

# De la coopération à la compétition

*ou Comment le code informatique est devenu propriétaire*

Louis-Philippe Véronneau<sup>1</sup>

décembre 2018

**Abrégé :** Avec l'émergence de l'informatique comme discipline universitaire dans les années 1960 apparaît une nouvelle génération d'étudiant-e-s à qui on donne accès aux ordinateurs de l'université. Cette génération forme la première vague de *hackers*, une communauté dotée d'un *ethos* promouvant le partage et la curiosité – parfois au mépris de la loi. Le cadre juridique flou (voire inexistant) encadrant la propriété intellectuelle du code informatique faisant fonctionner les ordinateurs ainsi que les pratiques du milieu informatique de l'époque rendent possible la croissance de cette communauté qui met de l'avant la coopération comme moyen de production économique. En parallèle de l'émergence du mouvement *hackers*, les politiciens américains profitent de la mise à jour du *Copyright Act* pour réfléchir sur les questions entourant la propriété intellectuelle du code informatique. Après un rendez-vous manqué avec l'histoire en 1976, la *Commission on New Technological Uses of Copyrighted Works* (CONTU) est chargée par le Sénat de faire des recommandations en vue d'une modification du *copyright* concernant le code. Une décennie après le début des travaux de la CONTU, les tribunaux américains établissent finalement une jurisprudence forte sur cette question.

**Mots-clefs :** Copyright, code informatique, États-Unis, CONTU, hackers

**Nombre de mots :** 6969

---

<sup>1</sup> Étudiant à la maîtrise en sciences économiques à l'Université du Québec à Montréal, pavillon des sciences de la gestion, 315, rue Sainte-Catherine Est, Montréal (Québec), H2X 3X2, Canada, email : [louis-philippe@riseup.net](mailto:louis-philippe@riseup.net). Merci à Till Düppe, Adam Aberra, Alexis Bureau-Thibault, Frédéric Legault et Pier-Alexandre Nadeau-Voynaud pour leurs commentaires.

# De la coopération à la compétition

*ou Comment le code informatique est devenu propriétaire*

« When I first started in the field in the '60's, basically software was free. And we went in cycles. It became free, and then software manufacturers, in the need to expand their markets, pushed it in other directions. »

– Richard M. Stallman, *Free Software: Freedom and Cooperation*

## Introduction

Le progrès en informatique dans les quarante dernières années a eu un impact important sur nos sociétés. Réservés à ses débuts à la recherche militaire puis universitaire, les ordinateurs sont rapidement devenus des objets indispensables, tant dans la sphère professionnelle que dans la sphère privée. Alors qu'une partie écrasante de l'économie contemporaine repose maintenant directement ou indirectement sur ces ordinateurs, nous semblons souffrir d'un important manque de recul historique quand vient le temps de parler d'informatique. Pris dans l'incessante quête visant à renouveler toujours plus rapidement nos outils technologiques, les changements institutionnels engendrés par ces derniers sont trop souvent analysés *ex-post*. Et comment pourrait-on en faire autrement lorsque le slogan officiel de la *Silicon Valley* – « Move Fast and Break Things » (Blodget, 2009) – nie la pertinence même d'un regard historique dans ce domaine. Si un modèle de téléphone intelligent révolutionnaire il y a 5 ans est maintenant complètement désuet, pourquoi nous intéresserions-nous aux dinosaures technologiques des années 70 et aux débats institutionnels qui les ont vu naître ? Malgré ce que peuvent en dire les Jeff Bezos et les Elon Musk de ce monde, le passé est bien souvent porteur de conseils. Le retour

en force des *Crypto Wars*<sup>2</sup> - débat que l'on croyait complètement gagné par la communauté *hackers* en 1996 – en est une preuve flagrante.

Dans ce texte, nous nous intéresserons principalement à l'histoire des changements légaux entourant la propriété intellectuelle du code informatique aux États-Unis entre 1960 et 1990. Nous avons choisi de restreindre nos recherches aux États-Unis non seulement de par la complexité d'aborder de manière internationale un sujet aussi encastré dans les procédures légales, mais également à cause de la prépondérance historique des États-Unis dans ce domaine. Du projet ARPANET au quasi-monopole des compagnies américaines comme IBM et Apple en passant par l'établissement de l'*Internet Engineering Task Force* (IETF), les États-Unis ont toujours été à l'avant-garde des changements en informatique. Il est donc conséquent que l'histoire des droits de propriétés intellectuelles aux États-Unis ait également été précoce. Dans notre recherche, nous nous pencherons plus particulièrement sur l'évolution historique du *copyright* américain et des discussions entourant la volonté politique de faire du code informatique un bien économique comme les autres.

Notre travail se divisera en quatre temps. Nous aborderons tout d'abord le vide légal des années 60 & 70 ayant permis aux communautés *hackers* de proliférer. Nous analyserons dans un second temps la période allant de 1961 à 1976 et qui constitue les balbutiements du désir de régulation du code informatique de la part du législateur américain. Véritable tournant dans le cadre légal en ce qui concerne l'informatique, la troisième partie sera consacrée à l'étude des travaux de la *Commission on New Technological Uses of Copyrighted Works* (CONTU), allant de 1974 à 1978. Finalement, nous terminerons par l'analyse d'un jugement post-CONTU

---

2 Alors que le chiffrement robuste aux tentatives de déchiffrement par les gouvernements devient accessible à toutes et à tous par le partage de code informatique libre sur l'internet au début des années 1990, le gouvernement américain décide de considérer ces méthodes comme des armes et en criminalise le partage à l'extérieur de ses frontières. Après l'établissement d'un consensus fort opposé à ces *Export Laws* et une mobilisation importante dans l'industrie de l'informatique, cette loi est graduellement abolie entre 1996 et 2000. Pour en savoir plus sur les *Crypto Wars* des années 1990, voir le livre publié par Steven Levy en 2002 sur le sujet.

extrêmement important, soit la décision de la cour d'appel des États-Unis dans le jugement *Apple v. Franklin* de 1983.

## **Le vide légal des années 60 & 70 : une coopération naturelle et bénéfique**

Avec l'émergence des premiers ordinateurs sur les campus universitaires au début des années 1960, une génération de jeunes étudiant-e-s a soudainement accès à des machines qui étaient autrefois réservées à la recherche militaire. Les ordinateurs disponibles sont toutefois bien différents des ordinateurs modernes que nous connaissons.

En effet, si des entreprises commerciales comme Digital Equipment Corporation vendent des machines aux universités qui peuvent se permettre de les acheter, ces dernières ne sont généralement pas des produits clefs en main comme peuvent l'être un MacBook contemporain. Quand il existe, le code informatique qui vient avec ces machines est souvent très rudimentaire et parfois même complètement inutilisable. Dans ce contexte, une grande partie du code informatique qui roule sur ces machines est écrit par la communauté universitaire. Par exemple, le système d'exploitation utilisé en 1963 au MIT AI Lab – un centre de recherche à l'avant-garde dans le domaine informatique aux États-Unis – nommé *Compatible Time-Sharing System* implémente un des tous premiers systèmes de partage de temps et est à des années lumières du système d'exploitation fourni à l'achat de la machine (Williams 47-48).

C'est dans ce contexte qu'apparaissent les premières communautés connues sous le nom de *hackers*. Fortement motivés par leur amour des ordinateurs et leur curiosité sans bornes, les *hackers* créent des cercles sociaux régulés par un *ethos* très puissant qui guide leurs actions. Dans *Hackers: Heroes of the Computer Revolution*, Steven Levy résume les cinq points les plus importants de l'éthique *hacker*:

1. *Access to computers — and anything which might teach you something about the way the world works — should be unlimited and total. Always yield to the hands-on imperative!*
2. *All information should be free.*
3. *Mistrust authority — promote decentralization.*
4. *Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.*
5. *You can create art and beauty on a computer.*

Comme le note Gabriella Coleman – anthropologue spécialisée dans les communautés *hackers* – il est néanmoins réducteur de parler d’un seul code d’éthique applicable à l’ensemble des communautés *hackers*. L’*ethos hacker* est mieux présenté comme un code social flou plutôt qu’une série de règles rigides (19-20). Dans les communautés *hackers*, c’est le partage désintéressé qui permet de créer des liens et de bâtir un ensemble cohésif :

*If a program or software fix was good enough to solve your problems, it was good enough to solve somebody else’s problems. Why not share it out of a simple desire for good karma? (Williams 5)*

Dans les années 60 et 70, les *hackers* se regroupent souvent la nuit dans les universités pour utiliser les machines qui sont autrement occupées le jour par les professeur-e-s. De par leur curiosité et leur expérience technique importante, ces communautés prennent rapidement une place majeure dans la maintenance et l’innovation technique des machines. Par exemple, au MIT AI Lab, le système d’exploitation *Compatible Time-Sharing System* écrit par les professeur-e-s est jugé peu efficace et est réécrit par la communauté *hacker* de l’université. Ce nouveau système d’exploitation intitulé avec beaucoup d’humour *Incompatible Timesharing System* est tellement

supérieur que l'université décide de l'utiliser et remet les rênes des ordinateurs du AI Lab aux *hackers* (Williams 47-48).

Tout manquement à l'*ethos hacker* est reçu avec beaucoup de colère et d'incompréhension de la part de la communauté. Richard M. Stallman – *hacker* de renom au MIT AI Lab – explique ainsi lors d'une entrevue les problèmes causés par un don d'une imprimante de la part de Xerox au Lab. Si la machine semble à première vue merveilleuse – elle imprime mieux et plus rapidement – elle diffère de l'ancienne machine du fait que le code informatique qui la fait fonctionner n'est pas accessible. Contrairement à l'ancienne imprimante modifiée par les *hackers* pour régler certains problèmes flagrants, seul Xerox peut régler les problèmes de cette nouvelle imprimante. Stallman décide donc de rendre visite à une personne qui détient le code en question :

*And then I heard that somebody at Carnegie Mellon University had a copy of that software. So I was visiting there later, so I went to his office and I said, "Hi, I'm from MIT. Could I have a copy of the printer source code?" And he said "No, I promised not to give you a copy." [Laughter] I was stunned. I was so — I was angry, and I had no idea how I could do justice to it. All I could think of was to turn around on my heel and walk out of his room. Maybe I slammed the door. (Stallman)*

Ce manquement à l'*ethos hacker* n'est tout simplement pas compréhensible par Stallman, ce qui génère chez lui un sentiment de colère et d'injustice profond. Si Xerox n'est pas à même de régler les problèmes de ses machines, pourquoi refuse-t-elle que les *hackers* le fassent à sa place ?

Loin d'être un cas isolé, le MIT AI Lab représente très bien le modèle de production ouvert, décentralisé, libre – voire communiste – des communautés *hackers* qui domine largement le monde informatique des années 60 et 70.

## **1961 à 1976 : un désir de réviser le *Copyright Act***

En 1961, le gouvernement américain décide de lancer un processus de révision du *Copyright Act*. En effet, la dernière révision du *copyright* américain date alors de 1909 et n'offre donc pas de protection spécifique aux technologies issues de la Première et de la Seconde Guerre mondiale; la radio et la télévision – des sources importantes de litiges fondés sur la propriété intellectuelle – n'existaient tout simplement pas à l'époque (Ringer 2). Le comité sur le *Copyright Law Revision*, composé d'une quinzaine de membres du Sénat américain, poursuit ses travaux jusqu'en 1964. Le comité arrive cependant rapidement à un consensus sur la question du code informatique, et plus généralement sur le lien entre les ordinateurs et le *copyright* : le domaine est trop jeune et toute tentative de l'encadrer pourrait avoir des conséquences néfastes. Le comité affirme avoir peur d'encadrer les droits de propriété intellectuelle en informatique de manière trop rigide et de causer des torts durables dans ce domaine florissant (CONTU 82).

Il faut dire qu'à l'époque, il n'y a pas encore de jurisprudence établie sur le sujet : l'informatique est un domaine trop récent et les questions de droit d'auteur n'ont pas encore été traitées par les différentes cours américaines. Ce manque d'intérêt légal est également visible dans la provenance des mémoires et commentaires envoyés au comité. Lors du dépôt de son rapport final en 1965, sur un total de 14 mémoires provenant d'acteurs institutionnels, seul *General Electric* écrit pour demander au comité de traiter de la question du code informatique ("Copyright Law Revision, Part 5" 271).

Comme nous l'avons vu dans la partie précédente, le code informatique n'est alors tout simplement pas vu comme un bien économique à part entière. Marginal, sa production est encore réservée majoritairement à la recherche académique et aux *hackers*. Le code écrit par ces derniers est partagé librement et les rares compagnies comme IBM qui produisent du code informatique le



considèrent comme un bien secondaire, un peu comme on donnerait une cuillère de plastique à un client achetant un sorbet. Qui plus est, ce code est généralement accessible et modifiable par quiconque ayant accès aux machines.

C'est donc dans ce contexte que le comité décide de laisser volontairement tomber tout ce qui a trait à l'informatique et relègue cette question difficile aux cours de justice subséquentes :

*Are computer programs 'copyrighted works'? [...] The Register of Copyrights replied : 'I don't think there are any more difficult or important problems than the ones you have raised... We deliberately avoided any specific references to 'computers' [...]. We think that there are many developments that are going to come in the immediate future, and we think it safer to draft general language which can be interpreted by the courts to apply to particular usages'. (CONTU 82)*

En 1964<sup>3</sup> le *Copyright Office* américain – visiblement à l'avant-garde des débats à venir – publie un avis juridique affirmant la validité de l'application des lois sur le *copyright* au code informatique. L'*Office* ne tranche pas sur le fond dans cette question, mais – en accord avec sa politique sur le sujet<sup>4</sup> – décide que le code informatique doit être traité comme le serait une œuvre littéraire et ouvre la porte à l'attribution de *copyrights* en vertu de l'Act de 1909 (CONTU 82). C'est cependant loin d'être une victoire complète : pour que le *Copyright Office* accepte de protéger un programme informatique, une copie complète de ce dernier doit y être enregistré, et ce dans un format lisible par un être humain.

Si ces restrictions ne posent normalement pas entrave à la protection d'un livre ou d'une œuvre littéraire, il en est tout autrement pour le code informatique. En effet, une des raisons qui poussent les entreprises à se tourner vers le *copyright* pour protéger leur propriété intellectuelle

---

3 Le comité sur le *Copyright Law Revision* n'a alors toujours pas remis son rapport final au Sénat.

4 Historiquement, le *Copyright Office* a toujours décidé de trancher en la faveur de l'application restreinte du *copyright* lorsqu'il est saisi d'une question et n'arrive pas à émettre d'avis clair sur cette dernière (CONTU 82).

en informatique<sup>5</sup> est que ce dernier offre typiquement une protection immédiate et ne nécessitant pas d'attribution par un organisme externe. Comme le code informatique est sujet à des modifications rapides et constantes (*live patching, software updates, etc.*) et qu'il est souvent distribué sous forme binaire (et donc dans un format illisible par un être humain), la décision du *Copyright Office* ne sera appliquée que dans un très faible nombre de cas. En effet John Hersey – commissaire nommé par le Sénat lors de la *Commission on New Technological Uses of Copyrighted Works* (CONTU) – note que « de 1964, à partir du moment où le commissaire [au *copyright*] reçu le premier programme pour fin d'enregistrement, jusqu'au 1<sup>er</sup> janvier 1977, seuls 1 205 programmes ont été enregistrés (et deux compagnies, IBM et Burroughs, étaient responsables de 971 d'entre eux). »<sup>6</sup> (CONTU 34).

Suite au dépôt du rapport du comité sur le *Copyright Law Revision* en 1965, plusieurs projets de lois sont discutés au Sénat. De 1965 à 1966, ce dernier entend différents acteurs et actrices du milieu souhaitant que la réforme du *copyright* encadre l'utilisation des ordinateurs. Les débats ne tournent cependant pas directement autour de la question du code informatique, mais se concentrent plutôt sur la reproduction numérique d'œuvres littéraires, musicales et artistiques. D'un côté, les libraires souhaitent commencer à numériser et à photocopier certains contenus, alors que de l'autre, les grandes maisons d'édition ont peur de perdre des revenus provenant du droit d'auteur (CONTU 89-104).

Conséquent avec la recommandation du comité sur le *Copyright Law Revision*, le Sénat décide que l'informatique est une discipline trop jeune pour qu'il soit possible de se prononcer de manière éclairée et laisse le soin aux législateurs futurs de le faire. En 1967, le Sénat et la chambre des représentants (qui décide également de suivre les recommandations du comité sur la

---

5 Nous aborderons plus en détail les avantages et désavantages qu'à le *copyright* sur les autres méthodes de protection de la propriété intellectuelle dans la section suivante.

6 Notre traduction du texte original anglais vers le français.

question informatique) entérinent la nouvelle réforme du *copyright*. Au grand dam des responsables du projet de loi, le vote se fait cependant trop près de la fin de la session parlementaire et la réforme n'est jamais mise à exécution. (CONTU 86)

Déçu de ce résultat, le sénateur John Little McClellan – fer de lance de la réforme proposée du *copyright* – voit déjà la nécessité en 1967 de s'occuper de la protection du code informatique :

*I, therefore, concluded that the most desirable course of action would be to [...] establish at the earliest opportunity a national commission to study the copyright implications of these technological advances and to make recommendations to the President and Congress concerning the need for any changes in our copyright law or procedure.* (CONTU 86)

La réforme du *copyright* ne verra le jour qu'une dizaine d'années plus tard. En effet, tous les projets de loi déposés en ce sens en 1969, 1971 et en 1973 meurent au feuillet. Ce n'est qu'après avoir été ramené pour une 5<sup>e</sup> fois en 1974 que le texte sera finalement ratifié par les deux chambres puis entériné en 1976 par le président Gerald Ford (CONTU 87-89). Quant à elle, la commission qui deviendra la CONTU – pourtant si pressante aux yeux du sénateur McClellan – ne sera pas créé avant 1974.

## **1974 à 1978 : la *Commission on New Technological Uses of Copyrighted Works***

Suite à la décision de ne pas prendre en compte l'informatique dans la révision du *Copyright Act* adoptée en 1976, plusieurs membres du Sénat proposent de mettre en place une commission pour réviser les lois traitant de la propriété intellectuelle en informatique. L'avancée rapide des nouvelles technologies a en effet rendu les lois existantes désuètes dans ce domaine. Fidèle à ses intentions en 1967, le sénateur John Little McClellan décide peu après l'adoption de la révision du *Copyright Act* par les deux chambres en 1974 de proposer officiellement la création d'un comité chargé d'étudier la question de la relation entre les nouvelles technologies et le *copyright*.

Plus précisément, la *Commission on New Technological Uses of Copyrighted Works* (CONTU) reçoit le mandat de se pencher sur deux choses : l'application du *copyright* au code informatique et les restrictions devant entourer la photocopie de documents sujets au *copyright* (CONTU 105). Comme la question des photocopies n'est pas pertinente à notre thèse, nous n'aborderons pas ce sujet.

Pour remplir son mandat, la CONTU fait produire toute une série d'études, certaines empiriques comme des sondages ainsi que d'autres plus normatives comme des analyses économiques. La plus intéressante à notre avis est un sondage effectué par une firme spécialisée dans la question des nouvelles technologies. Ce sondage mené auprès de firmes d'informatiques conclut que la protection légale du code informatique n'intéresse pas réellement l'industrie. Présentant le portrait d'une entreprise typique, Richard Miller affirme ainsi que :

*This [typical] company relies largely upon its technological resourcefulness in a burgeoning industry. It is not particularly concerned with the protection of the software that it develops or purchases and, to the extent that it is, would prefer to rely upon physical, technological, and contractual modes of protection rather than legal*

*monopolies. It is not at all convinced that legal protection is necessary and feels that it is generally ineffective even when invoked. The company may – just “may” – take advantage of legal protection if it is offered, provided that it is simple, accessible and inexpensive. The absence of legal protection, however, will not in any way deter it from developing or marketing new programs. (Miller iii)*

Malgré cet avis d’expert pourtant très clair, les membres de la CONTU décident tout de même d’aller de l’avant avec le projet de régulation des droits de propriétés intellectuelles encadrant le code informatique. La première question que se pose la commission est celui du choix de l’outil légal à mettre de l’avant pour protéger les logiciels. Trois grands moyens sont de ce fait étudiés : les brevets, le secret commercial et le *copyright*.

Les brevets s’appliquent normalement à des idées ou à des processus techniques. Cela constitue selon la CONTU une première barrière pour leur application au code informatique; plutôt que de protéger les idées elles-mêmes, les brevets protègent l’exécution technique d’idées. De plus, pour qu’un brevet soit attribué, il est nécessaire de passer à travers un processus rigoureux qui vérifie l’utilité de l’invention, sa nouveauté ainsi que son caractère non-évident.

Conséquemment, la CONTU juge que ce genre de processus prend trop temps et n’est pas assez flexible pour s’appliquer avec succès au code informatique. De surcroît, lorsqu’un brevet est attribué, les fiches techniques de l’invention sont mises à la disposition du public. Dans le cas du code informatique, ce mécanisme légal rendrait alors public le code, ouvrant la porte au plagiat. Finalement, alors que la durée de vie utile d’un programme peut être très longue<sup>7</sup>, les brevets n’offrent une protection légale que pour une vingtaine d’années. Pour toutes ces raisons, les brevets ne sont pas retenus par la CONTU comme un moyen de protection viable (CONTU 16-26).

---

<sup>7</sup> Beaucoup de banques utilisent encore le code qu’elles ont écrit dans les années 1970 et peinent à embaucher du nouveau personnel ayant les connaissances pour le maintenir.

CHARACTERISTICS OF PROTECTIVE MECHANISMS

Considerations	Copyright	Patent	Trade Secrecy
<i>General</i>			
National uniformity	Yes	Yes	No
Protection effective upon	Creation of work	Successful prosecution of application	Entrance into contractual relationship
Cost of obtaining protection	Nil	Moderate	Moderate
Term of protection	Life plus 50 years or 75 years	17 years	Possibility of both perpetual protection and termination at any time
Cost of maintaining protection <sup>1</sup>	Nil	Nil	Significant
Cost of enforcing rights against violators <sup>2</sup>	Moderate	Moderate	Higher
Availability of (a) statutory damages (b) attorney's fees from infringers	(a) Yes (b) Yes	(a) No (b) Yes	(a) No (b) No
Protection lost by	Gross neglect	Unsuccessful litigation	Disclosure
<i>Software, including effects of Commission proposals</i>			
Consistency with other copyright areas	Yes	No	No
Availability of protective mechanism for some programs <sup>3</sup>	Yes	Unclear	Yes
Universal availability of protective mechanism for all programs <sup>4</sup>	Yes	No	No
"Process" protectible	No	Yes	Yes
Suited to mass distribution	Yes	Yes	No

<sup>1</sup> Once copyright or patent is secured, it costs little or nothing to keep it in force; on the other hand, expensive security measures must be taken to avoid losing a trade secret. At least part of the cost of this security is passed on to the user.

<sup>2</sup> Copyright and patent infringers in some instances may be persuaded to comply without the institution of a lawsuit. If litigation is necessary, it may be expensive, but in copyright and patent cases, attorneys' fees may be awarded to successful plaintiffs. At trial, the proprietor bears the burden of proving that the trade secret is valid; in patent cases, there is a presumption of validity; and in copyright actions, a registration certificate is prima facie evidence of the copyright's validity. The proof of the validity of a trade secret may be expensive and difficult, as it almost necessarily involves the retention of expert witnesses. Although witnesses may be needed in copyright and patent suits, in those cases there will have been at least some compliance with federal law regarding public notice of claimed rights before the lawsuit is initiated. A suit to enforce a trade secret, even though successful, may destroy the secret if it is offered into evidence and becomes part of the public record of the trial.

<sup>3</sup> As of the present, serious doubt exists whether programs are proper subjects for patent protection. (See this chapter under Copyright and Other Methods Compared.)

<sup>4</sup> Even if programs are patentable, only those that are truly novel and nonobvious will be protected. Trade secrecy is, of course, unavailable when the contents of a program have been disclosed.

Figure 1: Tableau de comparaisons des différents moyens de protection de la propriété intellectuelle (CONTU 19)

La seconde option étudiée par la CONTU est le secret commercial. Ce dernier est alors surtout utilisé en ingénierie et dans le domaine alimentaire, où une entreprise désire garder la technique de fabrication d'un produit secrète. N'offrant aucune protection légale, la commission note que le secret commercial est principalement utilisé pour protéger des biens qui ne peuvent pas l'être par le *copyright* ou par brevets. Il n'est ainsi pas possible aux États-Unis d'obtenir un brevet ou un

*copyright* sur la recette d'un produit alimentaire (USPTO). Par nature le code informatique est distribué dans tous les ordinateurs qui l'utilisent. Il n'est donc pas réellement possible de garder secret le code utilisé, même en employant des méthodes techniques comme la compilation binaire<sup>8</sup> (Nussbaum 282). La commission décide donc que le secret commercial ne constitue pas un moyen efficace de protéger le code informatique.

Reste donc le *copyright*. Comme ce dernier s'applique immédiatement, est attribué à coût nul, a une durée de vie de près de 75 ans et ne nécessite pas de dévoiler publiquement le code, la CONTU juge que c'est le moyen le plus adapté au code informatique. (CONTU 16-26)

Il est assez frappant de voir que l'exercice de la commission est à notre avis effectuée à rebours. Ayant en premier lieu discuté du meilleur outil légal pour permettre la protection de la propriété intellectuelle du code informatique, la CONTU décide débattre des raisons pour lesquelles le code devrait en soit être protégé. Un processus logique aurait selon nous exigé une démarche inverse.

Pour justifier le choix de protéger le code informatique en vertu du *Copyright Act*, la CONTU met tout d'abord de l'avant la complexité grandissante des programmes. Pour contrer les coûts croissants engendrés par cette complexité exponentielle, la commission avance qu'il est nécessaire de donner des moyens supplémentaires aux entreprises pour protéger leur code. Cet argument ne se base cependant sur aucune étude et semble plutôt révéler d'une incompréhension manifeste du milieu informatique de l'époque de la part d'une majorité des membres de la CONTU<sup>9</sup>.

---

8 Sans entrer dans les détails du sujet, il est possible par des moyens de rétro-ingénierie de reproduire le code source d'un programme distribué sous forme binaire. C'est un processus long et coûteux, mais couramment utilisé par les grandes entreprises pour espionner le travail de leurs concurrents.

9 Denis Ritchie et Ken Thompson – deux fondateurs de l'informatique moderne – publient en 1974 un papier intitulé *The UNIX time-sharing system* qui cristallise une norme culturelle alors déjà centrale dans les milieux *hackers*. Connue de nos jours comme la « Philosophie Unix », ce paradigme encore dominant dans la pratique informatique peut être résumé par la phrase suivante : « *Write programs that do one thing and do it well* ».

Le second argument avancé par la CONTU est que le coût marginal de copier un programme est nul et que cela crée un incitatif positif au plagiat. La commission fait ainsi le parallèle entre le code informatique et les rouleaux perforés des pianos mécaniques. Ces derniers n'étaient pas protégés par le *Copyright Act* de 1909 et ont largement été copiés. La CONTU note toutefois que cela ne semble pas avoir eu d'effet économique néfaste sur la production de ces rouleaux, réduisant ainsi grandement la force de l'argument (CONTU 10).

En définitive, l'argument principal de la CONTU repose sur le fait que le coût marginal de copier un logiciel est nul et que les coûts fixes pour le développer sont importants. Les membres de la commission avancent ainsi qu'il est donc nécessaire de rentabiliser ces coûts initiaux en vendant des copies (protégées par le *copyright*) ou alors en finançant ces logiciels par une fondation ou par le gouvernement. La CONTU juge l'absence de protection sous le *Copyright Act* du code informatique entraînera forcément une réduction drastique du nombre de programmes mis sur le marché (CONTU 11). Cet argument se base principalement sur les résultats d'une étude faite par le département d'économie de l'université de New York (CONTU 125).

L'ensemble des commissaires de la CONTU ne sont cependant pas d'accord avec l'idée de proposer une modification au *copyright* pour que ce dernier s'applique aux code informatique. Le commissaire John Hersey réussit ainsi à convaincre les commissaires Karpatkin, Dix et Nimmer que la CONTU fait erreur (CONTU 27-46).

La critique du commissaire Hersey est assez virulente. D'emblée, ce dernier affirme que « les effets sociaux et économiques de l'existence parallèle du *copyright* et de ces autres formes de protections seraient en définitive négatifs »<sup>10</sup> (CONTU 28). Le cœur de son argument repose sur le fait que le code informatique n'est pas une œuvre littéraire et ne devrait donc pas être protégée

---

10 Notre traduction du texte original anglais vers le français.



comme telle. Alors qu'une œuvre littéraire est définie comme une forme d'expression d'un être humain à un autre, le code informatique est lui destiné à une machine. Dans le cas du code binaire, ce dernier est produit par une machine (un compilateur) dans le but d'être lu par une machine. Selon le commissaire, un programme est plutôt une suite ordonnée d'instructions qui effectuent une tâche performative, s'apparentant plus à un algorithme mathématique (non protégé par le *copyright*) qu'à un livre (CONTU 29).

Pour Hersey, le statu quo est totalement acceptable :

*It appears that the existing network of technological, contractual, nondisclosure, trade-secret, common-law misappropriation, and (in a few instances) patent forms of protection [...] will be wholly adequate, as they apparently have been up to now, to the needs of developers. (CONTU 30)*

Le commissaire termine son argumentaire en mettant en garde la CONTU contre les effets pervers de l'application du *copyright* au code informatique. D'après lui, un tel changement législatif ne peut que renforcer la concentration des pouvoirs économiques au sein de grandes entreprises<sup>11</sup>.

D'un point de vue extérieur, il est frappant de voir que la réflexion de la CONTU en faveur de l'application du *copyright* au code informatique tient sur deux pages seulement, alors que l'argumentaire du commissaire Hersey – journaliste et auteur récipiendaire d'un prix Pulitzer – est beaucoup plus étoffée et nécessite près d'une dizaine de pages. Alors que les autres commissaires se contentent d'une série d'arguments microéconomiques très simplistes, John Hersey pousse définitivement la réflexion plus loin.

---

<sup>11</sup> En rétrospective, alors qu'une partie important de l'informatique mondiale est contrôlée par quatre ou cinq mégacorporations, il semble que ce dernier a au moins partiellement eu raison.

Il est de plus assez étonnant de voir qu'aucun des 14 commissaires n'a de compétences particulières en informatique. Si le Sénat a pris la peine de nommer quatre libraires pour donner à la commission une expertise sur les enjeux de photocopie, il n'a pas jugé bon de faire de même pour le code informatique. En plus des quatre libraires déjà mentionnés, neuf commissaires sont avocat-e-s de profession. De plus, alors que le milieu informatique et plus particulièrement la communauté *hackers* est alors composée d'individus relativement jeunes, la moyenne d'âge des commissaires est de 53 ans et la médiane de 51 ans. Cet écart générationnel important explique à notre avis la méconnaissance de la CONTU de la culture et des modes de productions dans le domaine informatique à cette époque.

Suite à un vote majoritaire de dix contre trois (le commissaire Dix meurt peu avant la fin de la commission), la CONTU recommande les quatre points suivants au Sénat :

1. *Copyright should proscribe the unauthorized copying of these works.*
2. *Copyright should in no way inhibit the rightful use of these works.*
3. *Copyright should not block the development and dissemination of these works.*
4. *Copyright should not grant anyone more economic power than is necessary to achieve the incentive to create.* (CONTU 12)

Comblé de ne pas avoir à se pencher directement sur cette question épineuse, la modification du *Copyright Act* de 1976 proposée par la CONTU est adoptée par le Sénat en 1980 comme amendement au projet de loi 6933 « An act to amend the patent and trademark laws » sans débat et sans opposition (Samuelson 1984, 666).

## **1983 : la fin du débat – le jugement *Apple v. Franklin***

La modification du *Copyright Act* en 1980 n'est toutefois pas la fin des débats entourant la propriété intellectuelle du code informatique aux États-Unis. La nature même du système judiciaire américain donne en effet une place importante à l'interprétation des lois par les différentes cours de justice du pays. Ainsi, plusieurs procès ont lieu entre 1980 et 1983, le plus notable d'entre eux étant la cause *Apple v. Franklin* de 1983.

Fondée en 1976, Apple Inc. se démarque rapidement de la compétition en mettant sur le marché des ordinateurs destinés non pas à des universités ou à de grandes entreprises, mais bien à des individus. Leur premier modèle réellement populaire est l'*Apple II*, un ordinateur accessible à une large population tant par sa faible taille que par son prix. Mis en marché près de 4 ans avant le célèbre *Personnal Computer (PC)* d'IBM, l'*Apple II* est souvent cité comme l'un des premiers micro-ordinateurs utilisés par les hackers des communautés post-1980 (Coleman 28). *Personnal Computer World* – le premier magazine d'informatique britannique – décrit même en 1978 l'*Apple II* comme le seul ordinateur abordable qui offre un écran en couleur. Le succès du *Apple II* est quasi immédiat : d'un revenu de vente initial de 775 000 USD en 1977, Apple engrange plus de 335 millions de dollars de ventes en 1982 en dollars courants de l'époque ("Apple v. Franklin" 1982). D'un point de vue technique, l'*Apple II* utilise principalement un langage de programmation très courant dans les années 1970 et 1980, le BASIC. Cela permet ainsi à différentes entreprises ainsi qu'à certains individus d'écrire facilement des programmes pour ces ordinateurs sans devoir apprendre un nouveau langage de programmation.

Face à la très grande popularité de l'*Apple II* et de l'*Apple II Plus* (une version légèrement améliorée de l'*Apple II*), la Franklin Computer Corporation met en marché en 1982 deux clones de ces machines, l'*Ace 100* et l'*Ace 1000*. Physiquement très semblables aux machines d'Apple,

les ordinateurs de la Franklin Corp. utilisent des cartes mères virtuellement identiques. Un ordinateur est cependant beaucoup plus qu'une simple carte mère : pour s'assurer de la compatibilité de ses ordinateurs avec les programmes écrits pour fonctionner sur l'*Apple II*, Franklin copie 14 programmes inscrits dans la ROM des *Apple II*, dont le système d'exploitation, véritable clef de voûte de la machine<sup>12</sup>.

En 1982 Apple décide donc d'attaquer Franklin en justice et demande une injonction préliminaire pour faire arrêter la vente d'ordinateurs de son concurrent. En cour, Franklin se justifie en affirmant qu'en désignant un système d'exploitation propriétaire (une chose encore rare à l'époque), Apple prive toutes les personnes qui n'ont pas de machines manufacturées par Apple du code écrit pour ces machines. Le litige qui oppose Apple et la Franklin Corp. repose ainsi sur trois arguments juridiques faits par Apple :

1. Le système d'exploitation d'un ordinateur est une forme d'expression protégée par le *Copyright Act* et non pas une idée ou un processus (qui eux ne le sont pas).
2. Le code informatique sous la forme binaire est une forme d'expression protégée par le *Copyright Act* malgré le fait qu'il ne soit pas lisible par des êtres humains.
3. La ROM est un médium d'expression tangible adéquat pour recevoir une forme d'expression protégée par le *Copyright Act* et n'est donc pas une machine.

---

12 Si typiquement le code informatique est écrit dans un langage lisible par des êtres humains que l'on nomme code source, ce code est généralement transformé en code binaire (composé d'une série de 1 et de 0), lisible seulement par un ordinateur. Ces deux formes sont explicitement protégées par la version du *Copyright Act* adoptée en 1980.

Il existe cependant une troisième forme de code informatique, celui inscrit physiquement dans des puces informatiques. Cette forme de code – appelée *Read-only Memory* (ROM) – est composée d'une série de transistors dont l'agencement physique constitue une représentation immuable du code informatique en question. Encore de nos jours, les programmes écrits dans des puces informatiques de type ROM sont couramment utilisés pour permettre à certains ordinateurs de démarrer, ou encore pour accélérer le traitement de certaines opérations de bases très fréquentes dans une machine.

Le juge en charge du dossier affirme lui-même que le cas repose majoritairement sur la difficulté de traiter des questions informatiques du fait de leur relative nouveauté et de l'absence de jurisprudence et de définitions claires des termes utilisés :

*The problems raised here stem from the fact that there is no clear consensus on how to describe the technology employed in microcomputers. With no clarity there, the application of law to fact becomes unsure. (Apple v. Franklin, 1982)*

Rapidement, ce dernier accorde raison à Apple en ce qui a trait au système d'exploitation et au code binaire : la CONTU a été assez claire sur le sujet dans son rapport. Le fait de savoir si la ROM est un médium d'expression est à son avis cependant plus controversé.

En effet, si une jurisprudence existe déjà sur l'application du *Copyright Act* à du code informatique inscrit dans une puce de type ROM ("Tandy Corp"), le juge Newcomer affirme qu'elle contrevient en partie à l'avis de la CONTU sur la question, cette dernière s'étant explicitement opposé à l'application du *copyright* au « fonctionnement électromagnétique d'une machine » :

*Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law. Copyright, therefore protects the program so long as it remains fixed in a tangible medium of expression but does not protect the electromagnetical functioning of a machine. (CONTU 20-21)*

De plus, une jurisprudence contraire sur la question de la ROM existe également ("Data Cash"). Le cœur du débat est donc de savoir si la ROM n'est qu'une machine produite à partir de code informatique comme l'affirme Franklin, ou alors un médium d'expression

approprié qui héberge réellement du code informatique. Sans donner d'avis juridique clair, le juge Newcomer refuse en définitive la demande d'injonction à Apple car il n'est pas clair pour lui que la ROM n'est effectivement pas une machine.

Frustrée de la décision prise en *District Court*, Apple décide donc de s'adresser à la cour d'appel des États-Unis. Contrairement au rapport final du juge Newcomer, les juges Sloviter, Hunter et Higginbotham (tous trois en charge du procès *Apple v. Franklin* de 1983) émettent un jugement extrêmement clair : le code écrit en ROM ne doit pas être considéré comme une machine et constitue bel et bien un médium d'expression appropriée. Les juges sont cependant peu clairs sur les raisons qui les poussent à émettre un tel jugement autre que la jurisprudence existante.

Si à première vue *Apple v. Franklin* peut sembler être une cause parmi tant d'autres, elle joue un rôle fondamental dans l'établissement d'une jurisprudence forte sur la question. En effet, elle clôt complètement le débat sur l'application du *Copyright Act* à toutes les formes de code informatique, incluant celui présent sous la forme de ROM. De plus, contrairement à certains autres jugements précédents comme *Data Cash Systems, Inc. v. JS&A GROUP, INC.*, *Tandy Corp. v. Personal Micro Computers, Inc.*, ou encore *Stern Electronics, Inc. v. Kaufman*, ce jugement traite directement de la question de l'application du *Copyright Act* au code informatique. Finalement, comme ce jugement est rendu en cour d'appel, il fait jurisprudence pour l'ensemble des autres causes qui suivent.

## Conclusion

D'un idéal rassemblant un mouvement de contre-culture dans les années 60 et 70 à un bien économique protégé par un ensemble de lois et une jurisprudence solidement établie en 1980, le code informatique a beaucoup évolué en l'espace de 30 ans. Si de nos jours les communautés *hackers* survivent toujours grâce à la perversion des lois sur le *copyright* que représentent les licences de propriétés intellectuelles libres comme la GNU GPL, force est de se demander si l'application du *copyright* au code informatique était une bonne idée.

Alors que l'informatique est plus en plus centralisée dans les mains de quelques géants monopolistiques du web qui siphonnent nos données personnelles pour mieux les revendre, il est difficile de ne pas imaginer un monde où c'est plutôt l'*ethos hacker* qui aurait remporté la bataille. Une chose est cependant certaine : sans l'utopie désintéressée des *hackers* dans les années 60 et 70 ayant permis de combler les gigantesques lacunes du marché privé, il y a fort à parier que nous vivrions de nos jours dans un monde technologiquement bien différent.

# Bibliographie

- Blodget, Henry. « Mark Zuckerberg on Innovation ». *Business Insider*, 1<sup>er</sup> octobre 2009.
- Coleman, E. Gabriella. *Coding Freedom : The Ethics and Aesthetics of Hacking*. Princeton University Press, Princeton. 2013.
- Levy, Steven. *Hackers: Heroes of the Computer Revolution*. Anchor Press, New York. 1984.
- Levy, Steven. *Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age*. Penguin Books, New York. 2002.
- Lopatka, John E. "UNITED STATES V. IBM: A MONUMENT TO ARROGANCE." *Antitrust Law Journal*, vol. 68, no. 1, 2000, pp. 145–162.
- Nussbaum, Jan L.. « Apple Computer, Inc. v. Franklin Computer Corporation Puts the Byte Back into Copyright Protection for Computer Programs ». *Golden Gate University Law Review*, janvier 1984. Vol.14, Iss. 2, pp. 281-308.
- Ringer, Barbara. « Copyright and the Historian. ». *Biography*, 1979. Vol. 2, No. 1, pp. 1–24.
- Ritchie, Dennis & Thompson, Ken "The UNIX time-sharing system". *Communications of the ACM*. Vol. 17 Iss. 7, 1974. pp. 365-375.
- Samuelson, Pamela. « CONTU Revisited: The Case against Copyright Protection for Computer Programs in Machine-Readable Form ». *Duke Law Journal*, 1984. Vol. 1984, No. 4, pp. 663-769.
- Samuelson, Pamela. « Why the look and feel of software user interfaces should not be protected by copyright law ». *Communications of the ACM*, mai 1989. Vol. 32, Iss. 5, pp. 563-572.
- Stallman, Richard M.. « Free Software: Freedom and Cooperation ». Retranscription de conférence. New York University, New York. 29 mai 2001.
- United States Patent and Trademark Office. « Can Recipes Be Patented? ». *Inventors Eye*, juin 2013. Vol. 4 Iss. 3.
- Williams, Sam. *Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution*. Free Software Society, Boston. 2010.



# Archives

Apple Computer, Inc. v. Franklin Computer Corporation, 545 F. Supp. 812 (E.D. Pa. 1982). *United States District Court for the Eastern District of Pennsylvania*, 30 juillet 1982.

Apple Computer Inc v. Franklin Computer Corporation. 714 F.2d. (Federal Reporter, Second Series), No. 82-1582. *United States Court of Appeals, Third Circuit*, août 1983.

Apple Computer, Inc. v. Formula Intern., Inc. 562 F. Supp. 775 (C.D. Cal. 1983) No. CV82-5015-IH. *United States District Court for the Central District of California*, 25 avril 1983.

*Final Report of the National Commission on New Technological Uses of Copyrighted Works*. National Commission on New Technological Uses of Copyrighted Works, Washington. 31 juillet 1978.

*Copyright Law Revision, Part 5 : 1964 Revision Bill with Discussions and Comments*. Copyright Office, Washington. 1965.

*Copyright Law Revision (House Report No. 94-1476)*. United States House of Representatives, Washington. 1976.

“Copyright Registration for Computer Programs - Announcement from the Copyright Office” *Bulletin of the Copyright Society of the U.S.A.*, 1964. Vol. 11, No. 6, p. 361.

Coll, John & Sweeten, Charles. “Colour is an Apple II”. *Personal Computer World*, 1978. Iss. 8, p. 50.

Data Cash Systems, Inc. v. JS&A GROUP, INC., 480 F. Supp. 1063 (N.D. Ill. 1979). *United States District Court for the Northern District of Illinois*, 26 septembre 1979.

Miller, Richard I. *Legal Protection of Computer Software : An Industrial Survey*. Harbridge House Inc., Boston. 1977.

Stern Electronics, Inc. v. Kaufman, 669 F.2d 852, 855-56 (2d Cir. 1982). *United States Court of Appeals, Second Circuit*, 20 janvier 1982.

Tandy Corp. v. Personal Micro Computers, Inc.. 524 F. Supp. 171 (N.D. Cal. 1981). *United States District Court for the Northern District of California*, août 1981.