

CHRISTIAN JACQUEMIN

*Le bug dans l'œuvre DivXPrime de Bertrand
Planes : invention et mutation*

DivXPrime est une oeuvre de l'artiste plasticien Bertrand Planes : un clone modifié du codec²¹⁵ Open Source DivX, le programme informatique de lecture de vidéos compressées au format MPEG. Créé en collaboration avec des étudiants informaticiens de l'Université Paris 11, ce projet artistique aux multiples facettes peut être vu, dans une première approche, comme un générateur d'effets visuels. Au lieu de lire correctement une vidéo compressée, ce programme génère des artefacts visuels inattendus... S'il n'était que cela, ce projet serait à classer dans la catégorie « Encore un nouvel effet spécial pour le VJ'ing » et nous en resterions là.

Si la sortie de ce programme (l'image vidéo déformée) doit être vue comme un aspect périphérique de ce projet, les deux composantes centrales en sont l'invention (Où a-t-on choisi d'intervenir ? Comment et pour quel résultat a-t-on modifié le code ? Comment a-t-on scénarisé la modification réalisée ?) et la filiation potentielle (Qui pourra le réutiliser ? Est-ce toujours un codec vidéo ? Doit-on corriger les bugs ajoutés par l'artiste ou les conserver scrupuleusement ? Quelle dépendance s'établit entre la filiation de ce projet et celle du codec dont il est issu ?). L'emploi du bug dans la création artistique a un statut qui dépasse la simple retouche cosmétique d'un programme existant ; comme nous le montrerons, cette opération modifie profondément la nature du programme et lui attribue des caractéristiques inédites, germes potentiels d'une descendance originale. Il ne s'agit pas non plus d'une invention *ex nihilo* et nombre de caractéristiques du programme d'origine se trouvent réinvesties et remotivées dans l'oeuvre ainsi créée.

²¹⁵ Codec = *codeur/décodeur*.

Le bug conditionne donc profondément le travail réalisé dans *DivXPrime*, il interroge le statut de l'oeuvre d'art ainsi réalisée qui n'est ni une « amélioration » ni une « retouche » d'un programme existant, mais un nouveau concept artistique trouvant ses racines dans le programme d'origine, son histoire, sa communauté d'utilisateurs et de développeurs. L'oeuvre s'appuie sur le passé technologique et la vie communautaires du codec vidéo qu'elle transforme.

L'encodage séquentiel de la vidéo

Le mode de compression vidéo utilisé dans DivX – et plus généralement dans le format d'encodage séquentiel MPEG4 – est un mode *différentiel* : on encode les informations qui permettent de passer d'une image à la suivante dans des *images delta* (delta frames). Théoriquement il suffit alors de connaître une seule image initiale pour pouvoir reconstruire toute image de la vidéo pourvu que l'on connaisse les images delta pour obtenir l'image $n+1$ à partir de l'image n quel que soit n . En pratique, et en raison de la dégradation progressive de l'image due aux imperfections de la reconstruction différentielle, le codec reçoit régulièrement une image complète, appelée *image clé*, qui permet de repartir d'une image propre (Figure 1). L'insertion de nouvelles images clés est particulièrement utile dans les situations où une image ne se calcule pas bien à partir de la précédente : changements de plans, modifications brusques d'éclairage, déformations rapides, mouvements de rotation...

Notons un dernier point qui a son importance dans l'esthétique visuelle à la base de *DivXPrime* : l'encodage séquentiel repose sur des blocs d'image carrés. Le programme de compression recherche des blocs de pixels carrés communs à deux images successives ainsi que leur vecteur de translation, correspondant, par exemple, à un léger déplacement caméra. En conséquence, toute erreur de transmission des images delta lors de streamings imparfaits rendra visible cette matrice de blocs carrés qui sont incorrectement reconstruits.

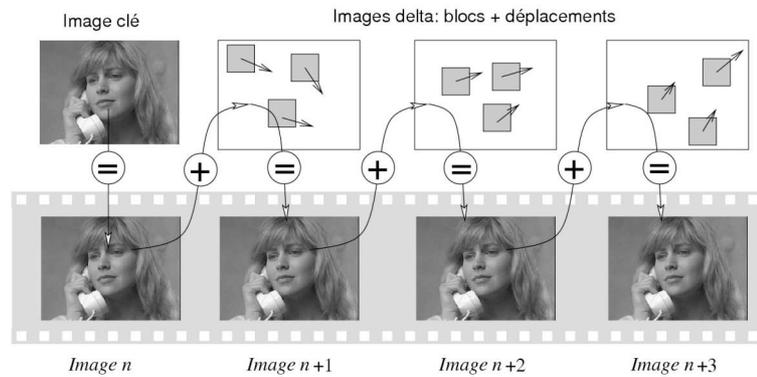


Figure 1. Schéma de l'encodage vidéo séquentiel en MPEG4.

***aPpRoPiRaTe* : une oeuvre sur le détournement de vidéos**

Il existe peu d'œuvres artistiques reposant sur la retouche de codecs vidéo. Nous n'avons eu connaissance que de l'oeuvre de Sven König²¹⁶ intitulée *aPpRoPiRaTe* dont la technologie est très proche du travail présenté ici. L'intention artistique de *aPpRoPiRaTe* repose sur le statut des oeuvres diffusées sur les réseaux *pair à pair* (P2P). Ces réseaux offrent un partage collectif de données et font l'hypothèse que les oeuvres reçues par tous sont identiques à celles émises. En modifiant des codecs vidéo, Sven König souhaite offrir à ceux qui téléchargent des oeuvres sur les réseaux P2P la possibilité d'avoir des œuvres originales, différentes de leur source diffusée. L'artiste intervient sur deux plans. Sur le plan du rendu graphique, il supprime les images clés afin de ne laisser que les images delta réduisant ainsi la qualité de l'image et faisant apparaître explicitement la matrice de blocs carrés reconstruisant l'image à chaque trame, ainsi que des rémanences d'images fantômes qui ne sont pas réinitialisées instantanément en l'absence d'image clé (Figure 2).

²¹⁶ Sven König, (2005) *aPpRoPiRaTe*. In *Readme 100 Temporary Software Art Factory*, Olga Goruinova (edt). Books on Demand GmbH, Norderstedt, Germany. <http://www.popmodernism.org/appropriate/>

Il intervient également sur la linéarité du film : la technologie qu'il a développée permet d'appliquer des algorithmes sur les séquences d'images delta telles que des boucles ou des sauts en avant. En combinant l'explicitation des artefacts d'encodage et la déséquentialisation de la lecture vidéo, Sven König intervient à la fois sur le rendu visuel (une sorte de post-production avec effets spéciaux) et sur le scénario (une sorte de montage dynamique à partir de séquences initiales du film considérées comme des rushes).



Figure 2. Images issues de *aPpRoPiRaTe* de Sven König.

L'invention de Sven König est donc focalisée sur le « produit » et ses « consommateurs » : Comment recréer du contenu à partir d'une source unique à destination d'un public de P2P (qui ne l'a pas nécessairement souhaité) ?

La création de Bertrand Planes est plus systémique : elle produit un nouveau lecteur plus qu'une nouvelle œuvre. Chez Bertrand Planes, la cible n'est pas le consommateur d'œuvres vidéo, mais le programmeur qui voudra avoir des sources informatiques originales à partir desquelles il pourra créer son

propre codec, ou l'artiste qui voudra utiliser son codec comme élément dans une chaîne créatrice. Les deux travaux s'appuient sur les communautés en ligne : Sven König sur les communautés virtuelles de partage via les réseaux P2P, Bertrand Planes sur les communautés de développeurs qui s'organisent autour de projets OpenSource²¹⁷. Donc, bien que les processus inventifs aient une source identique (l'observation et la simulation des erreurs de décodage révélant la matrice d'encodage et son quadrillage), ces travaux sont différents : celui de Sven König porte sur les œuvres, celui de Bertrand Planes sur les programmes et leurs environnements sociaux et technologiques.

***DivXPrime* : Décorrélation entre représentation et description**

Bertrand Planes a inventé *DivXPrime* en observant et en s'intéressant aux artefacts observés lors de streamings vidéo de mauvaise qualité laissant apparaître des blocs carrés d'encodage séquentiel. Était-il possible, en retouchant un programme de décodage vidéo, de retrouver des effets visuels incontrôlés, avec des caractéristiques graphiques remarquables, et signalant une dégradation de la transmission d'un flux vidéo ?

Afin de rendre explicite et maîtriser ces effets indésirables, Bertrand Planes a proposé à Javier Reyes et Mathieu Schaff, deux étudiants de Master1 de l'Université Paris 11, de modifier deux composantes des sources du codec OpenSource Xvid :

²¹⁷ Une communauté de développeurs constituée autour de la réalisation d'un programme OpenSource conséquent tel qu'un codec vidéo gère le travail collaboratif d'une équipe d'informaticiens par des outils d'échange d'informations tels que des forums, des listes de diffusion, des archives de programmes, des rapports de bug... Cette communauté réalise ainsi une oeuvre collective rémanente et évolutive. Le code ouvert produit est utilisable *tel quel* pour le recompiler sur sa propre plate-forme informatique. Il peut également être recyclé partiellement dans d'autres programmes qui, de ce fait, deviennent eux aussi OpenSource puisque cette licence est *contaminante*.

1. la partie où les images clés sont substituées au rendu courant,
2. la partie où les images delta sont utilisées pour déplacer des blocs par translation à partir de leur position dans l'image précédente.



Substitution d'image clé



Perturbation de vecteurs



Suppression d'image clé



Leurre d'image clé

Figure 3. Images courantes des quatre modes de DivXPrime.

Ces deux composantes interviennent à deux niveaux dans la formation d'une image vidéo. Les images clés constituent la *représentation* de l'image : une donnée globale, figée et inaltérable. Les images delta en constituent la *description* : des instructions pour la reconstruction temporelle de l'image, une sorte de guide de montage. L'image vidéo instantanée, hormis celle qui porte sur une image clé, est la combinaison d'une représentation, la dernière image clé rencontrée avant l'image courante, et d'un certain nombre de descriptions utilisées de proche en proche pour reconstruire l'image courante à partir de cette image clé.

Une fois repérés les lieux de définition de ces deux aspects de la reconstruction vidéo dans le programme, l'invention du codec *DivXPrime* a consisté à rompre la cohérence représentation/description ; les quatre modes instrumentaux suivants ont été envisagés et sont illustrés dans la Figure 3 ci-dessus.

1. *La substitution d'images clés.* En remplaçant une image clé par une autre, la représentation fournie est incohérente avec les descriptions qui vont suivre. L'effet visuel obtenu est le recouvrement de la forme spatiale contenue dans les images delta par l'image clé substituée. La description émerge sous la représentation incohérente fournie. Dans l'exemple réalisé, le visage émerge sous un champ fleuri et ce sont les parties les plus mobiles, telles que les paupières, qui vont apparaître en premier.
2. *La perturbation des images delta.* Dans ce mode, la partie descriptive est altérée en modifiant les vecteurs ou les blocs contenus dans les images delta. Dans le projet *DivXPrime*, l'exemple retenu est celui où les vecteurs de translation qui encodent le déplacement des blocs de pixels d'une image à la suivante sont modifiés en échangeant leurs composantes x et y . Il en résulte une reconstruction imparfaite de l'image avec des effets de coulure et de distorsion qui peuvent évoquer des images de visages déformés par la peine ou la douleur. Ce mode est cumulatif puisque les dégradations de la description s'appliquent en cascade à chaque image issue d'une succession de reconstructions incorrectes.
3. *La suppression des images clés.* Ce mode d'intervention a deux déclinaisons. Un premier choix consiste à ignorer une image clé et à se servir de la dernière image réalisée comme image clé. C'est ce que fait Sven König dans *aPpRoPiRaTe*, mais qui n'a pas été valorisé dans le cadre de *DivXPrime*. Cette modification est par nature contextuelle puisqu'elle dépend de la dernière image réalisée. Cela permet à Sven König de construire des boucles qui, bien que convergeant vers une forme stable, ne se répètent pas à l'identique puisqu'à chaque recommencement elles s'appuient sur une image finale légèrement différente. Le deuxième choix, retenu dans *DivXPrime*, consiste à remplacer l'image clé par une couleur uniforme, par exemple du noir. Ce choix est intermédiaire entre le mode 1 de substitution d'une image clé et celui-ci sur sa suppression. Sous l'image noire va progressivement émerger l'image dynamique, les parties animées se réalisant en premier. Ici, la description permet d'accéder progressivement à la représentation, le mouvement en étant le révélateur.

4. Le remplacement d'une image clé par un leurre. Ce mode est un cas particulier du mode 1 sur la substitution d'images clés, mais nous souhaitons le traiter séparément car son effet graphique – et la perception qu'en a le spectateur – sont très différents de la substitution d'une image clé. Le mode 1 consiste à remplacer une image clé par une autre de nature très différente comme la substitution d'un visage par un paysage. Ici, au contraire, l'image clé substituée est très proche de l'image d'origine et les déplacements de blocs qui s'ensuivront seront « presque » cohérents avec l'image de remplacement choisie. L'expérience réalisée dans le cadre de de DivXPrime a consisté à substituer l'image clé d'un visage par celle d'un autre visage calibré et positionné comme l'original. Il en résulte un trouble : l'animation d'un visage sous les traits initiaux d'un autre qui, progressivement acquiert les caractéristiques du visage d'origine par l'animation et les blocs déplacés. En raison de la proximité entre représentation et description, l'image qui se construit décale progressivement notre perception de la représentation initiale vers celle d'une autre personnalité en passant par des phases ambiguës où l'on est plus bien certain d'avoir affaire à la même personne.

Les quatre modes ont pu être réalisés par les étudiants sur la durée d'un stage de Travail d'Etudes et de Recherches, soit une demie journée par semaine pendant douze semaines dans le laboratoire et un travail personnel intermédiaire²¹⁸. Les étudiants informaticiens ont beaucoup apprécié le contact avec l'artiste; ils ont été très impliqués dans le processus inventif et ont contribué par la programmation et par leurs expérimentations aux quatre dispositifs conçus par Bertrand Planes. Cette expérience collaborative a donc été un succès sur le plan pédagogique, alors que les étudiants concernés n'avaient ni formation spécifique, ni goût particulier pour l'art contemporain²¹⁹.

²¹⁸ Javier Reyes et Julien Schaff. (2005) *Réalisation d'effets spéciaux vidéo par des retouches limitées sur un décodeur MPEG-4*, Rapport TER Stage, Université Paris 11, <http://divxprime.free.fr/Soutenance/rapport.pdf>

²¹⁹ Christian Jacquemin et Bertrand Planes. (2005) Travaux d'Etudes et de Recherche en informatique sur des thématiques artistiques. In Actes Atelier *Formations transdisciplinaires en interaction sensorielle pour la création artistique*, IHM'05. Toulouse. http://perso.limsi.fr/jacquemi/IHM05_FISCA/

Eureka ! Le moment de l'invention

Elle illustre une collaboration fructueuse entre art et science d'où émergent des problématiques intéressantes au croisement des disciplines. La coopération entre le LIMSI-CNRS et l'artiste plasticien s'est depuis poursuivie sur d'autres inventions artistiques avec un volet scientifique telles que Gate:3.5.²²⁰

Comme nous venons de le voir avec le quatrième mode, dans ses nuances les plus subtiles, l'œuvre de Bertrand Planes joue sur l'ambiguïté entre le masque – le visage figé initial – et la personnalité vivante – les formes émergentes qui construisent progressivement le visage animé, par touches – ; elle oscille entre icône et présence. Dans ses modes plus esthétisants, les trois premiers, cette œuvre est un générateur d'effets spéciaux permettant de recouvrir un volume avec une image, de dévoiler par bribes animées une surface uniforme, ou de produire des distorsions émouvantes sur une image mobile. Mais comme annoncé en introduction, cette invention interroge aussi le statut du programme informatique en tant qu'instrument de progrès au sein d'une communauté de développeurs et de programmeurs, il en questionne l'essence (la fonction) et l'évolution (les modifications permanentes au sein de cette communauté). C'est ce que nous développons maintenant.

DivXPrime : Invention et mutation

En complément des travaux informatique réalisés, et selon les principes du logiciel libre énoncés par Richard Stallman²²¹, le codec modifié issu du projet artistique *DivXPrime* a été remis à disposition de la communauté en réalisant un site web de distribution²²².

²²⁰ Frederic Wecker. (2006) Gate 3.5 : une oeuvre de Bertrand Planes. In *Art 21* (8): 58-59, Christian Jacquemin, Bertrand Planes et Rami Ajaj. (2007) Shadow Casting for Soft and Engaging Immersion in Augmented Virtuality Artworks. In *Actes, 9th ACM Multimedia 2007*, Augsburg, Germany. ACM Press.

²²¹ Richard Stallman. (1985) The GNU Manifesto. In *Dr. Dobbs' Journal of Software Tools* 10(3).<http://www.gnu.org/gnu/manifesto.html>

²²² <http://www.divxprime.org/>

La nouvelle version du codec s'utilise comme l'ancienne en complément d'un visualiseur de vidéos, mais elle dispose d'une nouvelle interface permettant de régler les paramètres introduits par les modifications réalisées. Le nouveau codec n'en est plus un, car qui utiliserait *DivXPrime* pour lire une vidéo compressée et avoir des effets graphiques indésirables ? *DivXPrime* est un hybride entre une œuvre d'art dont le lieu d'exposition est le web et le premier exemplaire d'une nouvelle famille d'outils dont la finalité serait la création graphique par exacerbation des artefacts visuels d'un codec standard.

L'artiste a donc agi comme un inventeur. Partant d'un objet commun, il a créé un nouveau produit aux propriétés inédites. Les qualités originelles ne sont plus celles du nouveau composé et, en contrepartie, celui-ci possède des nouvelles caractéristiques qui le rendent encore plus précieux. Le tout serait ainsi plus que la somme de ses parties ; les contributions apportées par les lignes de programme dissidentes confèrent au programme inventé de nouvelles propriétés créatrices que le précédent n'avait pas, puisque sa finalité essentielle était la reproduction la plus fidèle possible des vidéos avant leur compression. Sur ce point, et comme annoncé en introduction, le travail de Bertrand Planes et celui de Sven König se rejoignent.

Les grands projets OpenSource, tels que la réalisation de codecs vidéo, regroupent au sein de grands chantiers collaboratifs, des développeurs aux compétences variées, de nationalités et de cultures différentes, mais avec un projet commun reconnu et accepté par tous. Ces sortes de cathédrales des temps modernes, sont forcément vulnérables à la vermine. On doit ici, encore plus que dans des projets industriels fermés, éradiquer les bugs. Pour cela des processus de report et de résolution de bugs tels que Bugzilla²²³ sont mis en place, afin de signaler systématiquement tout dysfonctionnement, et d'en suivre la prise en charge et la résolution rapide.

²²³ <http://www.bugzilla.org/>

Eureka ! Le moment de l'invention

Il y aurait donc, en apparence, contradiction entre l'artiste qui vient introduire des bugs et la finalité sociale et industrielle d'un projet OpenSource.

Entre le bug de développement et le bug artistique la similarité n'est que formelle : les deux types de bugs ont en commun de ne porter que sur un sous-ensemble microscopique du code source. Dans le premier cas, il s'agit d'une erreur involontaire pour laquelle est mise en oeuvre une infrastructure de soins. Dans le deuxième cas, il s'agit d'un acte intentionnel de nano-chirurgie qui vise à modifier pertinemment une zone très réduite du programme afin de lui conférer de nouvelles propriétés. Le bug artistique est volontaire, réfléchi et fortement ancré dans le contexte du programme sur lequel il opère. Il vient modifier l'immense patrimoine génétique du programme OpenSource, dont le code ADN est constitué de centaines de milliers de lignes de programme qui reposent sur des compétences complémentaires, et qui ont été écrites, repentées, réécrites plusieurs fois comme en attestent les archives des versions successives. L'artiste généticien serait donc un générateur de PGM (programmes génétiquement modifiés) venant perturber la filiation de programmes affinés par des générations de développeurs.

Et pour mieux signer son acte hérétique, l'artiste se doit de diffuser son nouvel hybride via le réseau et un site dédié. Dans l'œuvre de Bertrand Planes, le nom du codec comporte un indice de série (le *Prime* de *DivXPrime*) et, malgré sa dissidence, revendique bien l'héritage de son noble ancêtre... Mais il appelle également à de nouvelles filiations. À la logique « traditionnelle » de l'OpenSource qui veut œuvrer pour une amélioration constante des outils qui s'y développent tant par l'éradication des bugs que par l'augmentation des fonctionnalités standard, l'artiste vient mettre son grain de sable en proposant un rejeton mutant dont les caractères diffèrent de ceux de sa lignée, tout en revendiquant une partie de l'héritage. À l'invention se superpose la mutation. L'artiste introduit un patrimoine génétique exogène dans une lignée historique et le dissémine en le mettant à disposition du plus grand nombre.

Retour aux sources

Revenons aux sources des programmes. Qu'est-ce qui confère au code modifié sur une fraction infinitésimale le statut d'invention et d'œuvre d'art ? Qu'est-ce que la création artistique dans le contexte de l'OpenSource ? Faut-il être artiste programmeur pour oeuvrer dans le domaine de l'art programmé ? Est-ce la programmation ou le programme qui fait l'oeuvre ?

Richard Wright, dans *Software Art after Programming* souligne que *The Open Source model is a dead end without an equivalent model of creativity* et engage les artistes à agir sur le domaine de l'OpenSource en inventant des logiciels qui ne sont pas inspirés par un mimétisme servile, mais qui puisent dans les valeurs et catégories philosophiques et culturelles²²⁴. Dans ce même article, il présente le projet Auto-Illustrator de Adrian Ward qui est une sorte de parodie de Adobe Illustrator l'outil de graphisme vectoriel bien connu des artistes. Le clone détourné incorpore de nombreuses références aux poncifs de l'esthétique de l'art numérique tout en restant dans le style de son vénérable référent²²⁵.

Il nous semble que le projet présenté ici œuvre dans le sens où le préconise Richard Wright. C'est bien le programme qui fait l'œuvre et non la programmation qui pourrait se résumer à quelques retouches homéopathiques si l'on voulait être sévère. En fait, ce n'est même pas le programme en tant que somme de ligne de codes qui fait l'oeuvre, mais le programme en tant que projet inséré dans une perspective temporelle communautaire. Le travail réalisé par *DivXPrime* se fait à l'échelle d'une génération au sein d'une entreprise multi-nationale et multi-générationnelle. Dans ces filiations dont la force créative s'essouffle par manque de créativité sous l'emprise d'une culture de la reproduction et de la standardisation, l'artiste intervient pour offrir de nouvelles perspectives.

²²⁴ Richard Wright. (2004) *Software Art after Programming*. In *MUTE magazine* (28). <http://www.futurenatural.net/texts/SoftwareAfterProgram.pdf>

²²⁵ Adrian Ward. (2001) *Auto-Illustrator*. <http://www.adeward.com/wiki/default/read/software+art>

Eureka ! Le moment de l'invention

Il joue un double rôle. Il est *inventeur* en faisant basculer un programme dans une autre famille d'outils par création et émergence de nouvelles fonctionnalités. Il est *généticien* en insérant le nouvel hybride dans la filiation sociale et technologique dont il est issu afin de brouiller les pistes eugéniques de la production standardisée.