

Usages et possibilités créatives de l'écriture du code: des langages de programmation à une langue poétique

Sujet

Les textes les plus importants de notre époque sont aussi les plus invisibles. Les centaines de milliards de lignes de code qui constituent le code source de nos systèmes techniques et, de plus en plus, sociaux demeurent encore largement exclus des champs de l'analyse esthétique et littéraire. Les langages de programmation constituent pourtant un domaine inédit de l'expression humaine, puisqu'il s'agit de la seule famille de langages pouvant être exécutés par une machine. Ceux-ci existent de manière ambiguë entre la machine computationnelle et l'individu, compréhensibles tant par l'un que par l'autre, incorporant l'exactitude du premier, et l'ambivalence du second. En tant qu'ensemble de signes et de structures qui permettent de décrire une série d'instructions précises devant être exécutées par une machine, on pourrait bien considérer ces langages de manière purement matérialiste —sous la forme de changements de courant électrique. C'est précisément cette nécessité de donner un *sens*, une *utilité* à ces variations électriques abstraites qui font des langages de programmation un moyen d'expression littéraire particulièrement contraignants —et pourtant, c'est bien par cette contrainte que des phénomènes esthétiques se manifestent, dans la lignée directe des recherches en art moderne du XX^e siècle.

Je propose alors d'étudier la pratique du code, dans sa lecture que dans son écriture, en tant qu'activité créative et esthétique. Les myriades de langages de programmation sont des lieux d'expression bien particulière; au-delà d'une simple nécessité d'efficacité, les questions de structure, de style, d'élégance et de beauté sont indissociables des discours qui se forment autour de l'écriture du code, et révèlent l'existence de préoccupations autour des notions du beau et de l'utile bien spécifiques à cette manière de faire. Dans la lignée du champ naissant des *software studies*, il s'agit d'examiner non pas le code en action, mais le code en rédaction, sous sa forme écrite, et non électrique. Il est en effet établi que ces nouveaux types d'écriture sous-tendent des systèmes numériques (logiciels) qui eux-mêmes permettent de nouvelles pratiques poétiques, prenant en compte les spéci-

ficités de la computation (e.g. procéduralité, spatialité, interactivité) afin d’explorer des possibilités nouvelles d’expressions artistiques. Ces pratiques regroupent aussi bien les fictions interactives et les automates de conversation, que des poèmes générés automatiquement par des systèmes de règles, qui ne sont pas sans rappeler les pratiques de groupes tels que les poèmes Dada, les calligrammes d’Apollinaire, les romans de l’OuLiPo ou encore les instructions du Fluxus. Ces spécificités touchent donc à l’utilisation de logiciels comme médium esthétique, et s’inscrivent dans une certaine tradition de l’écriture contemporaine, mais situent cette esthétique au niveau du *résultat* et non du *procédé*. En effet, il s’agit dans la plupart des cas de l’appréciation de l’objet créé par le programme, sans toujours prendre en compte le fait que l’apparition-même de ce résultat implique l’exécution, la disparition, du code —et donc l’absence quasi-totale de recherche artistique dans ce domaine. Il manque encore néanmoins une investigation rigoureuse des langages de programmation en tant que formes d’écritures.

Ce projet s’inscrit dans la continuité de mon travail de recherche appliquée lors de mon Master of Fine Arts à la New York University sous la direction de Mr. Bennett Foddy, ainsi que dans celle de ma carrière d’artiste-enseignant en art digital. Depuis, j’ai aussi bien communiqué autour de ces recherches lors de festivals artistiques (*Miracle Marathon* à la *Serpentine Gallery*), de symposia académiques (*Canadian Literature Symposium*) ou de revues artistiques (*NYU ITP Adjacent Possible*), que produit des oeuvres d’art à partir de l’écriture du code (notamment un des rares recueils de *code poetry* publiés, *Coroutines* chez *OfficialFan.Club* à New York). Au cours de cette carrière, j’ai cherché à mettre l’accent sur la tension unique entre l’objectivité d’un langage codé dit “universel” et les multiplicités de pratiques et tactiques propres au processus de l’écriture humaine tels qu’elles se manifestent au travers de langages de programmation, tant dans la rédaction de systèmes numériques commerciaux, scientifiques ou artistiques. Les choix particuliers de formulation, de dénomination et de structuration font de l’écriture du code un objet d’étude aussi pertinent que les actions des logiciels qu’ils sous-tendent, en tant que lieu où peuvent s’exprimer les personnalités et subjectivités de ceux et celles qui l’écrivent.

Notre approche se situe précisément au moment du développement des arts numériques, et de l’accélération de l’apprentissage de langages de programmation à différents niveaux des systèmes éducatifs. Ces nouveaux champs, tels que les arts numériques et les humanités numériques, manifestent une compréhension du computationnel comme un ensemble de possibilités pouvant être appréhendé en-dehors du seul domaine de l’ingénierie. Il existe bien un ensemble de recherches et d’études, principalement issues d’universités américaines, allemandes et néerlandaises, qui posent la matérialité des objets digitaux comme objets d’études. Cette matérialité, régulièrement ignorée par les effets de surface des interfaces de nos systèmes computationnels, est souvent rassemblée sous la notion générale du code. Là encore, cette notion de code demeure pourtant trop rarement re-

liée à ses instantiations spécifiques, à son existence concrète à travers les langages de programmation. Ce sont en effet ces langages spécifiques, et les textes aux moyens desquels ils sont écrits qui constituent l'objet de notre projet de recherche

Dans cette trajectoire, nous proposons donc d'approfondir ces nouvelles approches, et de considérer la matérialité de ces nouveaux systèmes dans leurs manifestations à travers les langages de programmation, et donc d'envisager ces langages d'un point de vue esthétique, philosophique et sémantique. Ces corpus possédant des particularités propres, autant dans leurs champs lexicaux que dans leurs structures syntactiques ou grammaticales, ils peuvent alors se prêter à des analyses à l'aide des outils des théories littéraires, esthétiques ou sémantiques, telles que le "*terministic screen*" de Kenneth Burke, les "*Sprachspiele*" de Ludwig Wittgenstein ou les "*fonctions du langage*" de Jakobson. Dans ce contexte, il s'agira notamment d'exhumer un ensemble de pratiques créatives liées à la spécificité de l'écriture en langages de programmation.

La différence ontologique entre ces langages machines et les langages humains se situe de plus dans leur processus de création, dans le fait que leur syntaxe et champ lexical sont conçus par un nombre connu d'individus, documentés, justifiés et publiés, puis mis-à-jour, modifiés ou abandonnés. Il semble donc nécessaire de prendre en compte ce processus de conception particulier au sein d'une étude de l'esthétique du code, d'en examiner les possibilités, contraintes, et décisions de conception, afin d'élucider comment ce type de spécification peut influencer sur les pratiques artistiques qui en découlent.

Cette nature bien particulière du langage de programmation permet alors de formuler l'hypothèse d'une large variété de pratiques expressives, reflétant celles qui existent déjà dans la littérature en langages "humains", autant à un niveau microscopique qu'à un niveau macroscopique. D'une part, les unités de bases qui constituent tout langage de programmation (e.g. variable, type, fonction, conditions, boucle, *try/catch*, arguments) seront examinées comme potentielles figures de style, motifs ou symboles en tant qu'elles portent en elles des représentations de concepts abstraits, initialement issus du domaine de la computation, et pouvant être rassemblées en une sorte de dictionnaire de rhétorique, explicitant chacune des potentialités spécifiques supportant de manière plus ou moins explicite l'expression de certaines idées créatives. De l'autre, il s'agirait de s'intéresser aux paradigmes généraux qui informent les différentes approches de conception et d'écriture de code et de les replacer autant dans un contexte stylistique que dans un contexte culturel. Ici, les questions de linéarité, de division, de manières descriptives et des paradigmes d'écriture peuvent initier une réflexion prolifique sur les propriétés poétiques —au sens d'une cosmogonie— au cours des lignes de code, examinée en termes de trames, d'arcs et des schémas narratifs.

Approche analytique et artistique

Mon projet de recherche s’articule donc autour des possibilités esthétiques des langages de programmation, en tant qu’ensembles conçus spécifiquement pour une fonction particulière et clairement définie, mais supportant pourtant déjà un usage artistique et créatif, de manière détournée, périphérique. Il s’agit alors de se demander comment les langages de programmation permettent des écritures poétiques. Quelles sont les places respectives du fonctionnel et du superflu, du technique et du culturel, dans une esthétique du code? Comment le cadre des théories esthétiques peut-il offrir une nouvelle perspective sur ces écritures du code? Enfin, comment ces propriétés esthétiques pourraient-elles être influencées par le processus-même de conception d’un langage de programmation?

Afin de répondre à ces questionnements, ce travail compte se baser sur une approche triple, d’abord analytique, puis historique, et artistique. Une des raisons principales de cette approche est l’absence de corpus de textes existants centrés sur les usages créatifs du code. Il s’agira avant tout de réunir des textes encore disparates, et d’établir une typologie des pratiques créatives existantes du code. Ce corpus sera composé de textes de code source sans limitation a priori quant au contexte de leur écriture. Ces textes pourront être issus du monde de l’ingénierie (les commentaires des *graduate students* citant les *Rolling Stones* lors de l’enclenchement de la procédure d’ignition du module d’alunissage d’*Apollo IX*), du monde du divertissement (les noms de variables à connotation fortement misogynes du code source du jeu vidéo *Dead Island*), des mondes amateurs (les concours d’écriture de code illisible de l’*Obfuscated C Code Contest*) et du monde artistique (les *code poems* partagé sur les plateformes en ligne de la communauté des *PerlMonks*). Le processus de constitution de ce corpus sera donc concomitant de leur analyse, sans se limiter à des bornes sociologiques ou historiques particulières. Cette approche large nous permettra d’identifier les grandes lignes des pratiques créatives de l’écriture et de la lecture du code, prenant pleinement en compte la dimension ludique de l’écriture du code, ainsi que la porosité des domaines de l’amateurisme et du professionnalisme dans ces pratiques. Toutefois, puisque ce travail s’ancre en partie dans une tradition de philosophie esthétique analytique, la limite de ce corpus se trouvera articulée autour de la relation entre productivité et gratuité, et mettra donc bien l’accent sur le code écrit plutôt que sur le code exécuté. Si cela exclut toute acte de créativité au sein de l’opération de logiciel, il faudra bien prendre en compte les langages de programmation en eux-mêmes, tant les langages standards les plus utilisés aujourd’hui (C++, Python, Java), mais aussi les pratiques des langages ésotériques.

En effet, l’examen des possibilités esthétiques des langages de programmation révèlent que ces dernières se posent bien en dégradé le long de notre limite de définition, de l’absolument fonctionnel à l’absolument inutile, et c’est autour de l’exploration de cette inutilité que s’organisent les pratiques contemporaines autour de ces langages, au moment où ils sont lus et interprétés par des humains, et non par des machines. Celles-ci se regroupent autour de la conception de nouveaux langages comme expression artistique —ésotériques—, au sein de communautés restreintes mais

actives. La relation entre écriture créative *en* langage de programmation et écriture créative *de* langage de programmation se retrouvera bien autour de la dichotomie déjà identifiée de l'utile/inutile. Ces langages ésotériques, tels que le *Piet*, *Brainfuck*, ou *TrumpScript*, feront alors également partie de notre corpus puisque leur conception constitue en elle-même une pratique créative du code.

Au cours de la constitution et de l'analyse de ce corpus, il sera nécessaire de d'introduire un point de vue historique. Malgré l'absence d'histoire compréhensive des langages de programmation, nous projetons d'ajouter à ces dimensions typologiques du corpus une dimension contextuelle. Nous faisons ici l'hypothèse que les circonstances économiques, techniques, sociales et culturelles de la conception et l'évolution de la popularité d'un certain langage, jouent un rôle dans la mise en place et le développement de ces pratiques, et que les changements de paradigmes de programmation (programmation fonctionnelle, orientée-objet, etc.) sont contemporains de changements sociétaux qui se reflètent de manière subséquente dans une écriture créative et stylistique. Nous examinerons par exemple la relation entre la constitution des communautés de poètes en Perl et le fait que ce langage soit le seul à être distribué sous License Artistique. Une certaine pratique du C serait, elle, replacée dans son contexte d'infrastructure de bas niveau de tous nos systèmes digitaux, restreignant alors peut-être ses usages créatifs à la stylisation du code source en calligrammes modernes. Quant à des langages comme le JavaScript, leur popularité tient peut-être moins à la robustesse de leur conception technique, mais bien à une tolérance à l'erreur bénéfique aux communautés amateurs. Ce sont ces contextes historiques qui ne pourront donc pas être ignorés lors de l'analyse des textes de code source.

Enfin, la nouveauté du domaine d'études lui-même et la limitation du corpus existant appelle une approche artistique. La recherche orientée sur la pratique est une méthodologie qui nous permettrait d'élucider, au fur et à mesure de la création, de nouvelles pistes de recherche; puisque ce projet se veut centré sur la concrétisation des possibilités créatives du code, il nous semble nécessaire de contribuer nous-même à ces pratiques, afin de confronter nos recherches scientifiques à l'activité artistique qu'est l'écriture du code. Ma pratique artistique personnelle s'organise autour des ces questions, et inclue un des rares recueils publiés de *code poetry* —*Coroutines*. Cette approche artistique, aura pour but de projeter notre recherche dans l'espace spéculatif de l'écriture créative du code. L'hypothèse, ici, serait de considérer l'évolution des langages de programmation, depuis les cartes poinçonnées du métier à tisser Jacquard jusqu'à la quasi-omniprésence du langage Python dans les salles de classe de collège et de lycée, à celle de l'écriture des langages humains au cours des cinq derniers millénaires. Ces derniers ayant commencé par représenter uniquement des quantités de marchandises, puis de monnaie, puis des édits royaux, puis des législations divines, la place de l'écriture humaine comme moyen d'expression poétique n'est pas immédiatement concomitante de son apparition en tant qu'outil. L'usage actuel des langages de programmations

n'empêche donc pas la possibilité de les imaginer, et de les pratiquer comme une nouvelle manière d'exprimer une certaine esthétique, se rapprochant d'avantage du roman et de la poésie que du protocole ou de la copie carbone. Enfin, cette approche artistique permettrait de générer non seulement d'avantage de questions de recherches, mais également d'autres sources premières —poèmes, nouvelles, etc.— pour des travaux futurs sur ce sujet.

Méthodologies et sources

Le sujet de ce projet de recherche étant directement relié aux technologies numériques de communication, la majorité de mes sources premières ont été créées, distribuées et archivées en ligne. En effet, l'écriture du code se retrouve étroitement liée à la philosophie de l'*open source*, du partage radical du résultat et des procédés liés au code, philosophie du partage que le programmeur Linux John Hall attribue d'ailleurs à une satisfaction et fierté esthétique. Il est donc facile d'accéder aux travaux artistiques d'amateurs éclairés au sein de sites bien définis, tels que la liste de courrier internet *nettime* hébergeant les travaux de Mezangelle et d'Alan Sondheim, les forums des Perl-Monks ou les réseaux d'art digital, notamment *runme.org*. Cette éthique de l'*open source* facilite également l'accès au code source de systèmes plus élaborés, tels que le kernel de Linux ou des parties du code source de Windows, nous permettant alors d'examiner les pratiques créatives qui ont lieu au sein même de systèmes hautement productifs et commerciaux. Enfin, l'augmentation de la popularité de sites de systèmes de contrôle de version, tels que GitHub, GitLab ou encore Bit-Bucket, rendent public les projets professionnels et individuels de chacun. Il faudra donc prendre ces sites comme lieu d'exploration et de récupération de textes pouvant affiner notre travail de typologie et d'analyse, une fois que celui-ci se sera basé sur des pratiques auto-définies comme artistiques des réseaux précédemment cités. J'ai déjà effectué en partie ce travail de recherche et de rassemblement lors de la conception de deux cours que j'ai donné à la New York University, notamment *Politics of Code* (IM-UH 3110) et *Software Art: Text* (IM-UH 2116).

Il existe également un ensemble très réduit de recueils publiés où les lignes de code constituent l'objet principal de l'attention, tels que les *{code poems}* d'Ishaac Bertram, *#!* de Nick Montfort, *Weird Languages* de Daniel Temkin ou encore *Algorithms* de Los Pequeños Glazier. Bien que ces recueils relevassent d'avantage du domaine du livre d'art ou du *zine*, ma fréquentation constante de ces milieux m'a permis d'acquérir la plupart de ces ouvrages et d'entrer en contact directement avec les auteurs au cours de mon séjour de trois ans à New York. Je compte donc aussi sur ces relations pour me diriger vers d'autres parties de cette scène artistique, et m'indiquer des aspects des pratiques créatives du code que j'aurai pu survoler jusqu'ici.

Le deuxième volet de cette recherche, celui de la recherche par la production artistique, sera donc basé sur un protocole opérant à plusieurs niveaux: intra-linguistique, inter-linguistique, et méta-linguistique.

Le niveau intra-linguistique consistera à écrire et ré-écrire des textes (poèmes, nouvelles, algorithmes, ou encore applications-types du genre serveur HTTP) dans un même langage —le langage Python, de par sa versatilité et sa popularité—, afin de mettre en exergue les différences stylistiques dans l’écriture d’un même programme, pratique autant littéraire qu’informatique. Il s’agit donc d’explorer les différentes directions dans lesquelles peut se manifester la subjectivité de l’écriture d’un programme, et comment cette subjectivité peut changer drastiquement sa réception par un humain affecter le résultat final.

Au niveau inter-linguistique, un même programme est ré-écrit dans différents langages. La pratique de traduire une même structure conceptuelle et de la retranscrire dans une différente implémentation du “langage universel” nous permettra alors de déplacer notre attention depuis les structures macroscopiques de style, de formulation et de structure vers les éléments microscopiques des types de variables, des systèmes de classe, du rapport entre interprétation et compilation ainsi que de la particularité des écosystèmes respectifs de ces langages (qui y contribue, quels sont les modules et librairies qui complètent le langage de base, comment se fait le processus d’apprentissage, quelles sont les valeurs des communautés qui les maintiennent, etc.). Nous aurons alors développé une série d’objets artistiques illustrant d’une part les différentes typologies établies dans la partie analytique, mais ayant également la possibilité d’ouvrir d’autres pistes de réflexions à travers le processus-même de création.

Enfin, le niveau linguistique consistera à déplacer notre attention non plus sur l’écriture de code à travers un langage de programmation, mais sur l’écriture d’un langage de programmation en code. La conception et l’implémentation d’un langage de programmation complet se tient en effet dans la suite logique d’un travail de recherche sur l’usage créatif du code puisque la nature artificielle de ces langages est elle-même un lieu d’expérimentation unique, et ontologiquement premier à tout usage créatif de code. Notre travail de conception d’un nouveau langage de programmation serait alors une application pratique de la recherche effectuée jusqu’alors et chercherait à faciliter les écritures créatives de code, de la même manière que des langages tels que le langage Rust met l’accent sur la sûreté des fils d’exécution, ou que le langage Perl met l’accent sur le traitement de séquences de texte.

Ce travail séquentiel, d’abord de constitution de corpus, puis de leur analyse scientifique et enfin de création artistique sera donc notre manière d’appréhender un champ de recherche encore nascent de manière inter-disciplinaire.

États des lieux de la recherche

L'approche proposée par ce travail de thèse s'inscrit dans des champs de recherches multiples, convoquant différentes disciplines (philosophie esthétique, études des médias, informatique et linguistique). En cela, ce projet se fonde sur une large production théorique, mais se positionne de manière nouvelle en les faisant dialoguer sur le lieu du code source.

La principale bibliographie concernant les langages de programmation se situe tout d'abord au sein du domaine de la science informatique. Ces ouvrages sont principalement concernés par la stylistique fonctionnelle de l'écriture de programmes, et non par l'apprentissage de la programmation en elle-même, adressés avant tout à ceux et celles qui savent déjà écrire du code et sont donc concerné.e.s par des préoccupations qui se rapprochent du domaine esthétique. Bien que constituant des fondamentaux du champ du génie logiciel, notamment *Structures and Interpretations of Computer Programs* et *The Art Of Computer Programming*, ces ouvrages se distinguent par leur ton parfois peu académique, et mettent l'accent sur la tension entre cohérence scientifique et pratique artisanale de la programmation. Cette partie de nos références nous permettra de baser notre recherche sur ce que devrait être l'écriture du code d'un point de vue strictement technique, et donc de le poser en contraste des écritures créatives qui s'en détachent.

L'étude du code en dehors du champ scientifique et du génie prend d'abord place au sein de l'études des médias. Si l'impact des systèmes codifiés à différents niveaux de nos sociétés est en plein essor, l'examen rapproché des bases matérielles et littéraires de ces systèmes sont moins nombreuses, même si ayant fortement influencé les travaux subséquents. Ainsi, les recherches de Lev Manovich, N. Katherine Hayles, Wendy Hui Kyong Chun en Amérique ou encore Friedrich Kittler et Siegfried Zielinski en Europe, traitent des implications politiques et cosmogoniques du numérique en tant que système d'écriture radicalement différent de l'écriture humaine. Le domaine des *software studies*, se déplaçant des universités américaines vers les universités européennes, notamment à travers le travail doctoral d'Anthony Masure en France, analyse à travers ce nouveau prisme les objets codés en tant qu'artefacts culturels. C'est en partie dans cette trajectoire que ce projet de thèse compte s'inscrire.

La pratique artistique digitale est également un champ d'étude en pleine expansion, autant d'un point de vue esthétique et littéraire qu'historique et culturel. L'oeuvre d'art assistée par ordinateur, de plus en plus populaire, et donc de plus en plus théorisée, limite pourtant son champ d'étude à l'interface et à l'action —c'est à dire au logiciel. C'est dans ce domaine que s'inscrivent les travaux de Philippe Bootz et de Serge Bouchardon, qui demeurent cependant des ressources précieuses pour avoir établies un travail de fond en termes de poésie numérique, technotexte et esthétique.

tique cybernétique. La recherche autour de la poésie digitale regroupe toutes les formes textuelles et littéraires qui prennent l'ordinateur en tant que médium différent de l'impression, et comprennent la poésie générative, les poèmes collaboratifs, la poésie animée (inspirée de la poésie concrète), les *codeworks* ou la fiction interactive. Les analyses qui se portent là-dessus se focalisent sur les propriétés uniques de systèmes computationnels lorsqu'il s'agit de permettre la création d'objets artistiques, dans la lignée de Jack Burnham, Janet Murray et de Charles Hartman.

Il faut enfin prendre en compte le fait que seule une petite fraction de ce champ de la poésie digitale prend pour véritable objet d'étude le code en lui-même, et donc il y a en fin de compte très peu de littérature sur cet aspect spécifique. Les travaux de Camille Paloque-Bergès sur ces thèmes-ci s'orientent d'avantage sur la circulation en réseau des écritures codées que sur leur processus même d'écriture, tout comme le travail doctoral de Maurice Joseph Black, qui met en exergue les traditions spécifiquement littéraires dans lesquels s'inscrivent les programmeurs. Florian Cramer, d'un côté et Geoff Cox et Alex McLean ont notamment réalisé deux études incontournables sur le sujet, sur lesquels je compte m'appuyer en mettant d'avantage l'accent sur l'esthétique propre de l'écriture de code source plutôt que sur leur signification culturelle au sens large.

Pour ce faire, nous nous appuierons sur une tradition de philosophie esthétique contemporaine, en ce qu'elle relie souvent sémantique, linguistique et esthétique. Les travaux de Ludwig Wittgenstein, de Nelson Goodman et de Kenneth Burke et dans une certaine mesure les recherches sur la poésie de Roman Jakobson nous permettront de qualifier les relations entre symboles, interprétations et contextes et nous seront utiles afin d'inscrire les pratiques analysées dans une véritable approche d'analyse de l'art et du langage. Ces approches de l'art ne pourront se dispenser d'inclure les ouvrages de l'esthétique moderne et post-modernes, notamment les travaux de l'OuLiPo et la poésie $L=A=N=G=U=A=G=E$.

Si la nature interdisciplinaire de notre projet peut donc nous permettre de nous appuyer sur une bibliographie vaste, la spécificité de notre approche autour des modalités esthétiques de l'écriture du code source délimite clairement notre objet d'étude. La spécificité de notre projet est donc de puiser dans ces multiples approches afin de mettre en avant, et en pratique, les manifestations pratiques de la complexité culturelle, sémantique et esthétique de l'écriture du code.

Bibliographie indicative

Informatique

- ABELSON, H. SUSSMAN J.G., SUSSMAN, J., *Structure and Interpretation of Computer Programs*, Boston, 1996.
- BIANCUZZI, F., *Masterminds of Programming*, New York, 2009.
- FRIEDMAN, D., P., MITCHELL, W., *Essentials of Programming Languages*, Boston, 2014.
- INCE, D., C., *Mechanical Intelligence*, Vol 1 (Collected Works of Alan Turing), Amsterdam, 1992.
- KERNIGHAN, D. , RICHIE D. *C Programming Language*, New Jersey, 1988.
- KNUTH, D., *The Art of Computer Programming*,
- SEBESTA, R., *Concepts of Programming Languages*

Études des médias

- BOGOST, I. *The Rhetoric of Video Games in The Ecology of Games: Connecting Youth, Games and Learning*, éd. Salen, K., Boston, 2008.
- CHUN, W., *Programmed Visions: Software and Memory*, Boston, 2011.
- COX, G. MCLEAN, A., *Speaking Code: Coding as Aesthetic and Political Expression*, Boston, 2012.
- FULLER, M., *Behind the Blip: Essays on the Culture of Software*
- GALLOWAY, A. *Gaming: Essays on Algorithmic Cultures*, Minneapolis, 2006.
- HAYLES, N. K., *My Mother Was A Computer: Digital Subjects and Literary Texts*, Chicago, 2005.
- KITTLER, F. *Discourse/Systems 1800/1900*, Stanford, 1999.
- MANOVICH, L., *The Language of New Media*, Boston, 2001.
- MASURE, A., *Le Design des Programmes: Des Façons de Faire du Numérique*, Thèse soutenue en 2014.
- MCLUHAN, M., *Understanding Media*, Berkeley, 1994.
- MONTFORT, M., BAUDOUIN, P., BOGOST, I., DOUGLASS, J., MARINO, M. C., MATEAS, M., REAS, C., SAMPLE, M., VAWTER, N., *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*, Boston, 2012.
- SACK, W. *Software Arts*, Boston, 2018.
- VEE, A., *Coding Litteracy: How Programming is Changing Writing*, Boston, 2018.
- WARK, M., *A Hacker Manifesto*, Boston, 2004.
- ZIELINSKI, S., *Deep Time of the Media: Toward an Archaeology of Hearing and Seeing by Technical Means*, Boston, 2006.

Art Digital

CHOI, T., *Poetic Computation: A Reader*, <http://poeticcomputation.info/> consulté le 09/12/2018

CRAMER, F., *Words Made Flesh: Code, Culture, Imagination*, Rotterdam, 2005.

BLACK, M. J., *The Art of Code*, travail doctoral, Philadelphie, 2002.

BOOTZ, P., *Poésie Numérique: Du Cybertexte Aux Formes Programmées*, Paris, 2006.

BOUCHARDON, S., *La Valeur Heuristique de la Littérature Numérique*, 2014.

BURNHAM, J., HAACKE, H., *Esthétique des Systèmes*, QUINZ Emanuele (ed.) Dijon, 2015.

HARTMAN, C. O., *Virtual Muse: Experiments in Computer Poetry*, Middletown, 1996.

MURRAY, J., *Hamlet on The Holodeck*, Boston, 1998.

PALOQUE-BERGES, C., *Poétique des codes sur le réseau informatique*, Paris, 2009.

Esthétique & Linguistique

ADORNO, T. *Aesthetic Theory*, ed. ADORNO G. et TIEDERMANN R., trad. Hullot-Kentor, R., Continuum, 2004.

BURKE, K. *Language as Symbolic Action*, Oakland, 1966.

GOODMAN, N., *Language of Art: An Approach to a Theory of Symbols*, Cambridge, 1983.

JAKBSON R., *Questions de poétique*, 1973

RANCIERE, J. *Aisthesis : Scènes du régime esthétique de l'art*, Paris, 2012

WITTGENSTEIN, L., *Tractatus Logico-Philosophicus*, Paris.

WITTGENSTEIN, L., *Recherches Philosophiques*, Paris.