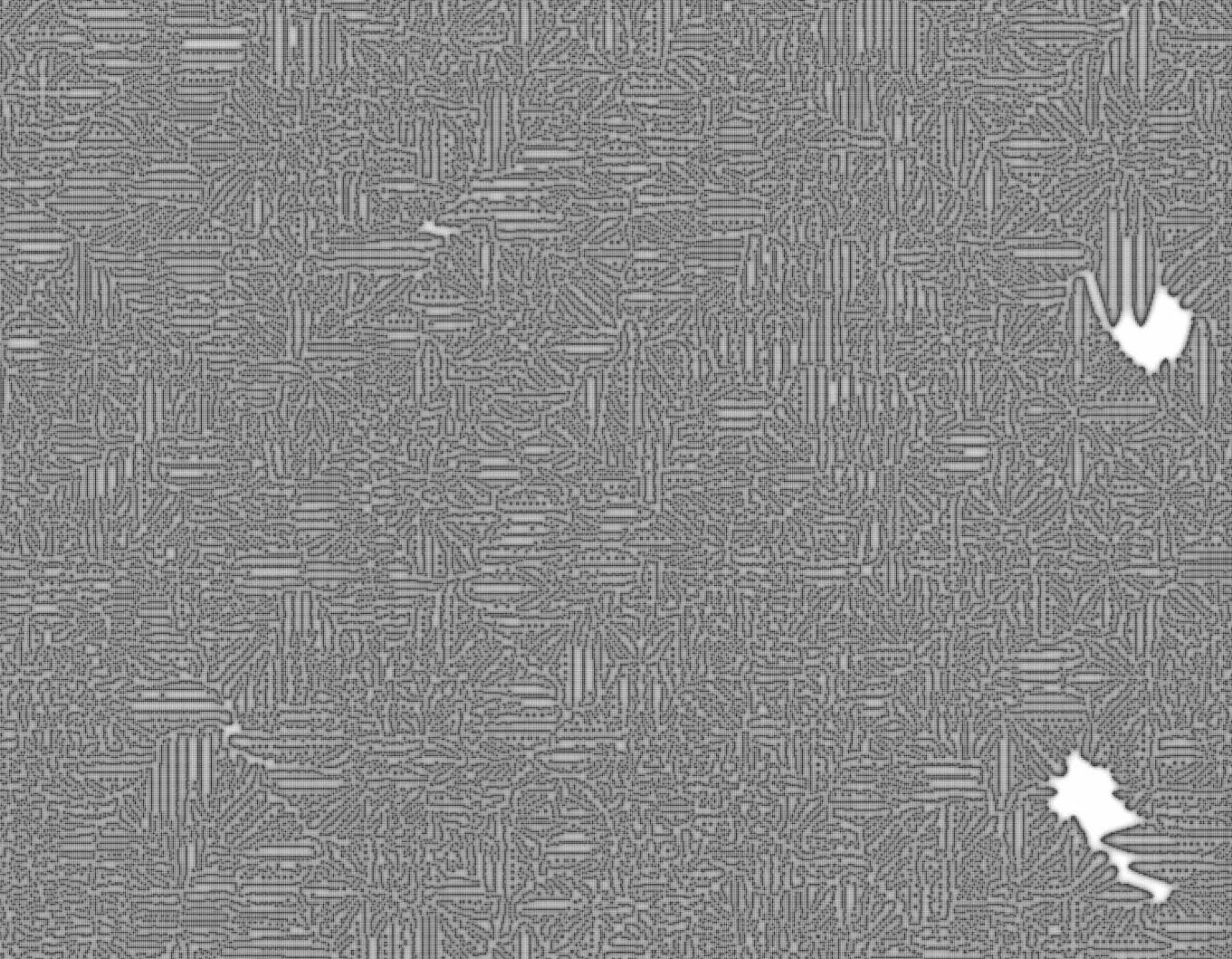
Jean-Jacques Hublin, Biologie de la culture, Collège de France, 2014

Cette question du retrait du designer avancée par Lust et les doutes émis par Stdin rejoint le débat entre l'interprétation forte et faible de la vie artificielle. Les affiches produites sont-elles réellement designées par le programme ou n'est-ce qu'une apparence, comme si elles semblaient avoir été designée?

misses de l'humanité.

Et plus largement, l'interrogation de Lust renvois au concept de la singularité technologique, selon lequel à partir d'un stade hypothétique de l'évolution technique, le progrès ne serait plus que l'œuvre d'intelligences artificielles, elles-mêmes en constante progression.

Comment agencer le programmé et le sauvage dans un design prenant en compte cette part d'autonomie de la production de l'objet ? Pour un design algorithmique, quelle maîtrise faudrait-il garder au détriment de cette autonomie et quel champ d'action faudrait-il lui laisser pour investir ses possibilités propres ?



Bibliographie

Vilém Flusser, Petite philosophie du design, Circé, 2002

Gary William Flake, The Computational Beauty of Nature, MIT Press, 1998

Frédéric Dufoing, L'écologie radicale, Infolio, 2012

Jean-Marc Lévy-Leblond, La symétrie aujourd'hui, Point Sciences, 1989

Paul Bourgine, Morphogenèse : L'origine des formes, Belin, 2006

Karl Gerstner, Designing Programmes, Lars Müller, 2007

Jean-Philippe Rennard, Vie artificielle. Où la biologie rencontre l'informatique, Vuibert, 2002

Anne M. Burns, Recursion in Nature, Mathematics and Art, Long Island University, 2006

Casey Reas, Chandler McWilliams, Lust, Form+Code, Princeton Architectual Press, 2010

Douglas Rushkoff, Program or be Programmed. Ten Commands for a Digital Age, OR Books, 2010

Herbert W. Franke, The New Visual language: The Influence of Computer Graphics on Art and Society, 1985

Vassily Kandinsky, Point et ligne sur plan, Gallimard, 1926

Thomas Dreher, Conceptual Art and Software Art: Notations, Algorithms and Codes, 2007

Stdin, Écrire le design. Vers une culture du code, Back cover n°4, hiver/printemps 2011

Peter Sloterdijk, La Domestication de l'Étre, Mille et une nuits, 2000

Conférences

Onformative, Cedric Kiefer, <u>Generative Design</u>, Campus Party, 2012

Janine Benyus, <u>Biomimicry in action</u>, TED, 2009

Bernard-chazelle, <u>Les incroyables algorithmes naturels</u>, Collège de France, 2012

Jean-Jacques Hublin, <u>Biologie de la culture</u>, Collège de France, 2014

Kevin Slavin, How algorithms shape our world, TED, 2011

Mémoires

Maxime Foisseau, <u>Du vivant à la simulation</u>, ESAD Valence, 2013

Kévin Donnot, <u>Outils numériques et design graphique</u>, EESAB Rennes, 2010

dont un extrait est republié dans Graphisme en France, code<> outils <> design, CNAP, 2012

Projets

Blog créé dans le contexte de ma recherche pour rassembler les artistes et designers qui ont un travail en lien avec le sujet : **naturalgorithm.tumblr.com**

Mémoire réalisé dans le cadre de mon Diplôme National Supérieur d'expression Plastique à l'École Supérieure d'Art et Design de Valence. Je remercie tous ceux avec qui j'ai pu échanger tout au long de ma recherche, tous ceux qui m'ont aidé d'une manière ou d'une autre à faire avancer cet écrit par de nombreuses remarques et discussions.

Ivan Murit

ivan@mailoo.org