

Université Paris 8

Master Création Numérique

parcours : *Arts et Technologies de l'Image Virtuelle*

**Le pipeline de l'image de synthèse :
définitions et enjeux pour les œuvres collaboratives**

Flavio Perez



Mémoire de Master 2, 2016 - 2017

Résumé

Fruit de plusieurs années d'expérience dans différents studios d'animation, la recherche proposée dans ce mémoire se focalise sur les définitions possibles et sur les grandes typologies d'applications qu'un pipeline peut constituer. En d'autres termes, comment se définit un pipeline ? Cette fondation stratégique des productions audiovisuelles 3D, peu encline à être étudiée de façon théorique, nécessite une observation et une expérience concrètes sur le terrain. Replacé dans le contexte spécifique de la création d'images pour films d'animation et les effets spéciaux ce mémoire propose donc de lister les différents niveaux d'implication d'un pipeline puis présente des expériences et réalisations de solutions. La question est étudiée au sens large, offrant quelques ouvertures vers des champs d'application étendus comme les formes d'organisation et la question de la gestion de production. Il se questionnera enfin sur le futur des pipelines.

Abstract

Result of many years of professional experience in several animation studios, the research presented in this master thesis is focused on the possible definitions and applications a pipeline can be. In other words, how is a pipeline defined? That strategic foundation of computer graphic based productions, is hard to study in a theoretical environment, and rather needs a field experience. Contextualised in the specificities of image creation for CG movies and visual effects, this thesis tries to list the different levels of implication in which a pipeline can act, and then presents the results of different experiences and solutions. It is a broad study, including extended definitions related to forms of work organizations and production management. Finally it will reflect on the future of pipelines.

Sommaire

Résumé / Abstract.....	3
Sommaire.....	4
Introduction.....	6
Partie I. Contexte historique et définitions.....	9
I.1. Petite histoire des VFX et des films d'animation.....	10
I.1.1. Une évolution fulgurante.....	10
I.1.2. La démocratisation des effets spéciaux.....	11
I.1.3. Vers la fin de l'indépendance des studios ?.....	11
I.2. L'art et la technique.....	13
I.2.1. Une histoire de relation technique/art qui remonte à loin.....	14
I.2.2. Une évolution qui ne s'arrête pas.....	16
I.3. Qu'est ce qu'un pipeline ?.....	17
I.3.1. Définitions.....	17
I.3.2. Donc qu'est-ce qu'un pipeline ?.....	19
I.3.3. Qu'est ce qu'un TD ?.....	20
I.3.4. Notes sur le manque de bibliographie.....	22
I.4. Conclusion de la première partie.....	23
Partie II. Un ensemble de problématiques.....	25
II.1. Initier un pipeline.....	26
II.1.1. Par quoi on commence ?.....	26
II.1.2. Donner un nom aux choses.....	27
II.1.3. Échanger des fichiers.....	28
II.1.4. Prémices d'un pipeline.....	31
II.2. Particularité des productions.....	32
II.2.1. L'impossibilité de prévoir et le flux tendu.....	32
II.2.2. Brassage des idées et héritage.....	36
II.2.3. Réservé aux gros studios ?.....	36
II.2.4. Postmortems.....	37
II.3. Différents types d'outils.....	40
II.3.1. Outils de confort et d'automatisation : la base.....	40
II.3.2. Outils d'information : l'intelligence.....	42

II.3.3. L'asset management : la maturation.....	46
II.4. Conclusion de la seconde partie.....	54
Partie III. Expérimentations d'outils en production.....	57
III.1. La nomenclature.....	58
III.1.1. Une base essentielle à l'organisation et à la collaboration.....	58
III.1.2. Inadaptation d'une nomenclature à l'évolution d'un studio.....	59
III.1.3. Mutualisation de briques technologiques libres.....	60
III.1.4. Accélérer la transition vers une nouvelle nomenclature.....	63
III.2. Des outils d'automatisation et de confort.....	65
III.2.1. Le cartouche de la Fée coquille.....	66
III.2.2. Un outil d'export Photoshop, un gain de temps considérable.....	67
III.2.3. Storymatic : au-delà du confort, une source d'informations.....	72
III.2.4. Les outils naissent de l'observation des méthodes de travail.....	75
III.3. Des outils d'assistance à la création.....	76
III.3.1. Outil de préparation au story-board et layout.....	76
III.3.2. Du code au service de l'image, l'exemple des cartographies.....	77
III.4. Les outils de suivi de production.....	79
III.4.1. La Feuille De Temps.....	79
III.4.2. Créer un outil d'aide au devis.....	82
III.4.3. Criquet : du provisoire qui dure.....	84
III.5. Outils fondamentaux : Une question de long terme.....	87
III.6. Le futur des pipelines.....	89
III.6.1. Des offres commerciales et des mutualisations.....	89
III.6.2. Les nouveaux standards: USD, EXR, ABC,	90
III.6.3. Pérenniser son pipeline.....	91
III.7. Conclusion de la troisième partie.....	92
Conclusion.....	93
Bibliographie.....	96
Remerciements.....	98

Introduction

À l'origine, j'aurais dû rédiger ce mémoire en 2008-2009, à la fin de mon cursus d'ATI, commencé en 2006, mais mon parcours de formation puis ma vie professionnelle m'ont amené à reporter ce projet durant huit ans. Parti pour apprendre la gestion de production audiovisuelle (que je ressentais comme une vraie nécessité et complémentarité à ATI) par une Licence professionnelle à Gobelins et à l'Université de Marne-la-Vallée, j'ai en effet été vite happé par le monde professionnel, et mon début de carrière qui sollicita toute mon énergie.

Depuis, j'ai navigué entre petits et grands studios d'animation français, à des postes idéalement placés pour observer et apprendre l'organisation des projets et leurs processus créatifs : assistant de production, puis directeur technique spécialisé en questions de gestion de production et enfin superviseur technique. J'ai eu la chance d'être témoin de différentes transitions charnières : tout d'abord chez Mac Guff Ligne devenu Illumination Mac Guff grâce à *Moi, Moche et Méchant*, puis chez In Efecto, studio amorçant un virage vers une structuration technique plus poussée, et enfin, en cofondant Les Fées Spéciales et en participant à la mise en place de la production du long-métrage *Dilili à Paris* de Michel Ocelot.

Aujourd'hui, l'opportunité de reprendre mon Master me permet de prendre du recul et d'analyser certains des processus créatifs et techniques que j'ai pu croiser ou mettre en place. N'ayant ni la place ni la l'ambition de tout raconter dans ce mémoire, j'ai décidé de me concentrer sur un aspect peu documenté, et pourtant facilitateur technique crucial des œuvres de collaboration dans l'image de synthèse : le pipeline.

Je propose donc une analyse rétroactive pour essayer de cerner ce qu'est un pipeline à l'ère des œuvres de collaboration numériques. Car beaucoup de questions se posent autour de ces fondations invisibles du public, loin dans les coulisses des projets de films d'animation, d'effets spéciaux, ou de quasiment toute autre forme de création collaborative d'images de synthèse.

L'étude menée dans ce mémoire est structurée selon trois parties.

La première posera le contexte historique et technique dans lequel s'inscrit le pipeline. À travers cette étape de contextualisation, il sera posé la nécessité de définir ce terme « pipeline » ainsi que son champ d'action et ses acteurs. Est-ce une étape ? Un outil ? À quoi sert-il ? Qui en est l'architecte et qui en est le commanditaire ?

La seconde partie permettra d'entrer dans le vif du sujet. À cette étape de la réflexion, il s'agira de classifier l'existant et les principales problématiques initiatrices de solutions que j'ai pu rencontrer. Autrement dit, nous explorerons les particularités des productions qui ont besoin de pipelines et nous isolerons les grandes familles de solutions apportées aux problématiques repérées.

Enfin, la troisième partie reprendra ces questions à travers les expérimentations diverses que j'ai pu mener au cours de ces dernières années.

Partie I. Contexte historique et définitions

Cette première partie va s'intéresser au contexte global dans lequel s'inscrit cet ensemble de processus de support que l'on appelle le pipeline. Il me semble important de commencer par regarder le contexte historique des studios qui fabriquent ces images afin de se demander qu'est-ce qui les a poussés à devoir se structurer et créer une telle discipline de soutien technique.

Ensuite nous remontrons encore un peu en arrière en observant la relation entre l'Art et la Technologie dans le Cinéma pour comprendre à quel point le Cinéma, puis l'image de synthèse dépendent d'équipes de support.

On se posera enfin la question de la définition du pipeline. Peut-on le cerner précisément ? Quel est son champ d'action ? Qui sont ses architectes ?

I.1. Petite histoire des VFX et des films d'animation

Le concept de pipeline tel qu'étudié ici est partie prenante de l'émergence de studios d'animation et d'effets spéciaux (VFX : Visual Effects). Le pipeline existe dans le contexte bien particulier qui a énormément évolué en 40 ans d'existence de l'image de synthèse.

I.1.1. Une évolution fulgurante

Bien que les premières images de synthèse exploitée pour le cinéma et la télévision datent des années 70 pour les Américains et des années 80 pour les Français, c'est à la fin des années 80 et au début des années 90 que l'industrie des effets spéciaux se met en place. Ce sont des films comme *abyss* (1989), *terminator 2* (1991) et *Jurrassic Parc* (1993) qui marquent réellement le coup d'envoi de cette industrie et de l'image de synthèse telle qu'on la connaît aujourd'hui (Pintea, 2003).

Ces succès, portés par quelques studios historiques (ILM en particulier), vont permettre de mettre en place tout un écosystème industriel dans les années suivantes : Sony Pictures Imagework, Digital Domain, Weta Digital pour les principaux.

Au milieu des années 90, la technologie s'étend à des projets de plus en plus gros et nombreux. Le premier long métrage d'animation en images de synthèse (*Toy Story*) sort en 1995, et des films utilisant de très nombreux effets spéciaux suivent, comme *Titanic* (1997) pour lequel Digital Domain a dû prendre en charge près d'un millier de plans¹. La même année Sony Imageworks signe les effets spéciaux de *Starship Troopers*. À la fin des années 90 et au début des années 2000 arrivent les premières grosses franchises exploitant et dépendant massivement des effets spéciaux numériques : la trilogie *Matrix* (1999, 2003, 2003) ouvre le bal tout comme la *Star Wars* (1999, 2002, 2005) et le *Seigneur des Anneaux* (2001, 2002, 2003). Les moyennes de plans truqués sur ces films sont déjà importantes : entre 500 et 2000 plans à charge des studios d'effets spéciaux (Butler, Langlands, & Ricklefs, 2008), quand un film est constitué de 1500 plans en moyenne.

La France suit le mouvement. Des studios comme BUF, Duran Duboi, Mac Guff Ligne, Mikros Images dès les années 80 et, plus tard, La Maison, Delaposte, Digital District, ... prouvent rapidement leur savoir-faire (Welker, 2015) et participent dès les années 90 puis début 2000 à de grosses productions américaines et européennes. Les écoles, le niveau technique, les systèmes de financements, le *système D* et le bon sens des équipes techniques, et des auteurs qui s'intéressent

¹<https://www.fxguide.com/featured/titanic-stories/>

aux effets spéciaux sont certains des paramètres qui maintiennent la France juste derrière le peloton de tête.

I.1.2. La démocratisation des effets spéciaux

Avec des projets de plus en plus importants, des studios de plus en plus nombreux, on assiste à partir de 1999² à la démocratisation et à l'industrialisation des effets spéciaux numériques dans les pays occidentaux et en Asie. On recense en 2017 plus de 2000 studios d'effets spéciaux de toutes tailles et partout dans le monde³ contre une trentaine en 1993. Ceci a pour effet de provoquer d'importants changements dans le milieu de l'image de synthèse :

- De gros éditeurs de programmes informatiques prennent le relais des développements maison (historiquement les studios développaient leurs propres solutions) et de nouveaux logiciels émergent. Souvent issus des studios, on peut noter *Maya* dès 1998, *Nuke* à partir de 2002, *Katana* issu de Sony Imageworks débuté en 2004⁴, puis repris et mis en vente en 2011 par the Foundry ; et avant ça l'incontournable *Photoshop*, initié en 1990 par les frères Knoll, dont le cadet, John, était déjà superviseur chez ILM⁵.
- les formations, qui étaient encore très scientifiques et expérimentales jusque-là s'organisent pour répondre à l'engouement (par exemple Supinfo.com, fondé en 1988, grossit et crée sa filiale à Arles en 2000).
- des standards émergent des studios, comme le format d'images *OpenEXR* (créé en 1999 et rendu public en 2003), *Alembic* en 2011, ...

Bref, l'industrie se professionnalise et 2009 voit la sortie d'*Avatar* comme une étape décisive et réussie du premier film d'animation d'effets spéciaux.

I.1.3. Vers la fin de l'indépendance des studios ?

Depuis 2012 l'indépendance des studios d'effets spéciaux (jusque-là relégués au rang de simples prestataires de postproduction) est remise en question. En effet tant le savoir-faire des studios est lié à l'existence même de certains films on observe des évolutions de statut ou des reconfigurations : Digital Domain devient une structure de production ; ILM, qui a aussi fabriqué un film d'animation

²date de sortie d'un important nombre de superproductions évoquées précédemment

³voir des sites comme www.cgstudiomap.org ou vfxworldmap.com qui en recensent une partie

⁴Le communiqué de presse (imageworks.com/newsroom.php?p=article&s=110311) relayé par Sony Imageworks pour sortir la version 1.0 de *Katana* rappelle les origines

⁵Histoire souvent relatée comme par exemple sur photoshopnews.com/feature-stories/photoshop-profile-thomas-john-knoll-10/

(*Rango*, 2011) a été revendu pour devenir une branche à part entière du groupe Disney (qui détient des franchises comme Marvel et Star Wars intimement liées à ILM). Le studio MPC conçoit *Le Livre de la Jungle* (2016) et s'attaque actuellement au *Roi Lion* (2019).

Ces évolutions structurelles sont liées aux nouveaux rapports de force entre studios et sociétés de production : d'un côté, les majors dépendent de plus en plus de la réussite de franchises utilisant beaucoup d'effets spéciaux. La part de l'image de synthèse sur le devis d'un film devient majoritaire et les majors dépendent donc de la capacité des studios à fabriquer les images. De l'autre, les studios ont les carnets de commandes bien remplis et risquent de repousser ou refuser des projets par manque de ressources tout en dégageant de moins en moins de marge dans un environnement très concurrentiel. Dans ce contexte, acquérir ces studios devient donc pour les *majors* une garantie de contrôler la faisabilité, le calendrier et le coût des films.

Un cas qui illustre bien cette dépendance est la fermeture de Rythm and Hues, concepteur du succès international *L'odyssée de Pi* (2012), qui a secoué la profession. Cette cession d'activité a alors mis en évidence la fragilité de la simple relation de client/prestataires sur de telles productions (Ross & Squires, 2013). Les studios de VFX qui sont devenus responsables de la réussite des films ne profitent pas de leur succès. Et en risquant de ne pas survivre, ils mettent en danger la fabrication des prochains films. Racheter ces studios permet aux majors de s'assurer que le savoir-faire de tels titres, réservé à une poignée de studios, ne périsse pas et reste sous leur contrôle exclusif.

Côté films d'animation, la relation studio-projet est déjà forte, puisque c'est un studio qui est à l'origine de la fabrication intégrale de l'image. Il est d'ailleurs habituel d'aller voir le nouveau film de tel ou tel studio. Ghibli, Disney, Pixar, Dreamworks, BlueSky, Cartoon Saloon, Les Armateurs, Folimage, sont des noms de studios plus souvent retenus par le grand public que le nom des réalisateurs eux-mêmes.

Ici aussi les rachats pour assurer le contrôle de la fabrication s'opèrent, avec Dreamworks qui a créé sa branche animation grâce au studio PDI. Mikros animation racheté par Technicolor qui avait déjà acquis MPC. Ou plus récemment la branche animation du studio français Mac Guff Ligne, rachetée par Illumination et devenue Illumination Mac Guff en 2011. Ce studio de fabrication appartient aujourd'hui à Illumination Entertainment, qui appartient à NBC Universal Comcast (qui détient aussi depuis peu Dreamworks Animation). Illumination Mac Guff travaille donc exclusivement pour Universal. L'autre branche de cette séparation, qui a gardé la structure historique, Mac Guff Ligne, n'ayant eu le droit de ne participer qu'à de "petits films" qui n'entravent pas les projets américains, comme *Kirikou 3* (2012) et aujourd'hui *Dilili à Paris* (2019) dont je reparlerai.

On sent alors que les questions qui étaient propres à une société isolée et autonome évoluent pour se confronter aux effets de mondialisation du milieu. Les pipelines doivent alors devenir des solutions globales pour accompagner des flux de plus en plus intenses et décentralisés.

Dans sa conférence introductive au Siggraph University « Digital Production Pipeline », Paul Strauss, ancien développeur chez Pixar, remarque que 3 facteurs clés et déterminants ont accompagné cette transition rapide de l'industrie des effets spéciaux.

D'abord de **meilleures** images dans les années 90. Fini le polygone, les représentations sont de plus en plus réalistes au point de devenir impossibles à distinguer d'images réelles⁶. Ensuite un paramètre **économique**, celui d'une recherche du coût de moins en moins cher pour une image. Et enfin un paramètre déterminant : la **technologie**, agissant en coulisse.

En effet, sur cette période de 30 ans nous sommes passés des mains virtuelles de Ed Catmull aux scans très hautes résolutions de sites entiers (décors de tournages, sites archéologiques, ...) ou de représentations ultra-précises de la lumière et de matières aussi fines que la peau (Texturing.xyz, Paul Debevec, ...). On a évolué des ordinateurs coûteux et dédiés à des tâches précises comme les Silicon Graphics, à des parcs de « simples » PC polyvalents sur Linux, profitant de processeurs (CPU et GPU) et de mémoire RAM offrant une puissance de calcul très bon marché (même pour des étudiants et particuliers). Le coût du stockage s'est effondré, et nous sommes passés de mégaoctets, à gigaoctets puis à téraoctets (et petaoctets pour les plus gros studios). Enfin, la question de l'*asset management* qui est venu structurer tout ça a permis de faire des projets encore plus complexes.

L'image de synthèse existe à travers l'utilisation absolument fondamentale et nécessaire de technologies très complexes que sont les ordinateurs et les programmes 3D. Elle est le résultat d'une fusion particulièrement forte de l'Art et la Technologie.

I.2. L'art et la technique

L'image de synthèse s'inscrit dans un ensemble plus global d'arts audiovisuels et notamment le Cinéma. Ce dernier est un art récent, mais on va voir que sa dépendance à la technique était déjà forte.

⁶On atteint aujourd'hui clairement un des enjeux initiaux de l'image de synthèse avec cette course au réalisme et au mimétisme de la nature où l'ordinateur doit se faire oublier, comme quand les tableaux de la Renaissance cherchaient à être une « une fenêtre sur le monde »

I.2.1. Une histoire de relation technique/art qui remonte à loin

Remarquez avant d'entrer en matière que ce mémoire n'entrera pas dans des débats de définitions et d'exceptions sur ce qui fait une oeuvre audiovisuelle et en particulier un film, ce n'est pas son objectif. Le pipeline, dont il sera question dans cette recherche, peut concerner la plupart des oeuvres audiovisuelles (de la pub pour la télévision au cinéma en passant par des installations de réalité virtuelle) sans distinction. Mais un petit retour en arrière s'impose.

Le cinéma est la rencontre entre l'Art et la Technologie depuis son premier jour. Cet art ne peut pas vivre sans une technologie de support très complexe, contrairement aux autres Arts dont les formes les plus simples peuvent s'exprimer sans technologie. En effet, le cinéma nécessite des technologies complexes de captation et de diffusion, qui ont mêlé chimie et mécanique au départ, puis électronique et numérique dans leurs évolutions les plus récentes.

Sans couvrir les interprétations que l'on fait des peintures rupestres ou du théâtre d'ombres à travers les âges, et si l'on en vient à la définition générale et contemporaine d'images animées, on remarquera qu'il s'agit du même défi mêlant moyens techniques et enjeux artistiques. Dès le XVIe, la *lanterne magique*⁷ fonctionne ainsi grâce à l'émergence de nouveaux procédés optiques. Suis ensuite l'effervescence du XIXe et toutes les inventions qui se succèdent pour donner vie aux images dans un large mouvement qu'on appelle le *précinéma*. On note, entre autres, le remarquable travail du *Théâtre optique* de Charles-Émile Reynaud qui dès 1892 propose un spectacle animé et projeté de 5 minutes. Trois ans plus tard, le *cinématographe* des frères Lumière va marquer le lancement du cinéma tel qu'on le connaît. C'est une date charnière qui va permettre la *prise de vue réelle*, c'est-à-dire capter le monde qui nous entoure.

Le XXe siècle va quant à lui être témoin d'une évolution constante et fulgurante. Le premier dessin animé sur support argentique sort en 1906 (*Humorous Phases of Funny Faces*, 3 minutes) de la main de James Stuart Blackton. Le procédé est un simple retour à la photographie : le film est tourné photogramme après photogramme. Entre les deux il prolonge son dessin à la craie sur un tableau noir.

La relation entre les ingénieurs, les inventeurs et le monde de l'animation est forte. En 1912 un animateur met au point la barre à tenons pour stabiliser son travail d'une feuille à l'autre. En 1914 un producteur et un réalisateur utilisent le film transparent non plus sous forme de bande (la pellicule),

⁷La lanterne magique est l'ancêtre des appareils de projection et particulièrement du projecteur de diapositives. Inventée au xviiie siècle par l'astronome Christian Huygens, elle permet de projeter des images peintes sur des plaques de verre à travers un objectif, via la lumière d'une chandelle ou d'une lampe à huile. fr.wikipedia.org/wiki/Lanterne_magique

mais sous forme de feuilles indépendantes et maniables, mettant ainsi en place la technique du *cellulo*. En 1915 Max Fleischer met en place le *rotoscope*, un dispositif permettant de redessiner avec une grande précision les mouvements d'un acteur et invente donc la rotoscopie⁸. Plus tard les ingénieurs de Disney mettent au point des inventions aussi complexes et industrielles que le *multiplan*. Bref, les ingénieurs permettent aux artistes de créer des images encore plus spectaculaires génération après génération. La profession standardise de plus en plus les techniques. En 1920 on fait les premiers films avec pellicule couleur (*The Debut of Thomas Cat*), mais il faudra attendre 1932 et un accord entre Disney et technicolor pour voir les premières grandes œuvres colorées.

Dans les années 30, les studios émergent et se structurent. Mais on avait déjà eu des longs métrages dans les années 10 en Argentine (*El Apóstol* de Quirino Cristiani, 70 minutes). En Europe le premier long-métrage (*Les Aventures du prince Ahmed*, 65 minutes) date de 1926 de l'Allemande Lotte Reiniger avec du papier découpé. Et en 1937 le public découvre *Blanche Neige et les sept nains* de Walt Disney et le *Roman de Renard* de Starewitch pour ne citer qu'eux.

L'arrivée de l'informatique permet dès les années 70 l'apparition des premières images de synthèse. Rares sont les artistes à pouvoir manipuler ces machines à cette époque-là. Ce sont des mathématiciens, comme Ed Catmull qui sera l'un des fondateurs de Pixar 10 ans plus tard, qui sont les premiers à créer des images animées numériques.

L'arrivée du numérique et de l'automatisation qu'il permet ne va pas masquer pour autant l'étroite relation entre l'art, ses enjeux et la technique, bien au contraire. John Lasseter, célèbre réalisateur et président des studios Pixar/Disney, le résume même ainsi (Iwerks, 2007) :

« L'Art défie la technologie. La technologie inspire l'Art »⁹

⁸ ce terme est toujours utilisé aujourd'hui, notamment dans les effets spéciaux, pour créer le détournement précis d'un personnage à l'écran par exemple, utile ensuite pour masquer des effets spéciaux en arrière-plan.

⁹ « The art challenges the technology, and the technology inspires the art »

I.2.2. Une évolution qui ne s'arrête pas

Les artistes, et particulièrement dans l'animation, sont des techno-enthousiastes. Leur curiosité les a toujours poussés à expérimenter. La longue évolution des techniques pendant plus de 100 ans a été marquée par la révolution majeure du passage au numérique qui ouvre de nouvelles et vastes possibilités.

Le Cinéma en général a été profondément affecté par l'arrivée de l'ensemble des technologies du numérique (Kenneally, 2012). En effet, l'animation va connaître sa plus grosse accélération et démocratisation grâce à ces outils qui permettent de produire plus, plus vite et à plus grande échelle. On l'a vu en partie I.A. Il faut au moins retenir que la standardisation qui s'était mise en place *de facto* dans l'animation traditionnelle va être au moins aussi forte avec la transition numérique et à mesure que la taille et la complexité des projets explosent.

Et les technologies continuent à évoluer, année après année. Que ce soit en termes de méthode de fabrication des images, de façons de raconter des contenus, et de solutions pour accéder à ces derniers.

Par exemple, la question des salles de cinéma est un enjeu important à l'heure où le grand public a, dans son cercle privé, des technologies de restitutions des oeuvres audiovisuelles performantes. Les espaces de diffusion publics doivent donc s'inscrire dans un élan de technologies de restitution des oeuvres cinématographiques toujours plus compétitives. Et ce afin de garder les spectateurs en se distinguant sur le marché par l'excellence de programmation d'un côté et par l'expérience de plus en plus immersive de l'autre (Son ATMOS, Éclair Color, taille d'écran, projection laser, ...). Les dispositifs privés misant sur de nouveaux supports émergents qui se démocratisent (360, réalité augmentée, réalité virtuelle, ...).

Pendant ce temps-là, les technologies permettent de faire des images de plus en plus précises et *réalistes* (4K, 8K, HDR, ...) pour ces différents marchés.

Les historiens pourront trancher plus tard sur le fait que nous atteignons aujourd'hui un seuil, à l'heure où les seules limites sont l'imagination et les budgets, mais non plus la technique. Peut-être sommes-nous tels des pionniers ayant traversé l'Amérique du Nord de part en part et n'ayant plus qu'à revenir explorer les terres sauvages que l'on a laissées derrière nous ?

Le pipeline va dès lors être un acteur fondamental. Caché en coulisse il sera le support de l'ambition artistique et permettra des réalisations de plus en plus techniques dans un contexte économique resserré.

I.3. Qu'est ce qu'un pipeline ?

Fabriquer un film en images de synthèses est un long processus faisant intervenir de nombreux corps de métiers tout au long d'une chaîne de fabrication complexe. Je ne détaillerai pas l'ensemble des étapes nécessaires, mais on peut en avoir un aperçu sur le schéma suivant.

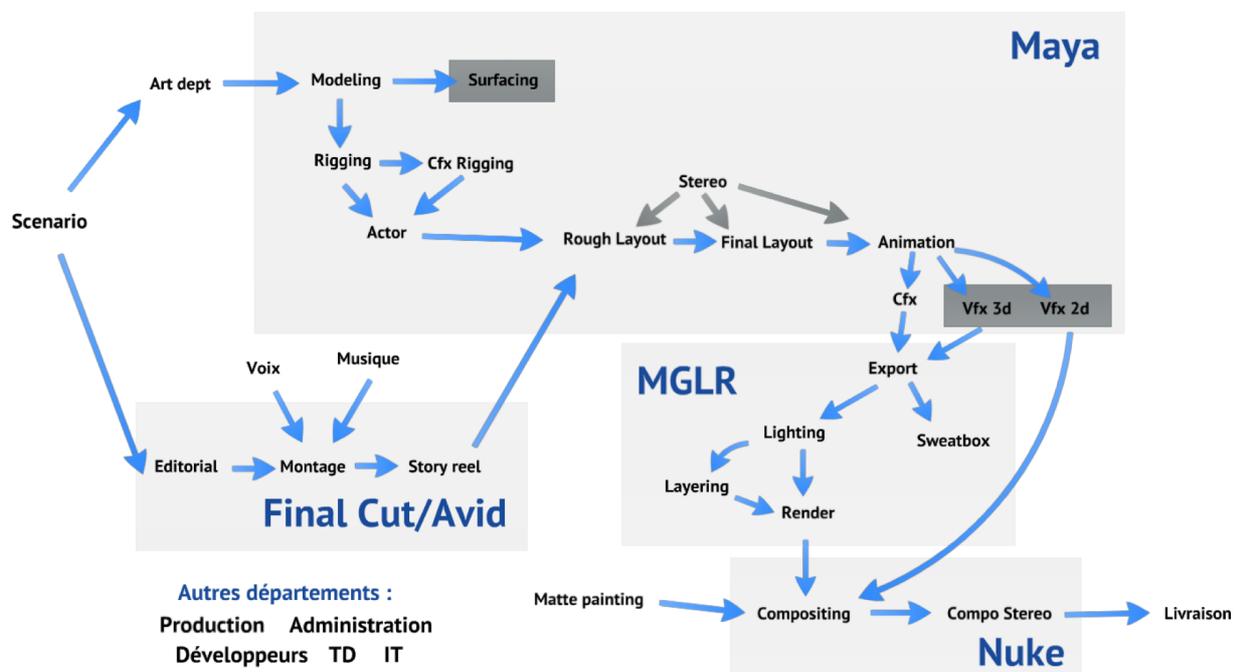


Illustration 1 : schéma simplifié des principaux départements d'Illumination Mac Guff sur *Moi, Moche et Méchant*. Le fond gris indique le principal logiciel utilisé, MGLR étant le moteur de rendu maison

Le pipeline va être l'ensemble des solutions et processus techniques qui permettent cette organisation. Mais on va le voir, un pipeline à une définition un peu floue à l'heure de cerner son champ d'action précis.

I.3.1. Définitions

Pipeline est un de ces mots protéiformes que l'on retrouve dans les contextes de production. Je propose donc un état des lieux de quelques définitions.

Anne-Laure George-Molland dans sa thèse *La collaboration au cœur du processus de création des œuvres audiovisuelles numériques* (George-Molland, 2007) utilise une définition générale et pragmatique :

Pipeline: terme s'appliquant habituellement aux systèmes de canalisation. Par extension, on l'emploie pour désigner le processus de fabrication, ou l'ensemble matériel et logiciel mis en place pour répondre aux besoins d'une production.

Dans *Production Pipeline fundamentals* (Dunlop, 2014), le seul ouvrage publié à ma connaissance sur cet aspect précis de la production, les auteurs entrent en matière en décrivant ainsi le pipeline :

Le pipeline est la colle qui unifie le travail de tous les artistes impliqués dans une production. Le pipeline est comme une chaîne d'assemblage sur laquelle chaque intervenant effectue sa tâche avant de transmettre leur travail fini au suivant sur la ligne. La principale différence entre créer du contenu artistique et fabriquer des produits industriels est que le processus créatif comporte des itérations de révision à travers lesquelles le produit final peut être révisé et, si nécessaire, le pipeline lui-même modifié.¹⁰

Darin Grant, de Method Studios, a une définition semblable et précise 3 composantes au pipeline (Grant, Stein, Laviertes, Gibbs, & Ford Grant, 2014) :

- Les **infrastructures** : Les machines et services centraux (gestion des users, réseaux, filesystem, ...) qui mettent à disposition les outils numériques nécessaires
- Le **Pipeline lui-même** : Les outils et les processus
- Et le **Workflow** : Comment quelqu'un utilise ces outils.

On voit ici aussi un exemple d'une utilisation polysémique du mot pipeline, utilisé comme terme général et comme une notion plus resserrée dans la définition même de ce terme. On le verra en partie II. mais la circulation de termes génériques mais définis est un problème à l'heure d'avoir des discussions dans lesquelles on doit toujours préciser quel sens d'un mot on utilise.

John Carey sur le google group *Global VFX pipeline*, prend un peu de hauteur et résume le pipeline de façon plus synthétique :

Un pipeline est un moyen formalisé de transformer des données d'un état à un autre à travers une série de processus de gestion.¹¹

¹⁰Les traductions sont faites par l'auteur. Traduit de : « The pipeline is the glue that holds together the work of each artist involved in a production. In this, a pipeline is much the same as an assembly line, in which each worker performs their task before handing off their completed work to the next. The main difference between creating art and creating factory goods is that the creative process incorporates review cycles through which the final product can be redefined-and, if necessary, the pipeline itself modified »

¹¹Traduit de « A pipeline is a formalized way of transforming data from one state to another through a series of management process »

Plus tard dans le fil de discussion du groupe il précisera deux niveaux d'abstraction en disant :

La gestion de production s'intéresse au workflow alors que le pipeline gère le dataflow.¹²

Ces deux notions, *workflow* et *dataflow*, sont intimement liées dans la notion générale de pipeline. Le premier définit dans quel ordre et de quelle façon les artistes et techniciens oeuvrent sur une image. C'est typiquement un processus itératif où des boucles peuvent avoir lieu et où l'on peut recommencer des étapes. Le second s'intéresse à la manière dont les données elles-mêmes progressent dans la chaîne de fabrication. Ici le processus est linéaire et les données ne reviennent pas en arrière. On y reviendra.

Enfin, pour Mike Stein de MPC résume ainsi la situation (Grant et al., 2014) :

Pipeline : un ensemble de logiciels et technologies qui permettent aux artistes de travailler sur une grande quantité de tâches de façon consistante, efficiente et répétée.¹³

Stein précise aussi dans la même conférence qu'un pipeline « n'est pas un processus de production, ni les solutions de communication (note de l'auteur : pour savoir quand un asset est prêt par exemple), ni un responsable de la qualité visuelle. Mais qu'un pipeline permet toutes ces choses. » On le voit, la définition du mot pipeline tourne autour d'un même centre d'intérêt, les processus qui permettent aux artistes de travailler, mais a une limite et un champ d'application un peu flous.

I.3.2. Donc qu'est-ce qu'un pipeline ?

À mon sens un pipeline est la fondation invisible aux enjeux artistiques, techniques et aux questions de productions.

C'est un ensemble de procédures très organiques qui n'a d'existence que par les artistes et techniciens qui le nourrissent en données que d'autres graphistes transforment.

Un pipeline n'a rien de révolutionnaire. C'est la structure nécessaire qui évolue en fonction des paramètres de production (qualité d'image, coûts et technologies employées). Le pipeline est au coeur du processus de collaboration dans les images de synthèse.

¹² Traduit de « Production management addresses issues of work flow while the pipeline deals production data flow »

¹³ Traduit de « Pipeline : A set of software tools and technologies that enables artists to perform a large number of tasks consistently, efficiently and repeatably »

Même de nouvelles formes d'organisation, comme les projets décentralisés que l'on verra plus tard, reposent entièrement sur un pipeline adapté.

Un pipeline s'ajuste donc aux choix de fabrication et comble les vides.

Un pipeline est souvent découpé en sous-pipelines, organisations plus petites au sein d'un dispositif beaucoup plus large. Des départements peuvent par exemple mettre en place des procédures qui leur sont propres et indépendantes du processus global. Une arborescence hiérarchisée permet alors de traiter différentes échelles de données.

La technologie est propice à la créativité. Du coup quelle que soit la taille de la structure, on demande de plus en plus de plans et de plus en plus de complexité. Les questions de *workflow*, *dataflow*, d'automatisation, d'*asset management*, que j'aborderai dans ce mémoire, sont certaines des questions aux multiples faces auxquelles les équipes de production doivent répondre. Car les standards de qualité et d'exigence ne cessent d'augmenter, mais pas le temps humain disponible. Les ressources logicielles et matérielles doivent donc prendre le relais pour soutenir autant que possible la création. Cela fait du pipeline une composante majeure de la continuité de l'évolution du secteur.

I.3.3. Qu'est ce qu'un TD ?

Avant de plonger dans le cœur du sujet il est important d'aborder l'un des acteurs principaux du pipeline : le TD (prononcé avec la phonétique anglaise) pour *Technical Director*.

La thèse d'Anne-Laure George-Molland définit ainsi ce rôle :

Les TD sont des infographistes ayant des connaissances techniques pointues, et pouvant s'attaquer à des problèmes de production complexe. Ils sont capables de développer des programmes pour résoudre des situations particulières du projet. On trouve des TD spécialisés pour chaque étape de fabrication et des TD généralistes touche-à-tout. Le TD s'occupe du suivi et de la maintenance des programmes qu'il crée. Certaines étapes étant plus techniques (comme la fabrication des shaders, l'éclairage, le rendu) on emploie très couramment le nom TD à la place de Artist.

La convention collective du cinéma d'animation¹⁴ ne reconnaît pas le terme TD, préférant *infographiste scripteur* ou *infographiste développeur*¹⁵ et le réduit à cette définition :

Conçoit des modules complémentaires aux logiciels de création et de production d'images utilisés dans le cadre de la production, par le biais d'une interface de programmation ou d'un langage de commandes.

Si vous regardez un générique de long métrage, particulièrement anglo-saxon, vous remarquerez néanmoins que TD s'utilise dans tout département dont les tâches comprennent des éléments techniques. Je trouve que c'est un raccourci car pratiquement tout dans l'image de synthèse est technique. Le terme est alors utilisé sur des profils qui ne pratiquent jamais l'écriture de code informatique. Mais peut-être est-ce le terme de TD en tant qu'infographiste scripteur ou pipeline-TD qui est trop générique.

L'utilisation de TD dans ce mémoire se réfère donc au sens généraliste, que l'on pourrait qualifier de "TD à la française", qui se situe entre graphiste et développeur et qui met en place des solutions techniques en utilisant du code. Les Anglo-saxons préciseraient le poste, comme pipeline-TD.

En plus des TD, nous évoquerons les équipes de Recherche et Développement (R&D) oeuvrant sur des programmes plus lourds (moteurs de rendus, formatage des données, etc.) ou plus expérimentaux. Ces équipes sont plus souvent composées d'ingénieurs et développeurs de formation.

Enfin, les TD sont en étroite relation avec les IT (Information & Technologie), terme qui désigne les compétences ou les postes liés aux Technologies de l'information et de la communication. L'utilisation la plus courante concerne les équipes chargées des questions informatiques au sens large : des postes de travail aux réseaux informatiques en passant par le stockage de données.

Développeurs, TD et IT travaillent ensemble et représentent l'équipe de support, noyau dur des aspects techniques, dans un studio d'animation ou d'effets spéciaux.

¹⁴La convention collective est un texte officiel définissant les particularités des postes de la branche professionnelle des salariés du cinéma d'animation

¹⁵Le second est aujourd'hui le terme officiel, pourtant le premier paraît plus judicieux car il me semble qu'un scripteur, ce qu'est souvent un TD, n'a vraiment pas la même approche qu'un développeur

I.3.4. Notes sur le manque de bibliographie

Contrairement à quasiment tous les autres champs de recherche autour de l'image, ou de l'image de synthèse, des processus créatifs, il y a peu de littérature sur le pipeline. C'est un champ d'études peu exploré alors même que ses procédures et solutions sont une absolue nécessité dont la taille et les capacités de charge sont fonction de la taille du studio et du projet. Pourtant le pipeline est relégué à des questions de support aux mains d'une toute petite poignée de techniciens. Peu évocateur, on n'en parle jamais dans les *making-of*. Car s'il y a bien une chose que l'on ne voit pas à l'image c'est le pipeline.

Lorsque l'on fait une recherche dans les bases de données de l'ACM Siggraph¹⁶, le mot pipeline renvoie à plus de 4000 résultats. Cependant, en dépouillant ces résultats, l'on se rend compte que moins de 1% ou 2% des articles sont vaguement liés au sujet du pipeline tel qu'on l'entend (les processus et organisations pour fabriquer des images). Car ce terme est utilisé dans bien d'autres situations qui génèrent plus de littérature comme la conception des processeurs. De ces 2%, la plupart utilisent le mot pipeline comme le terme générique d'organisation au sein d'une problématique précise, et non générale appliquée à la fabrication d'un projet audiovisuel. Par exemple, un pipeline pour faire des arbres réalistes dans le *Livre de la Jungle* (Schwank, James, & Micilotta, 2016).

Restent donc quelques articles qui évoquent le pipeline comme la fondation technique de la fabrication d'images animées et effets spéciaux. Ces articles sont des références pour ce mémoire et ils sont listés dans la bibliographie. Ce sont souvent des témoignages décrivant une façon de faire. Mais en terme d'analyse formelle, les résultats sont encore plus réduits. Il n'existe qu'un seul livre dédié à la question, *Production Pipeline Fundamentals for Film and Games* (Dunlop, 2014), et qui pose des questions générales et d'actualité. Il semble donc que ces questions fondamentales intéressent peu de monde, et seulement depuis peu. Nous parlerons par exemple de Bill Polson de chez Pixar qui a pris la hauteur nécessaire pour faire une analyse sémantique.

Mon intuition me fait dire que ces petites mains, pipeline TD et développeurs, architectes de ces solutions, sont toujours « en production », *la tête dans le guidon*, et n'ont pas l'occasion de prendre du recul. Les professionnels concernés tentent néanmoins d'organiser des groupes d'intérêts communs. Ce n'est que depuis 2011 que le sujet est traité au Siggraph lors de sessions informelles, les *Birds of Feather*. Mais les rares groupes de discussion sur internet ne sont pas très actifs en dehors des périodes de cette conférence. Enfin, on remarque que le milieu universitaire s'est

¹⁶Base de données d'articles et publications de recherche de références autour des questions de l'image de synthèse

rarement intéressé à cette question, qui nécessite une immersion profonde dans la production et des cas pratiques et s'étudie difficilement dans un cadre plus théorique. Dans ce contexte, à part quelques expériences isolées (O'Neill, Mavroidis, & Ho, 2007), la thèse d'Anne-Laure George-Molland, à ATI en 2007, et rendue possible par un doctorat en CIFRE et au contact de cas réels, fait figure de précurseur.

I.4. Conclusion de la première partie

L'art et la technologie sont intimement liés dans l'audiovisuel. Les oeuvres créatives sont le résultat d'un long processus de support direct et indirect aux créateurs. Quand on en vient à parler d'image de synthèse, c'est d'autant plus vrai que notre outil de travail est un ordinateur et des logiciels 3D, purs concentrés de technologies. C'est la somme d'innombrables générations d'ingénieurs, de chercheurs, de techniciens, de petites mains, ainsi que tout le système qui nous permet de les faire tourner (réseaux énergétiques et de communication). À des échelles différentes, cette constatation est vraie depuis le début de l'image animée.

Passé ce constat initial vertigineux, mais acquis, l'oeuvre de collaboration audiovisuelle pose ses fondations sur un ensemble de processus qu'on appelle le pipeline. Ce pipeline est cette base invisible qui tient la chaîne de fabrication et qui permet l'essor de création, en grandes quantités, d'images de synthèse. La création peut continuer à chercher ses formes les plus diverses, sur toutes sortes de supports, parce que des architectes veillent à ce que cette chaîne polymorphique continue à s'adapter. Que font-ils précisément ? C'est ce que nous allons voir dans les deux prochaines parties. Nous essaierons aussi, à travers des exemples, de clarifier le champ d'action du pipeline.

Partie II. Un ensemble de problématiques

Maintenant que l'on a pu survoler le contexte historique et technique dans lequel on met un place un pipeline, nous allons pouvoir nous interroger sur les grands concepts. Qu'est-ce que le minimum nécessaire dans une oeuvre de collaboration ? Qu'est-ce qui pousse des graphistes à structurer leur chaîne technique ?

Nous regarderons les spécificités de développer du code de support pour une oeuvre de collaboration artistique qui, par définition dans son processus créatif, est instable et imprévisible.

Puis j'essaierai de classer les grandes tendances d'outils qui existent et quelle est leur utilité dans ce pipeline. On se demandera si ces différentes familles de solutions sont accessibles à tous.

II.1. Initier un pipeline

Les besoins en pipeline ne se font pas forcément sentir en début de vie d'un studio d'animation. En effet rares sont les studios qui se mettent en place autour de projets d'envergure. On commence plutôt par faire de petits formats unitaires, comme des courts-métrages, de l'institutionnel ou du publicitaire. Bien que parfois poussés techniquement ou artistiquement, ces projets peuvent être faits avec une approche élémentaire de l'organisation. Ce n'est que quand des structures qui commencent à produire du volume (en quantités d'assets ou d'images à fabriquer), avec des équipes de plus en plus conséquentes, que les besoins d'organisations techniques s'imposent.

Dans le meilleur des mondes, ce ne sont pas que les architectes qui vont coder le pipeline qui participent à sa conception. La direction du studio donne le cap et la stratégie à moyen long terme. Les superviseurs et les directeurs de production expliquent comment ils veulent gérer la fabrication jour après jour. Des responsables créatifs de chaque département expliquent les besoins des artistes et techniciens travaillant avec eux. Enfin l'équipe technique, les IT, les développeurs, les TD, seront ceux qui pourront dire ce qui est faisable tant le pipeline impacte toute la chaîne technique.

II.1.1. Par quoi on commence ?

Quand on observe ce qui motive les pipelines de différents studios, on remarque qu'il n'y a pas vraiment de priorités universelles établies. Il est difficile de dire quelle est la première brique qui doit être mise en place par l'équipe de R&D. Tout dépend du projet.

Un pipeline est là pour permettre la fabrication d'un projet aux propriétés qui lui sont propres : la technique utilisée (2D, 3D, mixte, ...), les technologies choisies, le format et la durée (clip, court, spécial, long métrage, série, ...), la quantité d'assets et l'exigence artistique ou commerciale (*Adama ? Les Minions ? Avatar ?*), les délais de fabrication, les moyens humains (a-t-on seulement des TD prévus sur le projet ?), les moyens matériels, les supports de diffusion (on n'exige pas la même qualité pour le web, la télé ou le cinéma et la distribution internationale) et, bien entendu, le budget (de production et de distribution).

Chaque film est unique. Chaque projet est un prototype qu'on ne reproduira pas deux fois. Même dans la série, on optimise pour faire de la réutilisation, mais deux épisodes ne sont pas identiques. Et à l'intérieur d'un même projet, deux séquences peuvent avoir des modes de production différents à cause de leurs particularités.

Une structure de quelques graphistes et une structure de plusieurs dizaines ou centaines de graphistes auront des problématiques très différentes à gérer. On verra qu'il existe différents types d'outils, qui vont de l'outil d'automatisation et confort aux outils fondamentalement structurants.

On peut néanmoins percevoir que quelques notions seront incontournables quelque soit la taille du pipeline. C'est le cas de la question des entrées/sorties, c'est-à-dire la manière dont on reçoit du contenu (que ce soit un montage, des plans bruts et non compressés provenant d'un tournage, un cache d'animation, etc.) et ce que l'on va en faire, comment on les livre.

Une autre notion importante, au moment de réceptionner, modifier et transmettre ces données, c'est de mettre en place des méthodes d'identification. C'est-à-dire de trouver des règles acceptées par les intervenants pour nommer les tâches, les assets, les plans ou toute autre unité de travail.

Ce que le client appelle « LE MOULIN ROUGE », est-ce la séquence 12 ou bien un asset (une modélisation du célèbre cabaret parisien) ? Si c'est la séquence, sa nomenclature peut varier : « S12 » chez Les Fées Spéciales en Layout et animation des foules, « SEQ12 » pour l'équipe décors et « s012 » chez Mac Guff Ligne en charge de l'animation. Il faut alors prendre en compte ces particularités dans les outils de communications/réception/livraison, et penser à mettre en place un lexique.

II.1.2. Donner un nom aux choses

En effet, trouver un glossaire commun est primordial quelque soit l'organisation ou le projet. Bill Polson de Pixar, dans sa conférence *CG Pipelines design patterns*, exprime le besoin de mettre en place ce lexique spécialisé (Polson, 2014) (Polson, 2015). Sa démonstration s'applique à plusieurs termes très génériques et l'on peut bien comprendre la démarche autour des termes *versions*, *revisions*, *variantes* ou *variations* et *représentations* souvent utilisées de façon interchangeable. Ainsi, les mots pipeline ou asset peuvent avoir plusieurs sens selon le contexte. Référence et dépendance, cache et bake, publish et install, peuvent aussi être des termes ambigus.

Il prône donc de créer des définitions plus rigoureuses pour tous ces termes. Donner du sens à chacun de ces mots. Nommer les différentes étapes, sous-étapes et niches pour partager un langage commun clair.

Par exemple si l'on reprend les termes liés au concept de *versions* nommés ci-dessus l'on pourrait les définir ainsi :

- **revision** : une évolution importante
- **variation** : une intention artistique différente (changement de texture par exemple) sans être une correction majeure
- **représentation** : façon de sauvegarder une même intention artistique dans différents formats (un fichier maya, un cache alembic et un fichier fbx du même asset, comme un .pdf et un .doc d'un même fichier texte)

Ce lexique et cette façon de nommer les choses vont de termes abstraits au nom des fichiers et dossiers en passant par les plus petites entités dans la structure interne, souvent complexe, des fichiers 3D. Par exemple, la plupart des studios ont des règles pour nommer les composantes de ces fichiers, qui peuvent être de nature différente : géométries, groupes, contrôleurs, shaders, bones, couches de rendus, actions, etc.

Bien les nommer permet de mieux les retrouver et de trier, ou d'appliquer des scripts sélectifs. Ça permet à une équipe de travailler avec moins de stress dû à l'interprétation du champ lexical de son voisin. Dans le cas des rigs, avoir une cohérence dans la nomenclature permet d'appliquer l'animation d'un personnage sur un autre par exemple. Et cela ne fonctionne que si des conventions communes existent au sein d'un projet.

C'est un problème auquel de hauts responsables (Grant et al., 2014) aimeraient trouver des solutions à une époque où les projets sont souvent éparpillés entre plusieurs entreprises d'animation ou effets spéciaux. Mettre en place ces standards de nomenclatures permettrait donc de simplifier les échanges. Car si un studio crée la géométrie d'un accessoire, comme une pomme, et le nomme PROP_Apple_GEO et qu'un autre studio nomme le même asset 02_Apple_MESH (en ayant choisi arbitrairement que 02 soit la catégorie des accessoires), l'entente peut être ralentie.

On reparlera de nomenclature en partie III. Mais c'est une question qui est un sujet de recherche à part entière.

II.1.3. Échanger des fichiers

Faire des effets spéciaux ou des films d'animation requiert la création et la maintenance de très nombreux fichiers, parfois des centaines de millions pour les plus gros projets. Mais même un projet de petite ampleur peut générer de nombreux fichiers. Qu'on soit seul ou en équipe, le résultat

d'une étape de fabrication génère souvent un fichier utile à l'étape suivante. La notion de flux de données, qu'on appelle *data-flow*, est donc une notion intrinsèque de la fabrication d'images animées. Par exemple une modélisation sera animée, mise en lumière, et compositée pour obtenir un résultat final dépendant d'une typologie de fichiers variés : fichier de géométrie, puis contraintes de déformations d'animation, caches de ces déformations, puis enregistrement de séquences d'images, qui seront améliorées (compositing ou étalonnage) et compressées dans différents formats de sortie...

Toutes ces procédures et les fichiers qui en découlent et qui constituent le dataflow sont aussi liés à un ce que l'on appelle les entrées/sorties, ou *input/output*.

Se mettre d'accord sur le nom de ces fichiers, leur contenu, leur format, leur structuration est donc inévitable. Mais c'est aussi qui fournit quoi et quand. C'est un exercice commun à mettre en place dès le début de production avec les autres partenaires et à l'intérieur d'une même équipe. Les séries de procédures qui en découlent ne doivent pas être contournées au risque de provoquer des problèmes. Souvent des guides techniques, ou des schémas sont établis par un coordinateur pour que tout le monde comprenne la procédure globale comme on peut le voir dans la figure suivante.

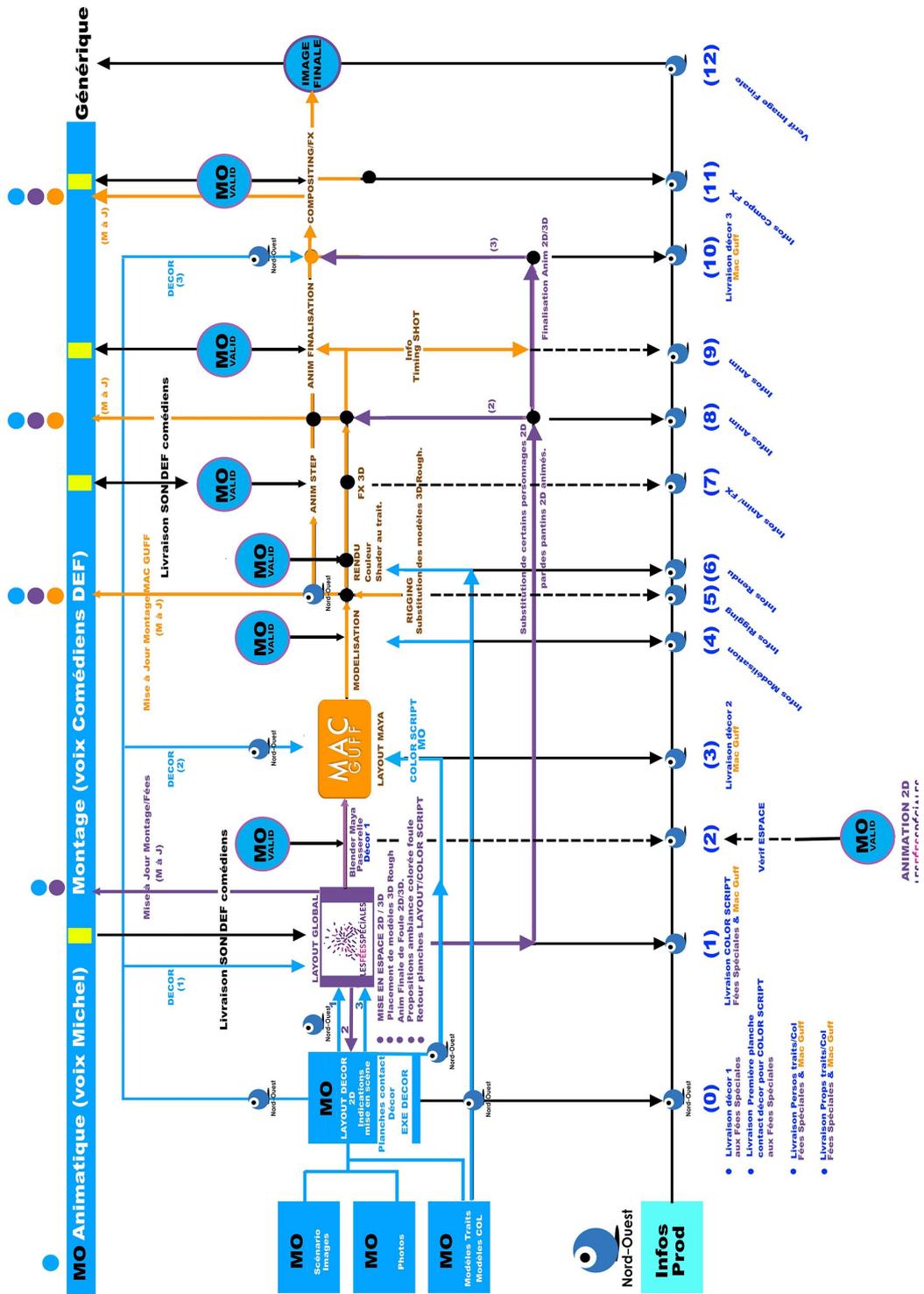


Illustration 2: Exemple de schéma mis en place en début de production pour (tenter de) clarifier les échanges entre les différents prestataires. Ici sur le long métrage Dilili à Paris

II.1.4. Prémices d'un pipeline

C'est souvent pour faciliter les questions de noms et d'échanges de fichiers que les premières briques de pipelines apparaissent autour d'un projet. Plus le pipeline mûrira plus on s'intéressera à garantir la sécurité des données (et donc à éviter de perdre du temps de travail humain).

Au début l'on va essayer de standardiser ces échanges. Par exemple, l'on va faire en sorte que les playblasts¹⁷ soient bien rangés, toujours à la même taille, avec un conteneur et un codec bien configuré. Ou en assistant le graphiste à enregistrer (automatiquement) le fichier au bon endroit sans se tromper de nomenclature. Ou l'on va assister le graphiste à toujours exporter un cache alembic avec la bonne configuration.

C'est la base de ce qui sera présenté plus tard dans la section sur les outils d'automatisation.

Mais le pipeline, en tant qu'ensemble de procédures visant à mener à bien un projet, peut aussi et même au contraire, quelque chose de non défini formellement, mais qui s'impose naturellement aux graphistes.

En effet, il n'est pas rare que la plupart des graphistes (étudiants ou professionnels) utilisent des scripts pour les assister à certaines tâches, en plus de ceux fournis via l'interface des logiciels, et ce sans savoir comment ils fonctionnent. Par exemple, beaucoup d'utilisateurs de Maya savent où est le script editor, comment on y colle une fonction (Mel ou Python) et comment on exécute (appelle) ensuite cette fonction. Les scripts utiles circulent avec une grande facilité au sein des équipes et on en trouve beaucoup en ligne (sur highend3d.com pour Maya par ex).

Il faut noter que cette circulation libre de scripts n'aide pas la mise en place d'environnements stables de travail. Cette remarque est vraie dans une configuration autonome (un graphiste chez lui) mais l'est encore plus dans un studio.

En effet, plusieurs questions doivent être posées lorsque l'on veut utiliser un script tiers. D'abord, est-ce qu'on a le droit d'utiliser ce script ou plugin¹⁸ ? Quelle est sa licence/coût d'utilisation, et est-ce une version légitime ? Ensuite est-ce que ce script est compatible avec la version du logiciel utilisé ? Est-ce que ce plugin est stable ? Est-ce qu'il vient avec un service après-vente (au moins un forum d'entraide) ? Quelle est son action réelle ? Est-ce que ce script a besoin d'une structure de données (organisation des fichiers et dépendances) qui n'est pas compatible avec l'organisation en

¹⁷.Aperçus d'animations aussi appelés line-tests, qui sont calculés rapidement

¹⁸Un plugin est un module d'extension à un logiciel existant.

place ? Est-ce que ce script n'altère pas les fichiers et leurs sorties au point de poser des problèmes plus tard ?

En équipe, toute intégration d'un nouveau script pose donc des questions et un contrôle doit être fait par l'équipe technique pour éviter de mauvaises surprises plus tard en production. Et chez soi, il est bien sûr tout aussi bon de se poser ces questions.

II.2. Particularité des productions

II.2.1. L'impossibilité de prévoir et le flux tendu

Un pipeline est difficilement prévisible. Parmi les équipes qui les mettent en place il est coutume de dire « *la seule chose qui est constante : c'est le changement* » (Dunlop, 2014).

Steve Lavietes, de chez Sony Imageworks, considère qu'il faut imaginer un pipeline en production comme « *un avion qu'on reconstruirait en plein vol* » (Grant et al., 2014). L'analogie me paraît adéquate.

Il est en effet très difficile d'anticiper les besoins d'un projet. On peut établir les grandes lignes, choisir un workflow et estimer le dataflow sous-jacent. Mais un projet et son pipeline sont un flux sans fin de changements incrémentaux. C'est comme surveiller un barrage et colmater les fuites au fur et à mesure avec une rivière qui change de trajectoire. Le pipeline parfait n'existe pas, il y a toujours des choses à améliorer, optimiser et accélérer et des exceptions à gérer. Un pipeline doit donc se caractériser par sa souplesse, son adaptabilité. Il est comme un moteur qu'il faut, a minima, entretenir et huiler de temps en temps.

Lorsque l'on fait un pipeline, l'on fait face à la réalité de l'avancement d'un projet. On remarque par exemple qu'une tâche devient bien plus compliquée que prévue, l'on cherche alors des solutions pour l'accomplir, et ce souvent dans l'urgence. Parfois l'on remarque qu'une procédure a mal fonctionné et l'on doit créer une procédure de correction.

Et les ressources étant limitées, on ne peut pas mettre en place tous les aspects d'un pipeline en parallèle. Il faudra donc faire des choix stratégiques et voir quelles sont les urgences, c'est-à-dire prioriser. Ces choix peuvent par exemple être motivés par l'ordre de mise en fabrication, ou la mise en place de fonctions indispensables au détriment du confort des utilisateurs.

De plus, on a rarement le luxe d'anticiper, faute de budget. Tant que le financement n'a pas été accepté, les producteurs n'ouvrent pas les crédits. Et lorsque le financement est verrouillé, il faut

commencer à fabriquer tout de suite. On se retrouve donc souvent à poser les fondations du chantier alors que les maçons fabriquent déjà les murs de la maison et que les peintres sont en route. Résultat, on ne fait souvent que parer au plus pressé.

Si l'on arrive à mettre en place un pipeline, avec son organisation et ses règles, il faut être prêt à les contourner. On parle de micropipeline [prod fondamentals], des procédures spécifiques et sorties du cadre général qui nous rappellent que les règles sont faites pour être brisées. Il faut essayer de ne pas généraliser, mais c'est inévitable et les architectes des pipelines ne doivent pas l'oublier. Parfois ces micropipelines prouvent leur utilité à long terme et rejoignent la structure globale. Dans ce genre de cas, on va assainir l'outil. Par exemple à Mac Guff il arrive que des outils développés rapidement par les TD prennent trop d'importance dans la production. Ils repassent alors sous le contrôle de l'équipe des développeurs qui reprennent et stabilisent son intégration dans l'écosystème de Mac Guff.

Il est donc éminemment difficile d'établir un cahier des charges précis et anticipé d'un pipeline. Ce qui rend parfois la collaboration avec des ingénieurs et des développeurs difficile à concilier avec cette réalité. Or le milieu a besoin de ces profils qui ont besoin d'anticiper.

Le dernier point est la grande difficulté à estimer le temps de travail nécessaire pour développer des outils et donc faire des plannings précis, importants pour gérer la production. Car un outil peut être sans cesse amélioré. Il peut passer d'une simple ligne de commande à un ensemble d'opérateurs autonomes et réutilisables (on parle de *factorisation* du code). Il peut gagner une interface graphique plus intuitive à l'usage et ainsi gagner en fonctionnalités. Au regard de ces transformations, nous pouvons donc nous demander comment est-il possible d'estimer le minimum viable et d'anticiper les problèmes dans les algorithmes existants ?

Je n'ai pas de réponses. Depuis 2013 je me suis retrouvé en position de responsable de la mise en place de solutions de pipeline, au studio In Efecto puis au sein de la SCOP Les Fées Spéciales. Et j'ai remarqué être dans une grande difficulté lorsqu'il s'agissait de quantifier le temps nécessaire pour créer des outils et donc donner des réponses précises à la production. Alors que de l'autre côté, il me semblait plus facile de quantifier le travail nécessaire pour faire le film. Lors de la recherche bibliographique pour ce mémoire, j'ai été étonné (et d'une certaine manière soulagé), de voir que quantifier le travail des développeurs est un vrai champ de recherche. La conclusion est que c'est très difficile de donner des estimations (Lewis, 2001).

Bref, faire du pipeline peut sembler ingrat : il est difficile de prévoir quoi que ce soit, on répond au plus urgent comme des pompiers passant leur temps à éteindre des incendies. Dorian Février, TD sur *Les contes de la Nuit* de Michel Ocelot (2011) avait fait le dessin page suivante pour résumer la fabrication. Il illustre bien ce qui était prévu ("pipeline d'origine"), et l'intense travail nécessaire pour essayer de maintenir le bateau à flot avec des solutions de bric et de broc.

II.2.2. Brassage des idées et héritage

Comme le souligne déjà en 2007 Anne-Laure George-Molland dans sa thèse, la particularité de ce milieu qui est d'utiliser des travailleurs temporaires "au projet", crée un véritable brassage au sein des entreprises. L'effet est même accentué en France avec le système de l'intermittence. Là où dans d'autres pays on essaie de maintenir des équipes permanentes, en France les graphistes vont et viennent plus facilement et ils rapportent avec eux les connaissances acquises dans les autres studios. Ils facilitent donc l'échange des solutions et des habitudes (bonnes ou parfois mauvaises du point de vue de l'organisation établie). Ils participent aux avancées techniques et artistiques des studios qui progressent sans cesse grâce à ce brassage.

Mais par la même occasion l'on peut aussi se demander s'ils ne contribuent pas à une certaine normalisation des méthodes, par croisements successifs. Des standards *de facto* se mettent en place (« c'est comme ça que tout le monde fait ») et peuvent être difficiles à contrer quand on cherche des solutions innovantes. C'est un biais dans lequel je me retrouve, avec parfois cette difficulté à remettre en question le *status quo*. Certes, la standardisation des pipelines sert à faciliter et coordonner les productions, surtout celles divisées sur plusieurs sites, mais il faut savoir contourner les règles pour pouvoir innover.

Toutefois, les équipes techniques font souvent partie des rares postes permanents d'un studio d'animation, et elles essaient de maintenir leurs acquis. Entre deux projets, dès qu'il y a une occasion, on fait le tri et on nettoie le code pour le rendre générique (applicable à plusieurs projets). Nous verrons d'ailleurs la question de la pérennisation des pipelines (partie III), car l'on ne recommence bien sûr pas tout de zéro à chaque projet. Petit à petit, au sein d'une même structure, le pipeline murit donc et devient de plus en plus efficace. Génération après génération on garde les solutions efficaces et l'on met en chantier de nouvelles mesures. Aussi faut-il maintenir ces équipes en place pour entretenir cet héritage, au risque de rompre la chaîne de transmission. On reparlera d'ailleurs de l'intérêt de faire des débriefings complets des projets terminés (partie II.2.4) avant que l'équipe ne soit éparpillée et occupée à autre chose.

II.2.3. Réservé aux gros studios ?

On a vu qu'établir un pipeline est quelque chose d'imprévisible qui demande donc des ressources pour réagir rapidement. On peut donc se demander qui a accès à ces équipes de support et à la mise

en place de ce socle technique. On le verra dans les types d'outils qu'il est possible de développer, qu'il y a clairement plusieurs niveaux.

Évidemment, les gros studios, vu l'ampleur de leurs projets, ont besoin de ces pipelines très développés et maîtrisés, et notamment dans les questions d'asset management (gestion des fichiers et leurs relations entre eux). Mais les petits studios, chacun à son niveau, doivent aussi trouver des solutions, plus ou moins organisées. Parfois il n'y a pas d'équipe R&D/TD/Pipeline réellement attirée, mais au moins un graphiste qui met en place des procédures partagées avec ses collaborateurs.

Mais une grande différence existe aujourd'hui : alors que dans les années 80/90 il fallait des développeurs, car il n'y avait aucun éditeur offrant des solutions commerciales, aujourd'hui ce n'est plus le cas. On trouve sur le marché des solutions « complètes », clés en main à la plupart des problématiques, même les plus lourdes. Se pose donc la question d'acheter plutôt que de développer. Les gros studios ont un héritage d'outils qui leur permettent de répondre aux projets ou bien ont assez de ressources pour continuer à développer des solutions personnalisées et adaptées. Il me semble que garder ces équipes de support est un avantage stratégique.

Les petits studios et les indépendants répondent plus facilement aux offres et solutions des éditeurs, comme le Creative Cloud de Adobe, ou les solutions toutes faites Maya-Nuke-Shotgun. Il paraît toutefois assez risqué de démarrer une production en misant sur ces packs. Les problématiques techniques sont trop nombreuses pour économiser des postes de TD. Beaucoup de petits studios cherchent à sanctuariser ces postes, soit en embauchant, soit en profitant de sous-traitants, TD et développeurs spécialisés, qui s'organisent en France (Kaleido-scop, KarLab, ...).

II.2.4. Postmortems

Le pipeline créé pendant une production a donc évolué de manière organique pour accompagner au mieux le projet. Une fois celui-ci terminé, dans la plupart des cas les entreprises se lancent sur de nouveaux projets, profitant du maintien des équipes et de leur savoir-faire. Les projets s'enchaînent parfois trop vite pour avoir le temps de prendre du recul et tirer les leçons d'une production. Et que le studio emploie des centaines de personnes ou une poignée, il est nécessaire de faire un retour d'expérience, débriefing global et par département, du projet une fois terminé, un **postmortem**¹⁹.

¹⁹Le terme, qui vient du latin "après la mort", est utilisé dans divers champs, notamment de la médecine légale, mais aussi dans le contexte de la gestion de projet où il signifie une analyse après la fin du projet

L'objectif des postmortem est d'apprendre de ses erreurs pour mieux anticiper les projets futurs. Il est en effet important que la mémoire des projets reste dans l'entreprise, à la fois pour la prise de recul de ceux qui ont vécu le projet, mais aussi pour les nouveaux arrivants. Car ceci permet à ces derniers de comprendre les choix actuels, en tirant au mieux parti de la culture de l'entreprise (comprenant les choix techniques et d'organisation), à travers les réussites et échecs du passé.

D'autant plus qu'en France, les équipes majoritairement composées d'intermittents sont très mobiles. En effet, rares sont les studios dont les équipes, et parfois même les responsables, sont les mêmes après quelques projets.

Le problème s'aggrave quand on sait que les membres d'une équipe ne sont pas présents de bout en bout : on n'a pas de compositeurs au début du projet alors qu'on n'a plus de modéleurs et rigueurs en fin de projet.

Arriver à mettre tout le monde dans la même pièce une fois le projet terminé pour faire un postmortem devient alors compliqué.

Pour *Azur et Asmar* (2006), Virginie Guilminot (Directrice de production) m'expliquait qu'elle allait voir les graphistes individuellement pour qu'ils fassent leur debrief. Cette procédure permettait de collecter les bilans, mais avait la limite de ne pas forcément partager ou croiser les informations afin que tous les acteurs du projet puissent en bénéficier.

Il est pourtant essentiel pour le succès du postmortem que les différents protagonistes comprennent, réagissent et rebondissent aux problèmes et aux idées exprimées par les autres. Par exemple, il est important que le département rigging comprenne les soucis qui se sont posés au modelling et connaisse les problèmes survenus en animation, car tout est lié dans un film d'animation et ce même s'il y a une fragmentation de tâches spécialisées.

Ed Catmull, le fondateur de Pixar, consacre un chapitre entier à ce sujet dans son ouvrage *Creativity Inc.* En plus de consolider ce qui a été appris et montrer à ceux qui n'étaient pas encore là, il distingue 3 autres raisons utiles à la mise en place de postmortem (Catmull & Wallace, 2014) :

- Ne pas laisser la rancœur prendre de l'importance : les erreurs viennent souvent d'incompréhension ou de problèmes face auxquels on a mal réagi. Si les gens ont un espace pour s'exprimer sur ces situations, d'une façon respectueuse, ils seront capables de mettre leur rancœur de côté et faire des propositions constructives.

- Utiliser le temps pour renforcer la réflexion : le temps passé à préparer un postmortem est aussi ou plus important que le temps en réunion. 90% de la valeur des échanges vient de ce temps de préparation
- Prendre de l'avance : dans un postmortem on peut poser des questions qui se présenteront au prochain projet. On n'aura peut-être pas les bonnes réponses, mais les bonnes questions permettront de mieux anticiper les problèmes.

Se pose ensuite la question de quoi faire de ces postmortem ? Quelles sont les traces que l'on doit garder et qui est responsable de l'organisation et de la conclusion de ce travail ? Chez Les Fées Spéciales on réfléchit actuellement à la forme de la synthèse et sa mise en disponibilité notre wiki ou notre système de forum interne.

À Illumination Mac Guff, chaque département fait son bilan et les comptes rendus, plus ou moins fournis, sont collectés par Étienne Pêcheux (responsable du département TD). Cette centralisation permet alors de faire des croisements utiles qu'Étienne utilise pour signifier aux différents intervenants de s'intéresser à telle ou telle question d'un autre département.

Aux Fées, ce n'est que récemment qu'on a fait nos premiers bilans. Pour le premier, pour un petit film animé de 30 secondes, fabriqué par une seule personne, j'ai tenu à ce que mes collaborateurs de l'équipe technique soient avec moi pour entendre ce que l'animatrice avait à en dire. Elle avait œuvré seule avec Blender et l'un de ses outils, le *grease pencil*. Elle n'avait pas eu besoin de nous. Mais ces deux heures de débriefing ont permis à l'équipe technique de tirer des conclusions sur l'utilisation de certaines fonctionnalités et d'imaginer des solutions pour le futur et d'autres projets.

Pour le second projet, *Antarctica*, le postmortem a été bien plus complexe. Le projet était plus long et beaucoup de leçons ont dû être tirées : de l'organisation des fichiers à la communication avec le client et le réalisateur en passant par la question du devis et de la facturation pour l'équipe de production.

Il a bien sûr été impossible d'aborder tous les sujets de front lorsque le projet était en cours. Et près de six mois après la livraison, ne pouvant pas réunir les conclusions d'ordre artistique et de production, seul le postmortem technique a pu avoir lieu. On est donc dans cette problématique de réunir les témoignages et retours avant que les intervenants ne soient trop occupés sur d'autres projets.

Pour l'instant ces discussions sont enregistrées sous forme d'archives audio ou de notes diverses. Mais il faudra prendre le temps d'achever l'exercice et d'archiver les conclusions.

Enfin pour le long métrage *Dilili*, et à l'heure où l'équipe des Fées achève et se retire progressivement du projet, nous avons lancé un fil de discussions sur notre forum interne pour récolter les avis. Les discussions sont variées et parlent de rig, de nomenclature, d'organisation et de communication. À partir de ces premiers échanges, nous prévoyons d'établir un ordre du jour pour mieux réunir les principaux intéressés afin de récolter les conclusions utiles. Ensuite, différents groupes pourraient préciser les discussions et lister les actions à mener pour le prochain projet.

Le pipeline doit tenir compte de l'ensemble des retours sur un projet et pas juste un débriefing sur les procédures qui le concernent. Car par effet de ricochet, certaines problématiques ou limites artistiques et techniques peuvent être liées à la façon dont le pipeline est organisé. Le regard sur le postmortem d'un projet dont le pipeline a fait office de fondation doit donc être global.

II.3. Différents types d'outils

Le pipeline représente les outils et les processus pour fabriquer un projet en image de synthèses. Maintenant que le contexte et le champ d'action sont établis, il est temps d'analyser les types d'outils et programmes que les équipes de support mettent en place.

Il existe plusieurs façons de cataloguer ces types de solutions. J'ai fait le choix, à partir de mon expérience personnelle, d'établir 3 groupes, qui répondent à 3 niveaux de maturation d'un studio :

- Les outils d'automatisation et outils de confort
- les outils qui vont de la simple collecte d'informations utile au suivi de fabrication
- et enfin les outils qui viennent structurer la gestion des fichiers : les asset manager

II.3.1. Outils de confort et d'automatisation : la base

Ce que je considère comme la première brique dans un pipeline ce sont les outils à destination des graphistes pour les délester de tâches redondantes. Cette catégorie d'outils est celle que les TD fabriquent le plus, car on les trouve à n'importe quelle étape de fabrication. Ces outils peuvent être très simples ou très complexes. Les exemples sont légion.

Pourquoi ce choix ? Parce qu'il semble que la plupart des graphistes qui se sont mis au script l'aient fait pour confier à la machine une tâche redondante, fatigués de la répéter encore et encore. Car la base d'un script est de lister une série de commandes pour pouvoir le re-exécuter²⁰ à la demande. Beaucoup de logiciels proposent des solutions simples pour créer cette série de commandes. Photoshop a un outil dédié bien pratique qui consiste à enregistrer des actions pendant qu'on les fait

²⁰On exécute un script dans le sens d'exécuter un plan

pour pouvoir les répéter; Mac OS offre un équivalent au niveau du file system, `automator`; et toutes sortes de logiciels offrent des `Macros` simples à appréhender. Les logiciels professionnels offrent généralement des solutions plus poussées pour donner des instructions au programme via un langage de programmation dédié (Maya `Mel`, 3D studio Max `MaxScript`, Houdini `VEX`, Renderman `Shading Language`, etc), et/ou un langage déjà répandu comme `Javascript` (Photoshop, Unity, ...) ou plus habituellement aujourd'hui une interface avec le langage `Python` (Maya, Blender, Nuke, Natron, 3DsMax, Houdini, ...).

Ce dernier s'est imposé dans la plupart des logiciels d'imagerie pour sa simplicité (la courbe d'apprentissage est plus facile que d'autres langages) et capacité à "tout faire bien" (mais rien "très bien") grâce à un catalogue de bibliothèques (nouvelles fonctionnalités) extrêmement diversifié. Le fait qu'un même langage soit utilisé par tous ces logiciels simplifie aussi le travail des TD.

Pour les besoins plus avancés, des langages plus *bas niveau* (proche du langage machine, dit « binaire ») peuvent être proposés, soit *managed* (C#, Java) soit *unmanaged* ou *langage système* (C, C++)²¹. Ceux-ci sont plus rarement utilisés dans les questions de pipeline et scripting d'automatisation et servent surtout à des tâches plus lourdes en calculs ou nécessitant un traitement bien plus rapide. L'équilibre à trouver se fait en prenant en compte la puissance et la performance du langage et son coût en temps de développement (cf figure suivante).

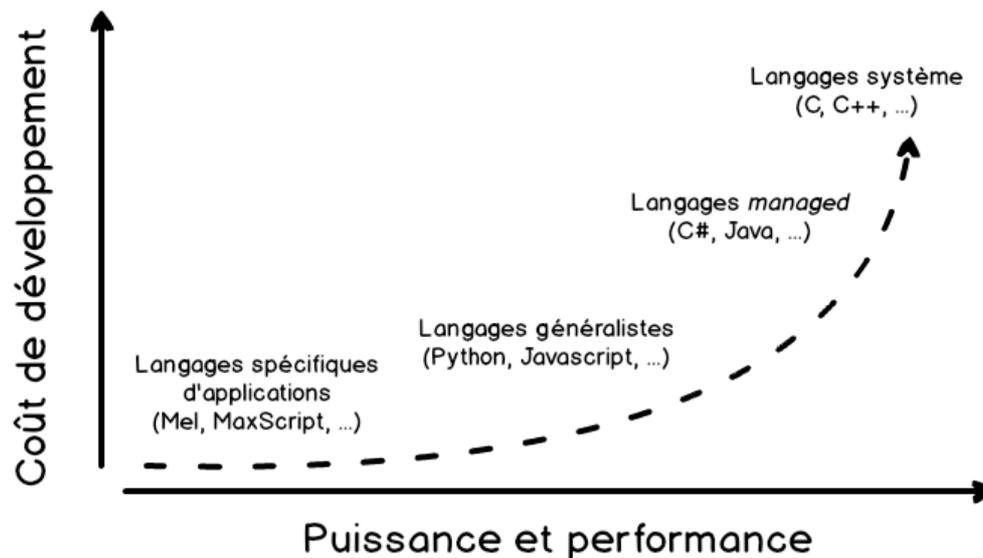


Illustration 4: Rapport puissance/coûts des grandes familles de langages utilisées dans les pipelines

²¹la répartition managed/unmanaged est une terminologie de Microsoft pour départager les langages qui s'exécutent dans une machine virtuelle (managed) et les autres (unmanaged)

La partie III. recense quelques exemples d'outils de confort et d'automatisation que j'ai eu l'occasion de concevoir.

II.3.2. Outils d'information : l'intelligence

Les outils qui récupèrent et manipulent de l'information sont souvent mis en parallèle, mais en dehors de la définition de pipeline. Car ces outils mènent généralement au suivi de production ou de fabrication, champ spécifique du chantier d'un projet audiovisuel. Mais pour moi, ce qu'apporte le pipeline à ces outils, et inversement, est tellement important que les deux vont de pair.

Le pipeline est alors plus que les simples fondations, et devient une source d'informations permettant de mieux superviser un projet. Ces informations, organisées et triées, permettent alors de prendre des choix intelligents pour continuer au mieux la production.

II.3.2.a Le pipeline comme source d'infos

Les graphistes qui travaillent sur des plans ou des assets, manipulent, parfois sans le savoir, de nombreuses meta-données utiles. Pouvoir les récupérer de façon automatique permet de ne pas les perdre.

Par exemple, un graphiste change le timing d'un plan, car l'animation nécessite 10 images de plus. Cette information est cruciale à bien des égards. Les outils qui préparent les caches doivent savoir que ces 10 nouvelles *frames*²² existent. L'équipe du montage doit prendre en compte ce temps supplémentaire. En fait tout le monde doit prendre en compte cette modification. Cette métadonnée, la durée du plan, est cruciale et on doit répercuter ce changement ou détecter un ajustement involontaire. Alors il faut que les outils que l'on met en place, d'une manière ou d'une autre, puissent récolter ces informations.

Au cours de mon expérience professionnelle, j'ai notamment travaillé sur cette collecte lorsque j'étais à Mac Guff. J'avais en effet créé un tableau de tous les plans d'une séquence qui allait dynamiquement chercher leurs durées dans tous les départements importants. Dans les fichiers EDL²³ du département montage, dans les scènes maya du layout, de l'animation et les fichiers Nuke du compositing. Ainsi l'on repérait tout de suite si un changement avait eu lieu et l'on vérifiait que tout le monde avait la même information.

²²Frame est le terme anglais pour "photogramme"

²³Edit Decision List : fichier standardisé qui indique les points de coupes, d'entrées/sorties d'une séquence et de ses plans lors du montage

L'exemple du timing d'un plan est simple, mais de nombreuses données plus ou moins *importantes* peuvent être récoltées. Par exemple on peut récolter le casting, c'est-à-dire quels assets (personnages, décors, accessoires) sont utilisés par les graphistes pour composer la scène. Si on détecte un problème sur l'un de ces assets plus tard il sera pratique de savoir dans quelles scènes le modèle défectueux apparaît. On peut suivre des questions de versions : savoir quelle version a été confiée à un autre département ou prestataire et à quel moment. Ou simplement le fait qu'un graphiste ait fait une proposition qui doit ensuite être visualisée et validée.

II.3.2.b L'arrivée de shotgun

Au cours des années 2000, le milieu des effets spéciaux atteint des niveaux très complexes de suivi de fabrication. Les plans truqués se comptent en milliers, et les assets sont encore plus nombreux. Les équipes d'effets spéciaux et d'animation atteignent parfois 1000 intervenants. Bref, le milieu s'est industrialisé et a besoin de revoir la manière de suivre la fabrication d'un projet.

Historiquement de grands tableaux `Excel` sont utilisés dans les studios ou même d'énormes tableaux imprimés et collés sur les murs des couloirs (voir figure suivante). D'autres studios se servent de documents partagés type `google spreadsheets`. Pourtant rares sont les studios qui utilisent des logiciels dédiés à la gestion de projets comme `Microsoft Project`. Ces outils n'ont probablement pas apporté de réponses utiles à la particularité de la complexité de la fabrication d'un film d'animation car leur système de visualisation n'est pas fait pour croiser et afficher autant de données.



Illustration 5: Le suivi de fabrication peut avoir diverses formes, ici des tableaux géants sur la production de Kubo au studio Laika (Academy Originals, s. d.)

C'est aux états unis qu'une petite équipe de développeurs a créé un outil qui va devenir le nouveau standard : shotgun.

Shotgun n'est rien de plus qu'une base de données dans laquelle on retrouve des entités définies (Film, Séquence, Plan, Asset, Utilisateurs, tâches, Notes, ...) auxquelles on peut ajouter autant de paramètres (colonnes) qu'on le souhaite. Ensuite un système de "pages", qui n'est rien de plus qu'une vue filtrée sur la base de données, permet de classer et croiser une très grande quantité d'informations.

Le système est efficient, répond aux besoins élémentaires et une API²⁴ permet à Shotgun de récupérer ou donner des informations à d'autres logiciels.

Shotgun est relativement simple au niveau de sa structure et le logiciel va vite s'imposer comme le standard de suivi de production. Tous les grands studios se l'approprient et en 2014 le logiciel est racheté par le géant de l'édition de logiciels Autodesk.

En 2007, alors que le studio parisien Mac Guff Ligne accueille les Américains d'Illumination Entertainment pour le lancement de la fabrication de *Moi, Moche et Méchant*, un des producteurs, Robert Taylor, ramène dans ces valises Shotgun. Étienne Pêcheux qui participe à la mise en place

²⁴Application Programming Interface : En informatique, une interface de programmation applicative est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels

technique du long, m'expliquera que Shotgun était ce qu'il avait toujours voulu mettre en place avec un outil interne appelé MGA (Mac Guff Administration). Étienne et un des développeurs, Damien Courreau, vont créer les connections nécessaires entre l'*asset manager* maison InK (cf Partie 2.C.3) et Shotgun. Shotgun est désormais la base de données dans laquelle on injecte toutes les métas données possibles pour qu'on puisse ensuite facilement les croiser, les trier et les chercher pour suivre la fabrication du long-métrage.

J'arrive à Mac Guff en 2009 et je ne participe pas les 6 premiers mois à Shotgun, qui ne sert alors qu'à l'équipe de production. Mais lorsque le studio commence à fabriquer le prochain long, *Le Lorax*, on confirme shotgun comme outil central de suivi de production. On me charge alors de créer un écosystème d'outils pour la production et les superviseurs, permettant à Mac Guff d'améliorer son utilisation de shotgun.

Shotgun et InK²⁵ sont les supports pipeline et suivi de production qui servent à fabriquer depuis 2007 des long métrage d'envergure tels que *Moi, Moche et Méchant* (2010), *Tous en scène* (2016) et une demi-douzaine d'autres. Et même si on trouve toujours d'autres solutions, il me semble que ce duo stratégique n'est pas étranger à la réussite d'une telle aventure. Ces films sont beaucoup trop gros et difficiles à fabriquer pour pouvoir tenir sur des solutions plus classiques.

II.3.2.c À la croisée des chemins : La Feuille de Temps

Faisons une petite pause à la croisée des chemins entre la réalité de la fabrication et les outils de suivi de production. Pour que l'équipe de production sache comment avancent les graphistes sur leurs tâches attribuées, il lui faut un outil en accord avec le workflow défini par le pipeline, car il est impensable d'aller collecter autant d'informations à la main.

La Feuille de temps est une solution de pointage du temps passé à oeuvrer à une tâche. C'est un outil mal aimé des graphistes et pourtant stratégique pour les studios. C'est la courroie nécessaire pour joindre la fabrication et la production.

La base d'un suivi de fabrication est de savoir ce qui est à confectionner et combien de temps a été passé pour l'élaborer (par exemple la durée de travail passée à animer un plan ou à modéliser un personnage). Il est important de le savoir, car un studio a un budget à respecter en fonction du devis qu'il a signé. Or, ce devis provient d'une estimation du temps de fabrication pour chacun des éléments à livrer. Ainsi, quand le projet est en production, savoir combien de budget a été consommé sur ces éléments permet donc de savoir où on en est. Est-on en avance ou est-on en

²⁵Un gestionnaire de fichiers en exploitation bien avant shotgun dont on parlera dans quelques pages

retard ? A-t-on alloué trop ou pas assez de temps par rapport au budget prévu sur telle ou telle tâche ?

Or, cette collecte d'information est ressentie comme un flicage par les graphistes. Les discussions sont souvent passionnées quand on évoque cette question. Pourtant savoir si trop de retard s'accumule et s'il y a un risque d'être hors budget est primordial pour maintenir le studio et ses emplois dans le futur. D'autre part, ce pointage permet aussi au graphiste de justifier son temps de présence et donc d'être payé de façon adéquate sachant que les heures supplémentaires sont monnaie courante.

Le degré d'information recherché est différent et dépend des outils et moyens humains pour traiter l'information. Du minimum (sur quel projet travail le graphiste qui peut être sur plusieurs projets au cours d'un mois) au plus détaillé (combien de temps le graphiste a passé sur la couleur des yeux d'un des personnages principaux). Les studios ont différentes approches et positionnements vis-à-vis de ces questions. Certains font circuler une feuille d'émarginement, d'autres se servent de solutions sous forme de pages web, et certains utilisent même des programmes qui analysent l'activité automatiquement (Simone, RADI, ou paper Siggraph)

On verra en partie III. la proposition que j'ai pu faire à Illumination Mac Guff.

II.3.3. L'asset management : la maturation

Quand un studio dépasse un certain seuil critique, avec trop de graphistes à gérer et d'assets et plans à construire, il faut mettre en place des solutions plus sécurisées pour que tout le monde puisse travailler en confiance. Plus on est nombreux, plus les projets sont complexes, et plus les risques d'erreur s'accumulent. Les studios doivent alors revoir leur structure et leur pipeline. Il faut dès lors des IT (l'équipe de support des questions de réseaux et stockage) qui gèrent des flux de données de plus en plus lourds, et il faut que les pipeline accompagnent cette charge. C'est en général à ce moment-là de l'histoire d'un studio qu'apparaît la notion d'**Asset Manager**.

Un Asset Manager est un programme chargé de gérer les fichiers d'une production. Il répond aux questions :

- de sécurité élémentaire (qui y accède ? Qui peut modifier les données ?)
- de relations entre les assets qu'on appelle des dépendances (de quels assets mon plan a besoin)
- d'historique (quelles versions existent ?)
- d'édition (quel programme -version et configuration- est nécessaire pour ouvrir ce fichier)

- et souvent d'interface pour l'accès aux données (où sont rangés les fichiers ?).

La mise en place d'une telle solution est lourde (il faut des ressources en développement et en IT) et donc réservée au stade de maturation évoqué plus haut. Par exemple Mac Guff ligne, studio fondé en 1988 n'a mis en place un tel outil qu'après avoir fait le long métrage *Azur et Asmar*. Constatant la complexité du travail effectué et anticipant les besoins futurs, l'équipe technique crée InK. La première version a été utilisée pour le long-métrage *Chasseur de Dragons* (2008) et la fin de la série *Pat et Stan*, avant de venir participer avec succès à la conception complexe de *Moi, Moche et Méchant*. Il aura donc fallu 20 ans à ce studio pour s'équiper d'un tel outil.

Toujours en France, dès 1999, le studio Chaman avait détourné le logiciel *Alien Brain*, dédié au jeu vidéo, pour gérer ces questions au cours de la fabrication du long métrage *Kaena la prophétie* [these ALGM].

II.3.3.a Locker les assets et ne pas impacter les autres

La plupart des asset managers offrent au moins 4 fonctionnalités essentielles quant à la relation entre l'utilisateur et un asset :

- Verrouiller l'accès aux données pour éviter que deux graphistes travaillent en même temps sur un fichier
- Récupérer les fichiers nécessaires pour y apporter des modifications dans un environnement qui n'impacte pas les autres utilisateurs
- Sauvegarder et partager avec les autres ces modifications
- Déverrouiller l'accès

À noter que plusieurs termes existent pour définir ces fonctionnalités selon les asset managers. Ceux en utilisation chez Mac Guff dans InK sont respectivement : *lock* (verrouiller), *grab* (récupérer), *publish* (publier) et *unlock* (déverrouiller).

Pourquoi ces fonctionnalités sont-elles importantes ? Par exemple, lors que l'on n'a pas d'asset manager et que les données sont en accès libre sur un réseau aux graphistes, il est impossible de savoir si deux graphistes ne sont pas en train de travailler sur le même fichier. Le premier qui sauvegarderait ses modifications perdrait tout quand le second viendrait écraser le fichier en sauvegardant son propre travail.

De plus, on va le voir dans la sous-partie suivante, un fichier, un asset, peut être appelé par un autre asset. Si un graphiste travaille sur un matériau et enregistre son travail au fur et à mesure, mais qu'une scène de rendu en cours appelle ce fichier, on risque de trouver des incohérences d'une

image calculée à l'autre. Pire, si un graphiste travaille sur un rig de personnage et qu'une sauvegarde intermédiaire casse le rig pendant qu'un animateur s'en sert, on risque de perdre le travail de ce dernier. Il est donc primordial d'offrir pour chaque tâche un espace de travail, déconnecté du reste de la production, pour ne pas impacter les autres.

Le minimum qu'on peut faire quand on n'a pas d'asset manager est de définir une nomenclature pour le fichier qui sert de référence. On va travailler 2 ou 3 versions par exemple (v1,v2 et v3) et définir la v3 comme étant la nouvelle référence (vref). On peut ensuite faire une v4, v5 et ainsi de suite tranquillement si besoin, et ce jusqu'à juger qu'une nouvelle version de travail puisse devenir la vref en remplacement de l'ancienne.

Quant aux droits d'accès, certains logiciels verrouillent par défaut l'accès au fichier en cours d'édition. Mais ce n'est pas systématique et même une bonne organisation et répartition des tâches risque de se retrouver en situation de conflit. Il vaut mieux prévoir de ne travailler que dans un espace personnel, et copier les dernières modifications dans l'espace partagé. En attendant d'avoir un véritable asset manager qui s'occuperait de verrouiller les fichiers automatiquement, voilà un bon candidat pour un outil d'automatisation.

Dans tous les cas, asset manager ou pas, un bonne sauvegarde est toujours nécessaire. Les risques d'erreurs humaines ou matérielles sont très importants, et des procédures de backup solides sont primordiales dans tout projet. De gros problèmes d'infrastructures peuvent avoir lieu, j'en ai été témoin. Et même les grands studios, comme Pixar, ne sont pas à l'abri²⁶ d'une réaction en chaîne.

II.3.3.b Les dépendances

Un asset manager est aussi responsable de la description des relations et des dépendances entre deux assets. Dans un outil comme InK, où tout élément est considéré comme Asset (de la modélisation à la séquence d'image qui sort du compositing), cette relation est très importante et se définit par deux aspects :

- tout d'abord son caractère, soit constant (toujours nécessaire), soit temporaire (nécessaire une fois).
- sa direction, c'est-à-dire dans quel sens un asset se réfère à un autre.

Cette information de liaison est importante à définir correctement. Une boucle (une suite de connexions qui reviendraient au point de départ) serait un grave problème (entraînant une mise à jour potentiellement cyclique et infinie).

²⁶thenextweb.com/media/2012/05/21/how-pixars-toy-story-2-was-deleted-twice-once-by-technology-and-again-for-its-own-good/

Illustrons cette notion de flux avec un asset de 'compo-render' tel qu'on peut le voir dans la figure ci-dessous. Il contient une séquence d'images résultant des calculs opérés au compositing par un asset de 'compo' qui est un fichier Nuke. Ce fichier Nuke a comme dépendance les séquences d'images des différentes couches qui sortent du lighting et qui sont nécessaires au compositing. On peut continuer à remonter le courant en amont, étapes par étapes, pour aller chercher les assets auxquels ce fichier Nuke est rattaché. C'est ce qu'on appelle dans InK un *upstream*. Mais on peut aussi redescendre et aller en aval, le *downstream*. Depuis notre séquence d'images qui sort du compo on pourra trouver un asset "compo-movie", qui est un fichier quicktime .mov généré à partir de la séquence d'image dont il dépend.

Il faut ajouter les besoins en stéréographie, c'est-à-dire le relief, avec l'oeil gauche et l'oeil droit (indiqués respectivement -L- et -R-) qui doublent parfois les assets.

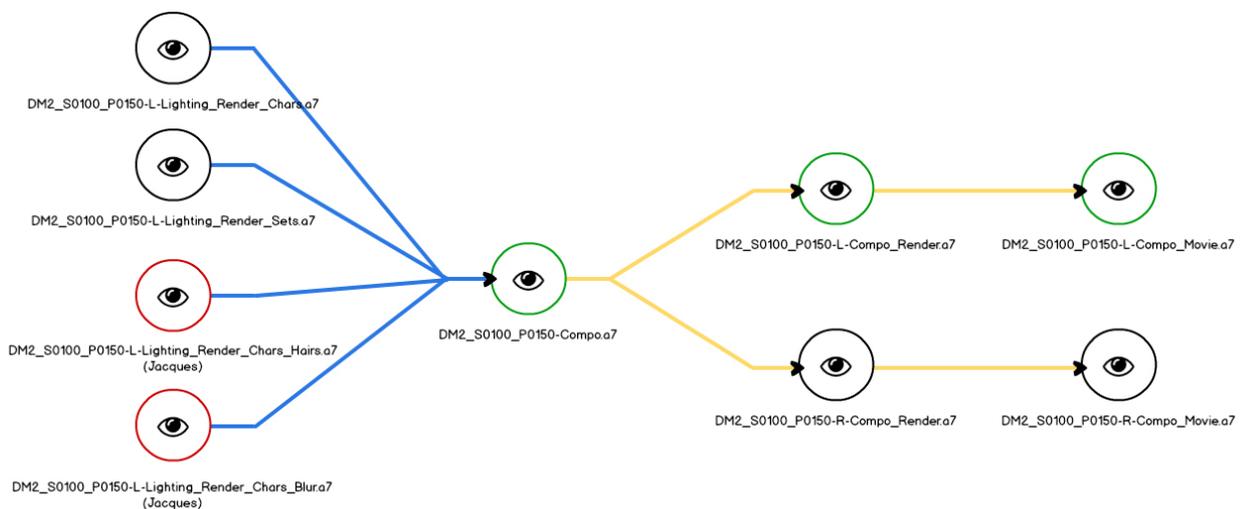


Illustration 6: Maquette d'un graph InK : relation entre plusieurs assets de compo (les sources du rendu, le fichier de compo, des rendus de compo puis des movies de contrôle). En Rouge les assets lockés par un autre graphiste, en vert ceux lockés par moi et en noir ceux que personne n'a locké. Les liens bleux sont des dépendances et les liens jaunes des références.

Ensuite il y a une notion d'obligation : celle d'avoir ou pas cette relation active. Il faut noter que certaines de ces relations sont toujours nécessaires : un fichier d'animation ne peut pas marcher sans le rig du personnage ou le fichier son pour connaître le dialogue à animer. Et un matériau a toujours besoin d'accéder aux textures qui le composent. On appelle ces relations des *dependencies*. Alors qu'au contraire d'autres liaisons sont optionnelles et peuvent ne servir qu'une fois. Lorsque l'on a fait notre playblast de contrôle (un aperçu vidéo d'une animation) ou que l'on génère un cache de fluides (pour les effets spéciaux) on n'a plus besoin de la scène d'origine et du logiciel 3d (maya, blender, houdini, ..) mais juste d'un lecteur vidéo (VLC, RV, dJv, QT ...) ou d'un interpréteur. On appelle ces

liaisons temporaires des *references*. Chez Mac Guff, dans InK ceci est symbolisé par des liens de couleurs différentes.

Enfin, il n'y a pas de corrélation entre la tâche d'un graphiste et le nombre d'assets. Une tâche, par exemple au département du *character Fx* ou CFX (animation dynamique des vêtements et cheveux) peut nécessiter plusieurs assets de InK : des rigs spécifiques, des caches venus du département animations, différents assets de bake du Cfx, et des movies pour contrôler le tout. On crée alors des "graph" qui permettent de sauvegarder l'ensemble des assets nécessaires à une tâche donnée sans devoir les retrouver un par un. Sachant qu'à partir de n'importe quel asset on peut demander au logiciel de nous montrer les liaisons en aval et en amont.

II.3.3.c Cache de géométrie

Un des derniers points techniques qui prouve la maturation d'un pipeline est probablement une maîtrise cohérente de la chaîne de la géométrie, en particulier lorsqu'elle est animée.

Le problème est particulièrement vrai lorsque l'on utilise des animations utilisant des comportements dynamiques (automatiques). Par exemple lorsque la course d'un chien fait bouger ses oreilles automatiquement. Si cette animation est utilisée en rendu des problèmes d'interprétation de la dynamique peuvent surgir et saborder les calculs (faisant perdre de longues heures de traitement). Dans ce genre de situation, l'utilisation d'un *cache* ou *bake* règle les problèmes. En effet l'animation est "écrite" une fois pour toutes et le moteur de rendu n'a plus à interpréter des calculs dynamiques.

Ce système utilisé à travers toute la chaîne de fabrication assure que la géométrie utilisée en entrée est stable. Car un cache est une vue précise à un instant T. Pour donner une analogie, cette vue n'est pas altérée par des paramètres extérieurs comme pourrait l'être un PDF vis-à-vis d'un fichier .doc. Dans ce cas le PDF est interprété de la même façon par tous les lecteurs alors qu'un .doc pourrait être représenté différemment selon l'éditeur ou la disponibilité des typographies.

La notion de cache de géométrie n'est pas récente, mais des standards se sont imposés génération après génération. On est passé du `point cache` au FBX en passant par des solutions `.obj`. Aujourd'hui c'est le format `alembic` (`.abc`) qui s'impose dans toute l'industrie.

II.3.3.d De la centralisation à la décentralisation

C'est en m'intéressant au développement généralisé de l'aspect multisite d'un même projet (c'est-à-dire l'éclatement sur plusieurs sociétés, et ce parfois dans plusieurs pays du projet) que je suis tombé

sur l'analyse de Richard Chuang et David DeBry présentée à Siggraph Asia (Chuang & DeBry, 2009). Leur conférence me sert de base pour décrire les 3 grandes catégories d'organisations des équipes et moyens autour d'un projet. C'est leurs termes que j'ai repris, abandonnant le terme simpliste de multisite pour décrire ce genre de productions éclatées..

Centralisées

C'est la forme la plus traditionnelle : un studio, un seul site (physique), un ou plusieurs projets, une équipe. C'est un modèle particulièrement utile face à des cas de nécessité tels que taille contrôlée de la production, la disponibilité des équipes ou la communication interne particulièrement importante pour une équipe novice. Ce type d'organisation est limité par la place disponible et partage un même pipeline, une même infrastructure technique (postes de travail et réseau) et logiciel.

La maintenance technique peut être simplifiée sauf si le pipeline commence à devenir spécifique en fonction de chaque projet lorsque le studio mûrit et commence à prendre des projets en parallèle. D'autant qu'une même équipe qui passe d'un pipeline à un autre peut être compliquée à former. Mais au moins tout le monde est logé à la même enseigne (parc informatique et environnement contrôlé) ce qui permet à l'équipe technique d'avoir un suivi facilité. La gestion de production est centralisée avec des solutions traditionnelles et surtout au contact direct des ressources (humaines et matérielles).

Ce modèle a néanmoins des limitations d'évolution et d'adaptation. Et passer d'un format à un autre (du long 3D au court 2D) peut être problématique.

À titre personnel, c'est le modèle que j'ai le plus vu et dont j'ai participé à la mise en place, notamment chez Les Fées Spéciales. Ce dernier point, celui de la capacité à s'adapter, est souvent le plus problématique et demande beaucoup d'effort à l'équipe technique et aux équipes de collaborateurs.

Semi-décentralisées

Un même projet, mais sur plusieurs sites (d'une même ou de plusieurs entreprises), mais où chaque site fonctionne de manière centralisée. C'est le modèle de plus en plus présent depuis une dizaine d'années dans les effets spéciaux ou dans les coproduction d'animation (en Europe notamment pour des questions de montages financiers). Certaines compagnies ont créé tellement d'annexes qu'on dit que *le soleil ne se couche jamais* sur ces structures²⁷.

²⁷À l'instar de l'Empire britannique qui s'étendait sur toute la planète, certains studios sont présents sur tous les continents et il y a toujours des employés à l'oeuvre quelque part dans le monde quelque soit l'heure de la journée

Car le modèle est souvent issu de gros studios qui ouvrent des satellites dans d'autres pays et sous-traitent une partie de la production sur ces sites. Les motivations peuvent être d'ordre économique (coût de la main-d'oeuvre et crédits d'impôts), humain (saturation du marché à un endroit et manque de main-d'oeuvre), logistique (au-delà de 1000 personnes sur un seul site, les équipes semblent saturer) et pratique (pouvoir avoir un travail continu pour produire plus vite). Le modèle décentralisé peut aussi s'appliquer à plusieurs studios proches les uns des autres, mais indépendants (comme plusieurs compagnies).

Les équipes suivent le modèle centralisé et passent rarement d'un site à l'autre (pour des questions de distances ou d'indépendance des sites).

Chaque site utilise son propre pipeline, et son propre ensemble de logiciels et infrastructures techniques. Le projet peut être synchronisé entre deux localisations, mais chaque site a ce qu'il lui faut pour travailler. Dès lors, des questions de nomenclature peuvent se poser ainsi que de continuité de l'intégrité des assets (chaque studio peut avoir une approche très différente).

Les projets étant déjà divisés par nature, on peut plus facilement faire intervenir des sous-traitants pour répartir la charge de travail ou travailler sur un point très spécifique et spécialisé. La gestion du projet est en général globale via un délégué, et ensuite chaque studio a sa propre façon de suivre la fabrication. Ce modèle nécessite beaucoup d'organisation, notamment contractuelle, pour que les choses avancent dans le bon ordre au sein de chaque entité participante.

C'est le modèle en place pour le long métrage *Dilili à Paris* où 4 entités sont en charge de la fabrication de l'image.

Complètement décentralisées

C'est un modèle émergent, car il n'est viable que parce qu'une technologie particulièrement pointue le permet (notamment une bande passante d'internet toujours plus performante). Dans cette organisation la notion de studio et d'équipe est complètement obsolète. Il n'y a plus de nécessité d'avoir un bureau central ou d'un *filesystem*, les membres de l'équipe peuvent travailler de n'importe où (Vanns & Carey, 2016). Ces membres peuvent travailler à n'importe quelle heure, dans différentes langues. Ils sont mobilisés en fonction de la dynamique du projet (par exemple selon leur expertise ou leur coût) de façon flexible en fonction des besoins. A priori il y a moins de problématiques de formation puisque les membres sont mobilisés pour leurs connaissances déjà acquises.

Néanmoins, c'est le nécessaire retour à un seul pipeline central qui connecte tous les intervenants. Des services de mise à jour à distance sont nécessaires pour maintenir l'homogénéité d'un parc au départ hétérogène. Les données quant à elles peuvent être à plusieurs endroits grâce à des solutions de type *cloud*²⁸. Et tout le système doit reposer sur une plateforme de distribution automatique, simple et invisible.

Dans ce modèle, le projet n'a pas de limite de forme (tous types de projets sont acceptés, le pipeline et les ressources s'adaptent automatiquement). Son échelle peut être très variable et n'est limitée que par le budget, l'ambition et la capacité à trouver les participants adéquats.

Par contre, la question de la gestion de production devient très complexe. Là où des modèles centralisés ont des points de référence, il est hors de question pour un graphiste au Japon d'attendre que l'équipe de production soit disponible en France pour savoir quoi faire. Il faut donc revoir le modèle autour d'assets qui ont assez d'informations embarquées pour alimenter les intervenants. C'est toute la question de la communication, dans son ensemble, et de la cohérence, qui sera le plus grand défi de ce modèle décentralisé.

À titre personnel, c'est un modèle qui m'effraie un peu dans le sens où la précarité des intervenants va devenir plus forte au profit d'un système de mercenaires. Par contre pour des équipes indépendantes, qui n'ont pas accès à la logistique habituelle (bureaux) ou veulent travailler par affinité, mais à distance pour ne pas déménager, c'est une voie très intéressante.

Quel modèle donc ?

Dans tous les cas le pipeline reste fondamental. Mais l'émergence du modèle décentralisé force à réfléchir. Pour certains de nos projets aux Fées Spéciales, nous avons des graphistes qui travaillent à distance. Leo Silly-Pellisier, animateur et ancien d'ATI, travaillait ainsi dans les studios de Mac Guff mais avec un pipeline identique au nôtre et des données synchronisées avec notre serveur. Le projet était donc en double, ce qui a posé des contraintes d'infrastructure et de bande passante importante et s'est apparenté à un modèle semi-décentralisé. Il est à noter qu'il est préférable de ne pas généraliser ce modèle, car il forcerait chaque intervenant à avoir localement des milliers de gigaoctets de données.

²⁸Le *cloud-computing* est l'utilisation de la puissance de calcul et du stockage disponible sur des serveurs distants, c'est-à-dire des services et solutions disponibles en ligne, par exemple les google docs, dropbox, les services numériques S3 d'Amazon, etc.

Un autre cas de travail à distance était celui de graphistes travaillant de chez eux et échangeaient via de simples FTP²⁹. Ce qui revenait à centraliser les données, mais faisait perdre du temps au coordinateur et aux intervenants.

Nous avons mis en place ce type de travail délocalisé, car depuis Montpellier il n'est pas aisé d'avoir les graphistes nécessaires sur place alors que nous faisons partie d'un petit milieu qui se concentre essentiellement en Île-de-France. Pour nos prochains projets, le modèle décentralisé risque de s'imposer et ce pas forcément selon le choix du studio. Parfois, ce seront les choix de vie de nos collaborateurs qui nous l'imposeront quand d'autres fois ce sera la disponibilité des ressources humaines. Nous devons donc avoir une réelle réflexion quant à la manière d'aborder au mieux ce casse-tête.

Les VPN³⁰ par exemple pourraient constituer une voie possible, mais la bande passante, nécessaire pour manipuler des données lourdes, fait que les VPN ne sont pas encore *la* solution. Quand on aura de vrais réseaux asymétriques de 1Gb, par exemple pourraient constituer une voie possible de repenser ce genre d'options qui nécessitent tout de même une connexion sans interruption.

II.4. Conclusion de la seconde partie

On a vu que les outils de pipeline ont plusieurs formes et apportent des solutions plus ou moins structurantes pour faciliter la création d'oeuvres de collaboration.

Les studios doivent s'équiper de pipelines pour répondre à leurs besoins spécifiques. C'est d'autant plus nécessaire quand les studios dépassent le stade de la *start-up* et commencent à recevoir des commandes et des productions de plus en plus ambitieuses. Certains studios vivent mal cette transition, prenant d'urgence des pipeline-TD pour palier à un manque historique. Le type d'outils créé dépend donc en partie de la maturité de ce studio et de son besoin de pérenniser des équipes de support. Des profils aux compétences spécifiques s'associent alors pour répondre à l'ingénierie nécessaire. Ceci a un coût, amorti par le coût qu'aurait une oeuvre de collaboration sans pipeline. Ce coût est souvent fonction de la taille de la structure.

On ensuite a vu plusieurs grandes familles d'outils. Ceux que les TD font le plus sont des solutions d'automatisation et confort pour le graphiste. Éventuellement des TD peuvent s'intéresser aux questions de production (collecte et traitement de l'information). Mais plus rares sont les

²⁹Un FTP est un protocole de communication destiné au partage des fichiers sur un réseau

³⁰Réseau virtuel privé, un collaborateur se connecte à notre réseau local comme s'il était présent à côté de nous

développements plus 'bas niveau' sur les questions d'asset manager que seuls les plus gros studios maîtrisent. En partie III, les exemples de programmes que j'ai pu mettre en place se concentrent sur les deux premiers niveaux.

Enfin on a pu voir que les pipelines sont adaptés aux besoins de la structure où ils s'inscrivent. Chaque pipeline est unique même si les grands concepts sont présents dans tous les studios. Des efforts de standardisation des méthodes sont faits pour simplifier les échanges. Le tout est imprévisible. Les TD s'adaptant au fur et à mesure de la production, comme des pompiers tentant de maîtriser un incendie.

Partie III. Expérimentations d'outils en production

Dans cette dernière partie, je partage quelques-unes des expériences et applications concrètes que j'ai pu mettre en place. Au cours des 8 dernières années, de mes stages à la cofondation du studio Montpelliérain Les Fées Spéciales, en passant du géant Illumination Mac Guff aux *petits* Autour de Minuit et In Efecto, j'ai eu l'occasion de créer toutes sortes d'outils pour faciliter la création d'oeuvres audiovisuelles.

Reprenant des exemples divers j'ai cherché à voir si je pouvais répondre aux différentes problématiques que pose le pipeline.

III.1. La nomenclature

S'il y a bien une chose que l'on doit mettre en place en premier lorsque l'on parle de pipeline c'est l'organisation des dossiers (l'arborescence) et fichiers selon leur nomenclature. Cette pratique s'opère en général naturellement quand on entame un travail de création collaboratif, souvent à minima, et parfois par la force des choses. L'équipe qui veut se structurer mettra en effet en place des règles communes. Et mieux elle structurera ces règles moins elle aura de surprises par la suite.

III.1.1. Une base essentielle à l'organisation et à la collaboration

Fabriquer un projet numérique génère très vite de nombreux fichiers : des documents de préproduction, comme des recherches graphiques, des *modeles-sheets*, etc. puis des assets : il faut modéliser, créer des textures, des rigs ... puis concevoir les plans : on procède au layout, on anime, on crée des effets spéciaux. La finalisation prend enfin le relais : on met en lumière, on sort des séquences d'images, par couche, puis on composite et on sort encore des séquences d'images "finales".

Tout le long, le processus génère des vidéos d'aperçu, des caches, des *logs*. Le tout est versionné, multipliant ainsi le nombre de fichiers.

Donc même un petit projet, comme une publicité de 30 secondes, peut générer des dizaines de milliers de fichiers. Et le travail en équipe exige alors de se mettre d'accord sur l'organisation.

Une équipe doit mettre en place un élément fondamental, à savoir une arborescence, c'est-à-dire une description de l'organisation des dossiers et fichiers, pour s'accorder sur une nomenclature commune des différents éléments.

Certains éditeurs proposent à travers leurs logiciels des arborescences-type lors de la création d'un projet. C'est par exemple le cas d'*Explore-TDI*, puis les *Maya-Project*. Mais il faut souvent aller au-delà de cette organisation, suffisante pour quelques plans, mais limitée, voire contraignante, sur de gros projets.

Il d'ailleurs intéressant de constater que ce sont des expériences d'organisations bancales, voire désastreuses, qui poussent les équipes à réfléchir à mieux s'organiser. C'est le cas chez les étudiants (d'ATI que j'ai pu auditionner pendant les projets intensifs, mais de toutes autre écoles aussi), mais aussi chez les professionnels.

J'ai fait le même constat à partir d'un projet mené avec les Fées Spéciales : sur *Antarctica*, pour Arte, nous avons démarré en séparant les typologies de plans (cartographie, incrustation d'effets spéciaux, ou intégralement en image de synthèse) et laissé les graphistes autonomes sur chaque partie. L'équipe technique étant trop occupée par le long-métrage *Dilili*, le projet a donc démarré sans grande préparation.

Contraints de remettre à plat l'organisation du projet dans sa phase finale, nous avons perdu du temps à ce moment crucial, au prix d'un grand stress et de nuits de travail. Des conditions aussi éprouvantes rappellent alors l'importance de l'anticipation et du temps pris pour l'organisation.

Il est donc primordial de mettre en place cette organisation et nomenclature dès le début, et trouver ensuite des solutions pour accompagner et encourager les graphistes à l'utiliser au mieux. Les TDs peuvent créer des outils-assistants, pour automatiquement créer l'arborescence. Par exemple dans les années 90, les graphistes de Mac Guff Ligne disposaient d'une commande `createproj`. Cet outil a évolué (MGA - Mac Guff Administrator) pour structurer des projets plus complexes, créer des arborescences de projets plus conséquentes et gérer leur répartition sur plusieurs disques durs.

Aujourd'hui les solutions de stockage professionnelles, comme les systèmes SAN ou NAS, évitent ces divisions, qui sont gérées de façon transparente par les IT.

III.1.2. Inadaptation d'une nomenclature à l'évolution d'un studio

En mars 2013, je suis recruté par In Efecto, un petit studio établi depuis près de 8 ans sur Montpellier. Jusque-là, le studio se structurait autour des arborescences-type proposées par le logiciel Maya. L'organisation consistait à créer une arborescence-maya par plan pour chaque projet. Après avoir réussi à fabriquer le court-métrage *Premier Automne* de Aude Danset, le studio envisage de poursuivre avec des projets plus importants en volume. Il me propose de reprendre en main la question de la nomenclature.

Une ébauche avait été réalisée avant mon arrivée, héritée d'un studio parisien grâce au brassage des intermittents. Cette nomenclature était conçue avant tout pour faciliter la navigation de l'utilisateur, car elle proposait de préfixer tous les dossiers de numéros pour les afficher selon un ordre de fabrication linéaire.

Ainsi, l'on avait : 01_layout, 02_anim, 03_vfx, 04_lighting, 05_compositing, ...

Cette organisation poussait les graphistes à connaître la cote de chaque niveau de l'arborescence par coeur. Elle laissait aussi peu de place à l'évolution imprévisible des dossiers : que se passerait-il

quand un département CFX (effets de simulations de vêtements) entrerait en jeu ? Faudrait-il décaler toute la numérotation, perturber tous ceux qui l'ont appris par coeur et mettre à jour les scripts qui l'utilisent ?

Mon expérience de terrain m'a démontré que l'arborescence évolue souvent d'un projet à l'autre et qu'elle peut même parfois évoluer en cours de projet. Une arborescence peut donc être complexe tant qu'elle répond aux cas nécessaires sur une production. Cela implique que l'arborescence et la nomenclature ne doivent pas être un problème laissé aux graphistes, mais bien gérées en amont et en parallèle de la production par des outils dédiés. Ce sont des tâches qui peuvent en effet être déléguées à la machine, mais à la condition que les graphistes aient des outils pour s'y retrouver d'une manière autonome.

À partir de ce constat, j'ai donc fait le choix de réécrire le cahier des charges pour une version très simplifiée de l'arborescence en reprenant celle pratiquée pendant plus de 3 ans chez *Mac Guff Ligne*. Ce choix était motivé par une connaissance précise de cette organisation modulable et évolutive.

III.1.3. Mutualisation de briques technologiques libres

Parallèlement à mes réflexions sur la nomenclature de In Efecto, j'ai pu retrouver un de mes anciens collaborateurs de chez Mac Guff Ligne, Damien Coureau (alias « Dee ») lors d'une visite au studio parisien Supamonks.

Dee avait entamé le développement d'un asset manager complet, Kabaret, basé sur ses réflexions et constats après plus de 10 ans de métier. C'est un outil qui n'est pas encore finalisé mais que Damien et le studio Supamonks souhaitent mettre à disposition sous licence libre. Néanmoins, suite à cette rencontre, il m'a confié une brique importante de son outil : le module dédié à la gestion de la nomenclature : `Kabaret.naming`. Un vrai cadeau !

Cette brique est une bibliothèque python³¹ qui requiert qu'on lui fournisse la nomenclature d'un projet via une description sous la forme d'un système de classes³². Le point fort de *Kabaret.naming* est qu'il peut accepter n'importe quelles structurations d'arborescences et de nomenclatures, la bibliothèque ne faisant que vérifier la conformité des chemins de fichiers existants ou à construire l'arborescence avec la description préalablement établie.

³¹Ensemble de fonctionnalités assemblées et réutilisables par différents programmes

³²« boîte à outils » qui permet de fabriquer un objet en programmation orienté objet

Après une première appropriation de cet outil, j'ai pu définir un modèle de nomenclature privilégié dont voici la structure :

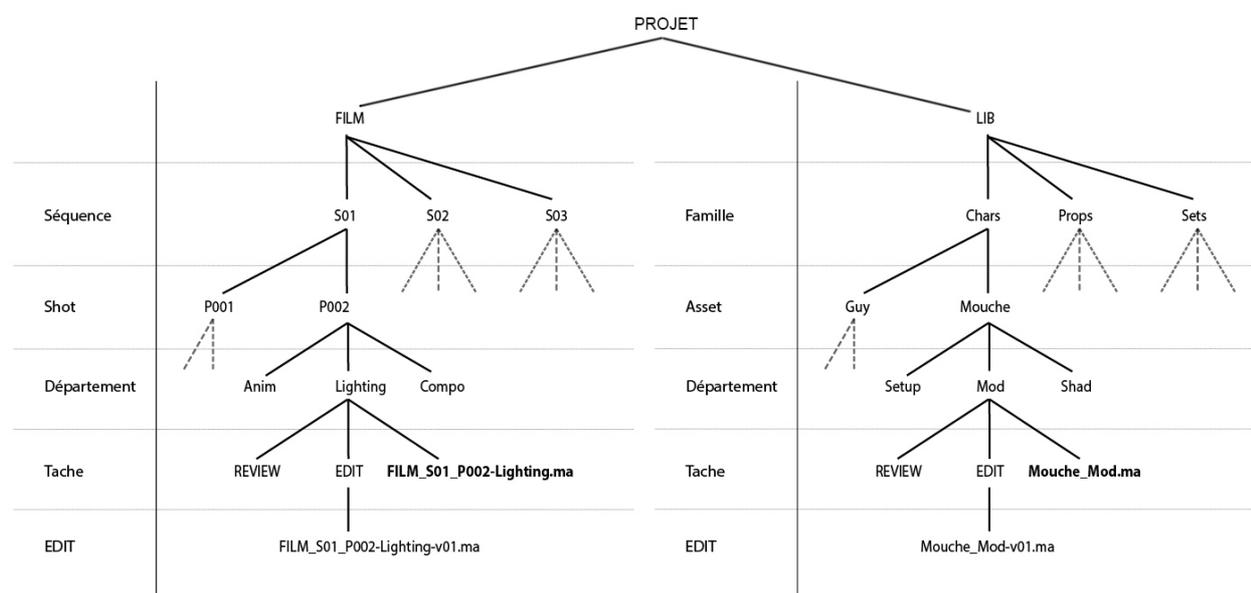


Illustration 7: arborescence simplifiée d'un projet avec son FILM, contenant plans et séquences, et sa LIB, la bibliothèque d'assets

Un dossier projet contient donc :

- Une (ou plusieurs) LIB (Library)³³ où l'on trouvera différents types d'assets (personnages, décors, accessoires), qui auront chacun une famille (principaux, secondaires, etc.), puis un nom. Ensuite l'on trouvera des départements (modeling, shading, rigging, ...) et enfin des fichiers de travail versionnés.
- Un FILM (ou plusieurs dans le cas d'épisodes d'une série) où l'on trouvera des séquences, qui contiennent des plans qui contiennent eux-mêmes des départements (Anim, Layout, Lighting ...) puis des fichiers de travail versionnés.

À tout élément de cette description il est possible de définir des règles : par exemple, dans un cas, on peut forcer certains choix, dans un autre cas, le nom d'un asset ne peut pas comporter d'espaces ou de caractères spéciaux, un plan commence toujours par un « P » et une séquence par un « S » suivis de chiffres, une version s'écrit toujours avec un « v » minuscule suivi de deux chiffres, on utilise des underscores « _ » pour séparer les blocs de noms, etc. Une fois cette déclaration faite, il n'y a plus qu'à s'en servir comme socle à tous les outils en contact avec l'arborescence.

Dès lors, on peut procéder de 2 manières :

Soit fournir un ensemble de paramètres (sous la forme d'un *dictionnaire* python) : « je veux fabriquer tel fichier, ce sera de l'animation pour le plan 10 de la séquence 3 » afin que le programme

³³Bibliothèque des assets utilisés dans un film

définisse lui même l'arborescence complète. Soit, au contraire, en fournissant au préalable un chemin au programme et en vérifiant que ce chemin est valable selon la nomenclature préétablie. Si ce n'est pas le cas, la librairie précise automatiquement quel élément de ce chemin est erroné.

Exemple de script python :

```
# vérifier qu'un chemin est valide
f = StoreFolder.from_path("T:/monProjet/PROJ/SEQ01/P001/Anim/PROJ_SEQ01_P001-
Anim.mb")
if f.is_wild():
    print("why ? : %s" % f.why())

# créer un chemin a partir d'un dictionnaire de description
config = {'Project': 'monProjet',
          'Film': 'PROJ',
          'Seq': 'S01',
          'Shot': 'P013',
          'Dept': 'Anim',
          'Extension': 'ma',}
anim_file = store(**config)
file_path = anim_file.path()
print(file_path)
>T:/monProjet/PROJ/S01/P013/Anim/PROJ_S01_P013-Anim.ma
```

À partir de cette brique, j'ai donc pu très rapidement créer des outils sur mesure de support pour les graphistes :

- De simples assistants sous forme de formulaires pour créer des arborescences complètes et structurées
- Un *Opener*, c'est-à-dire un assistant chargé d'ouvrir des fichiers sans se perdre dans l'explorateur de fichiers

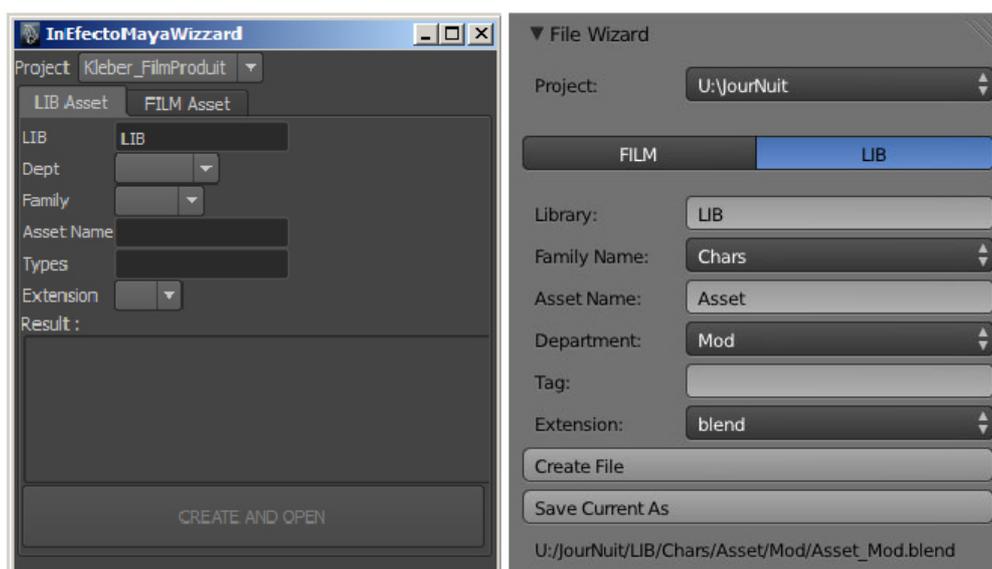


Illustration 8: Exemple des premiers assistants de création de fichiers utilisant la description d'arborescence Kabaret, dans Maya et Blender

III.1.4. Accélérer la transition vers une nouvelle nomenclature

Un changement de nomenclature est un événement sensible sur un projet comme nous avons pu le constater plus haut avec l'exemple d'In Efecto. Les graphistes doivent en effet prendre en main une toute nouvelle organisation lorsqu'il y a ce changement : c'est comme lorsqu'on arrive dans un supermarché que l'on connaît bien et qui a été complètement reagéncé. Il faut s'habituer aux nouveaux rayons, retrouver de nouveaux repères et les maintenir pour ne pas se tromper et trouver ce que l'on cherche aisément. En parallèle, tous les outils doivent eux aussi être mis à jour pour fonctionner avec cette nouvelle nomenclature. C'est une tâche de grande ampleur et ce particulièrement pour un studio qui a déjà un historique important de développements. Ce changement a donc été plus simple à gérer à l'échelle d'un petit studio comme In Efecto car il disposait de peu de structuration technique.

Dans ce cas, afin d'accélérer la transition, plusieurs solutions ont été mises en place :

- Créer des assistants, présentés dans la partie précédente (figure 8), afin d'aider les graphistes à utiliser la nouvelle arborescence de façon transparente.
- Créer des outils qui parcourent tous les fichiers d'un projet et qui en listent les fichiers mal nommés ou mal rangés.
- Faire en sorte que les fonctions utilisées par les graphistes ne fonctionnent que si le fichier de travail est bien nommé, et l'alertent en cas d'erreur. Le graphiste étant forcé de travailler sur des fichiers bien rangés.

Pour rebondir sur cette dernière solution, il est important de noter que l'objectif par rapport au graphiste est de bien repérer et au plus tôt les problèmes de nomenclature pour lui éviter de travailler sur des fichiers mal nommés. Ceci est d'autant plus important dans une arborescence de projets en images de synthèse car il y existe de très nombreuses dépendances d'un fichier vers un autre. Par exemple, une scène d'animation dépend de fichiers « personnages » qui dépendent eux-mêmes de textures. Si l'on nomme mal ces textures et qu'on les renomme par la suite, beaucoup de fichiers peuvent encore chercher l'ancien nom et ne pas retrouver ces textures en question.

Néanmoins il est à relever que les graphistes se seraient plaints si l'organisation avait été trop stricte. C'est pourquoi des systèmes de tag sur le nom du fichier (sortes de préfixes) permettaient de créer des variations non prévues par la nomenclature préétablie. Cette flexibilité avait bien entendu été longuement discutée avec les responsables des différents départements du studio. Le défi est donc à la fois d'imposer à tout instant que le fichier soit bien à la bonne place tout en assistant les graphistes avec des outils adéquats pour leur permettre une appropriation rapide de cette nouvelle nomenclature.

`Kabaret.naming` à donc été ma première utilisation d'une brique libre d'une telle importance et provenant d'un studio tiers. C'est son interopérabilité, offerte par le langage Python³⁴, et sa structuration selon des librairies flexibles qui ont permis de mettre rapidement en place une structure inédite dans le studio In Efecto. En effet, cette brique a permis de déployer des outils dans divers logiciels tels que maya, puis des commandes shell et des outils web³⁵. Plus tard, avec Damien Picard (alors stagiaire TD), nous avons transposé ces outils à au logiciel libre `Blender`, ce qui a nécessité une conformation assez compliquée de la librairie depuis python 2 vers python 3. Il est à noter que l'avantage de travailler en licence libre sur des outils comme `Kabaret.naming` est de pouvoir faire bénéficier à ces outils à et ses créateurs originaux des évolutions que nous lui apportons nous-mêmes.

Pour revenir à la flexibilité de cet outil, son système de librairie permet donc à plusieurs programmes d'utiliser la même description d'arborescence les libérant de garder en mémoire les règles d'enregistrement des fichiers. Cette fonction est en effet externalisée par `kabaret.naming` qui guide les programmes dans cette tâche.

³⁴Python s'est imposé comme le langage des logiciels liés à l'image de synthèse

³⁵Ici aussi tout développement se fait en python avec des serveurs web python tel que Tornado

Ainsi, toute mise à jour de l'arborescence devient plus aisée. Si l'on veut par exemple créer un nouveau département pour la création d'un plan, par exemple le CFX, il suffit d'ouvrir la description de la nomenclature et d'ajouter le nouveau département dans les choix possibles. Les programmes se basant sur `kabaret.naming` répercutent ces changements automatiquement et n'ont donc pas besoin de mises à jour, les champs étant remplis dynamiquement.

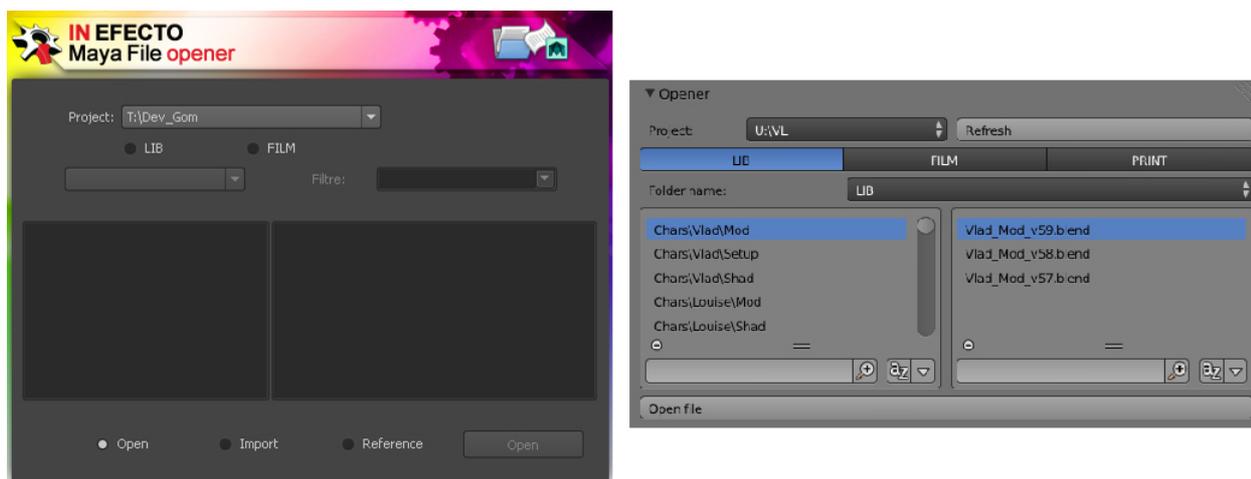


Illustration 9: Figure : Exemple des premiers assistants d'ouverture de fichier utilisant la description d'arborescence Kabaret, dans Maya et Blender

Depuis le lancement des *Fées Spéciales* l'utilisation de `kabaret.naming` n'a pas encore été adaptée. Mais nous allons continuer à collaborer, avec le studio Supamonks autour de cette série de solutions logicielles.

Car rappelons-le une fois encore, le logiciel libre a une carte importante à jouer, en permettant à plusieurs entités de mutualiser des développements pour créer des programmes encore plus stables et efficaces, capables de prendre en charge plus de cas différents d'utilisation.

III.2. Des outils d'automatisation et de confort

Suite à l'étude du pipeline et de ses différents enjeux et composants présentés précédemment, je vais à présent vous exposer différentes réalisations concrètes que j'ai pu développer au cours de ma formation ou dans le milieu professionnel.

Dans ce chapitre, je vais donc vous présenter 3 scripts que j'ai mis en place dans une perspective d'automatisation et de confort, et donc que je considère comme "accessoires" pour l'organisation et le pipeline.

Ces outils sont créés dans le cadre de productions où les budgets sont souvent limités et pour lesquels automatiser et déléguer les tâches peu créatives permet alors aux graphistes, artistes et

techniciens, de se concentrer sur des tâches plus gratifiantes, utiles et créatives. Des tâches que la machine ne peut pas, de toute façon, effectuer.

III.2.1. Le cartouche de la Fée coquille.

Ma première expérience d'outil d'organisation remonte à mon stage de Master 1, en 2008. Je suis embauché en tant qu'assistant de production sur la série télé *La Fée Coquille* de Jean Christophe Roger. En poste chez *Les Armateurs*, je rejoins la fabrication de l'épisode test, repartit entre l'équipe de préproduction à Paris et l'équipe de fabrication chez 2 minutes à Angoulême. Nous devons donc échanger de nombreux fichiers, des dessins principalement, entre les deux studios. Dans ce processus il est très important de pouvoir savoir un certain nombre de métadonnées utiles : ce que représente le dessin (un personnage, un décor, un accessoire... et lequel), qui l'a fait, à quelle date, et quel est son statut (brouillon, recherche de couleur, design final et validé ?). C'est d'autant plus important qu'il s'agit d'échanges qui peuvent être de nature contractuelle et que des envois ultérieurs peuvent annuler et remplacer une version précédente. Pour assurer au mieux la communication, le document échangé est généralement placé dans un cartouche : c'est-à-dire une page dans lequel un cadre regroupe le nom de la production et les copyrights qui s'appliquent, mais surtout les métadonnées citées précédemment (illustration 10).

Généralement la personne en charge de figer un état d'avancement (le graphiste, l'assistant-réalisateur, ...) utilise un fichier Photoshop. C'est un modèle (*template*) préorganisé et prérempli, dans lequel on vient placer l'élément graphique avant de changer les informations, et que l'on enregistre dans un dossier de suivi pour une communication ultérieure. C'est une procédure lourde et fastidieuse. Et lorsque cette tâche est répétée des dizaines de fois par jour, c'est source d'erreurs, d'oublis, et d'un précieux temps perdu.

Dans ce contexte, et en profitant des connaissances acquises en M1 j'ai pu créer mon premier script de production. Il s'agissait d'automatiser l'action décrite ci-dessus en glissant le fichier image à cartoucher sur un exécutable qui ouvrait un petit formulaire (python avec TkInter³⁶). Ce formulaire proposait alors des options simples et préremplies si le fichier était bien nommé, l'exécutable enregistrerait un fichier cartouche, en png ou pdf, et ce au bon endroit.

³⁶Python est un langage de programmation. TkInter est une librairie permettant de créer de petites interfaces, comme un formulaire.

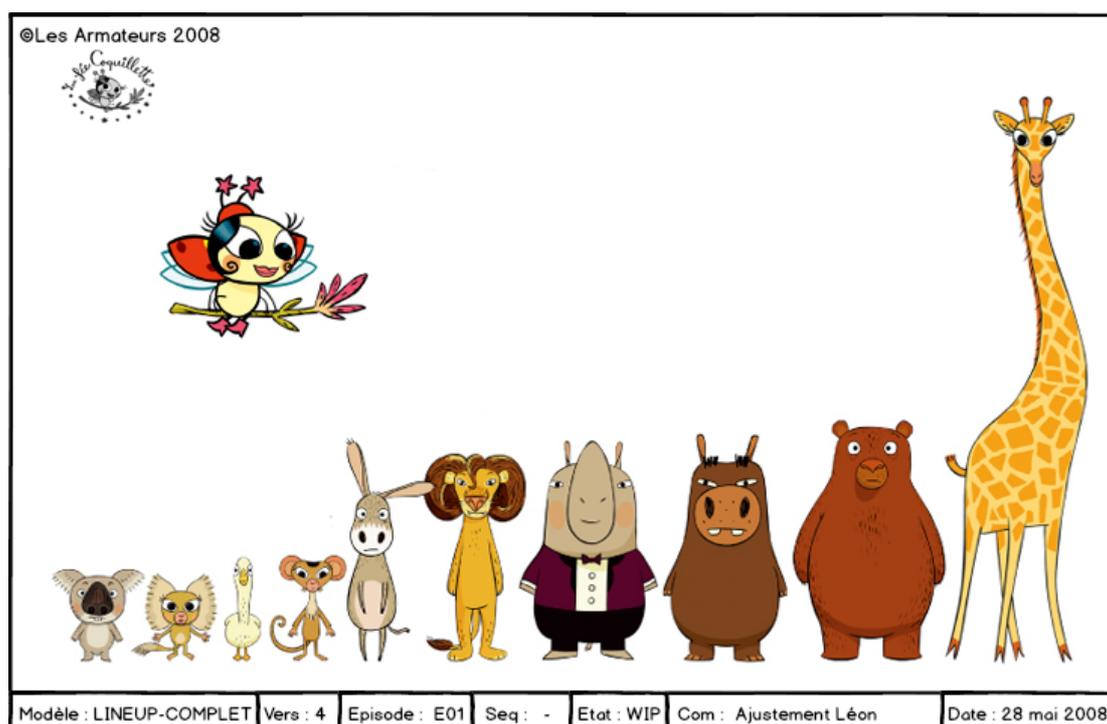


Illustration 10: Maquette de cartouche qui peut accueillir toutes sortes de visuels et qui indique quelques métadonnées importantes.

Ce premier exécutable que j'ai pu mettre en place a donc permis d'économiser la pénibilité d'une création manuelle de cartouche et a rendu les données plus sûres, donnant du confort, de la simplicité et du temps à l'artiste.

III.2.2. Un outil d'export Photoshop, un gain de temps considérable

Pendant la production du long métrage *Dilili à Paris*, j'ai été mandaté par Les Fées Spéciales auprès de l'équipe décors qui travaillait pour une autre société que nous assistions. Les décors de ce film étaient fabriqués sur Photoshop, sous la forme de matte-painting illustratifs, avec de nombreuses sources d'images et de peintures numériques. L'équipe décors venant du monde du dessin animé dit "2D", c'était en effet un parti-pris fort d'avoir une approche traditionnelle de ce travail par des décors pensés en 2D avant tout. Peu de place était donc offerte aux optimisations offertes par la 3D (UV, multiples textures, banques de matériaux, ...).

Par conséquent ces fichiers pouvaient être lourds. Par exemple lors de la création de décors pour des travellings, il n'a pas été rare de trouver des fichiers Photoshop de plusieurs dizaines de milliers de pixels de large. Le poids des ces fichiers de travail était alors énorme et toute manipulation ou action dans ce fichier très lente.

Partant de ces fichiers, les artistes doivent ensuite exporter ce travail en couches exploitables par nos scènes 3D. Ceci permet de vérifier le décor en situation en exportant les données utiles. Une série d'actions fastidieuse est alors nécessaire.

Ces fichiers Photoshop peuvent être composés de plusieurs couches, ou « plans » : des avant-plans (OL : *overlay*), des arrière-plans (UL : *underlay*), un sol, un ciel, etc. Selon les besoins d'actions (des personnages passent devant ou derrière) ou selon les effets de caméras (effet multiplan par exemple) il y aura plus ou moins de couches (figure 12).

Les graphistes travaillent dans un fichier versionné, mais doivent préparer plusieurs éléments lors de la procédure d'export :

- un aperçu *jpg* (au quart de résolution) avec le numéro de version (*Decors_Vxx.jpg*) et en gardant des informations de mise en scène (représentation des personnages ou actions)
- un fichier Photoshop (*psd*) « fusionné » dans lequel tous les calques (parfois des centaines) sont fusionnés (figure 13) afin de ne garder que les couches nécessaires. Ce fichier garde le numéro de version (*Decors_FSN_vxx.psd*) et l'on doit y avoir supprimé tous les repères de fabrications ou les informations de mise en scène qui étaient des références qui ne doivent plus apparaître à l'image finale.
- une image *png* pour chaque couche de ce fichier *psd*, portant le nom du décor et le nom de la couche (*NomDuDecor-Couche.png*), brute et prête à être utilisée dans les scènes de rendu 3D. Ces *png* ne sont pas versionnés : puisqu'on en a gardé le *psd* fusionné source, on écrase la version précédente dans un dossier spécifique. Ce dossier contient toutes les couches directement appelées par le logiciel 3D Blender. Ce logiciel se sert des *png* comme de textures pour des plaques 3D qui représentent le décor (illustration 11). Les décorateurs peuvent donc voir le résultat directement dans un espace 3D et/ou générer un *playblast* pour vérifier le résultat.

Afin d'évaluer les optimisations que l'on va pouvoir appliquer sur une procédure comme celle décrite ci-dessus, on peut se baser sur une unité simple comme le nombre de clics de souris pour mener à bien une tâche. On peut ainsi chiffrer la quantité d'actions et obtenir un score de pénibilité. Ce score de référence et un minutage du temps passé à effectuer une action serviront ensuite à comparer l'optimisation.

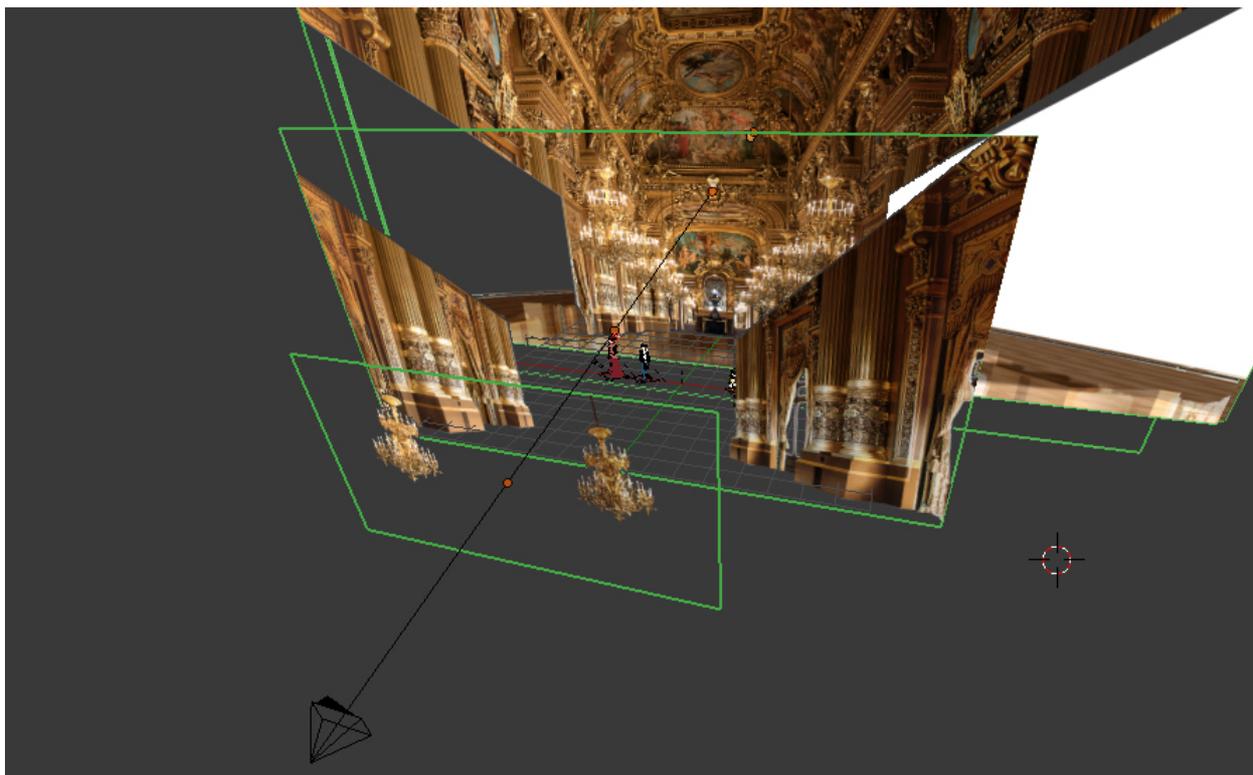


Illustration 11: Exemple de couches utilisées dans un espace 3D. Chaque plaque visible ici avec un contour vert correspond à un .png ©Nord-Ouest-Studio O

Prenons un cas simple avec 4 couches (cf. figure 12) : A_Exterieur, B_Sol, C_Arriere-plan, D_Lustres. En plus de ces 4 groupes contenant des dizaines de calques et calques d'effets, nous trouvons d'autres groupes : PERSONNAGES (Aperçu des personnages du Layout ou story-boards pour connaître les rapports de taille et cadrage), NOTE (prises de notes diverses), CADRE (le ou les cadres prévus dans la mise en scène), MARGES (le passe-partout noir autour de l'image qui marque la zone de sécurité), PERSPECTIVES (des traits d'assistance pour la construction des perspectives), DIVERS.

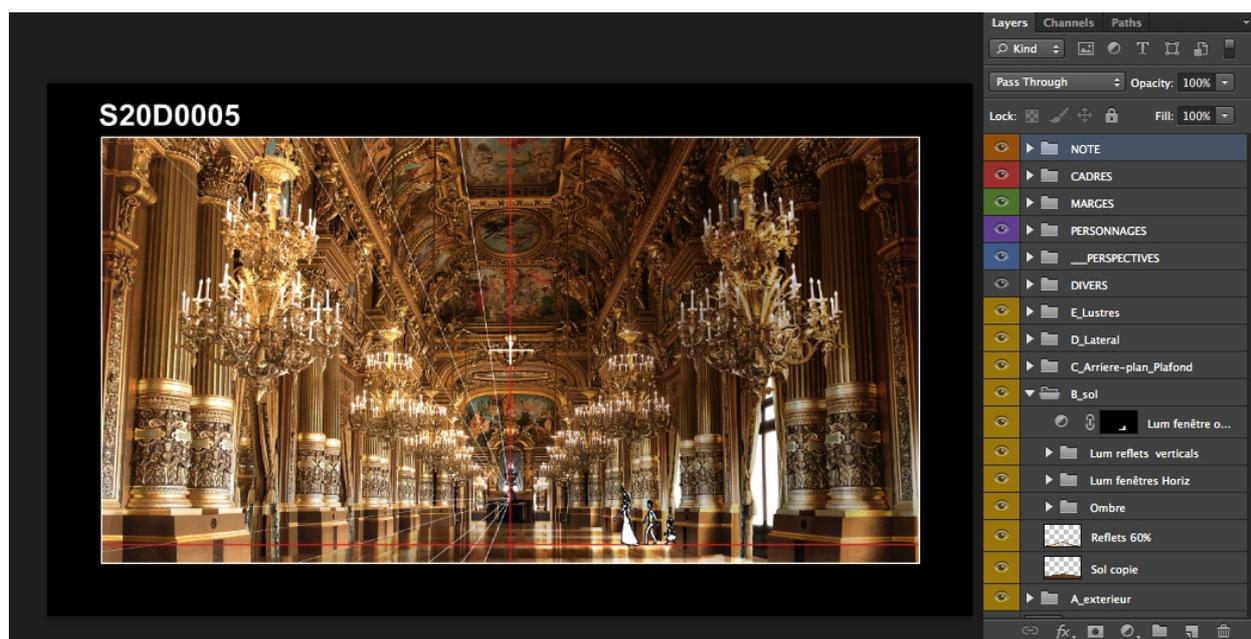


Illustration 12: Exemple de décor avant export. Chaque dossier peut comporter beaucoup de layers ©Nord-Ouest-Studio O

Voici la liste des actions et le nombre de clics correspondants pour réaliser un export complet du décor :

- Détruire les groupes NOTE, CADRE, PERSPECTIVES, : 5 clics
- Réduire l'image à 25% : 4 clics et un champ à remplir
- Exporter un jpg d'aperçu, trouver le dossier d'enregistrement, bien le nommer : 7-20 clics et champs à remplir correctement
- Revenir à la taille d'image initiale : 2 clics
- Sélectionner le groupe PERSONNAGES et MARGES et les détruire : 3 clics
- Écraser les 4 couches restantes en un layer (Merge) : 12 clics
- Renommer les layers (C_Sol-murs-fenêtres devient C) : 8 clics plus le nom à rentrer
- Sauvegarder ce fichier PSD fusionné au bon endroit : 7-20 clics et champ à remplir correctement
- Masquer 3 couches sauf la première : 3 clics
- Exporter le premier png, trouver son dossier et bien le nommer : 7-20 clics et champ à remplir correctement
- Masquer ce layer, afficher le suivant : 2 clics
- Recommencer l'export du png : 7-20 clics et champ à remplir correctement
- Recommencer ces deux étapes encore 2 fois pour les deux dernières couches : 16-45 clics et champs à remplir

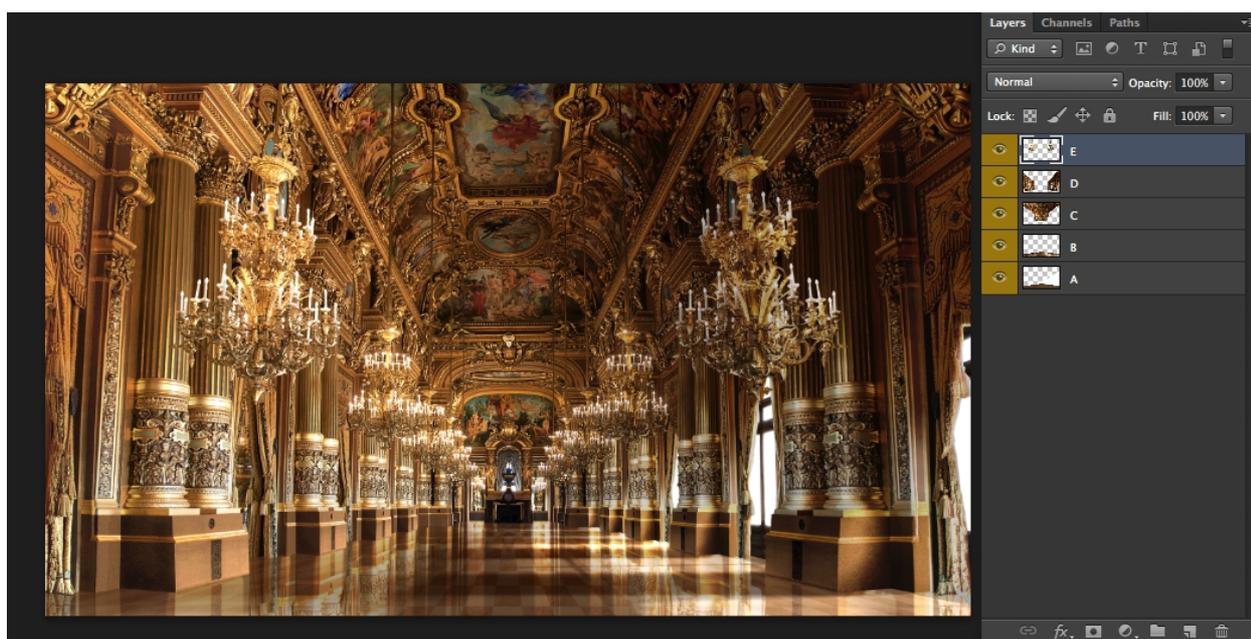


Illustration 13: Le même décor prêt à être exporté dans le fichier psd fusionné ©Nord-Ouest-Studio O

Ainsi, un simple décor constitué de 4 couches nécessite une série complexe d'actions qui prennent de 5 à 15 minutes de manipulations et d'attente (les grosses résolutions rendant de plus chaque export et sauvegarde très longs). Ces longues minutes correspondent à une fourchette allant de 80 à 160 clics et à 6 noms de fichiers à bien définir, et ce dans le bon dossier. À cela il faut ajouter que les décors peuvent contenir jusqu'à une douzaine de couches (le record est de 96 couches pour un plan particulièrement complexe). Autant d'énergie et de temps de travail perdu pour un graphiste en somme, et cela à chaque fois qu'il veut vérifier la mise à jour d'un décor, potentiellement une dizaine de fois par jour.

Tout ceci représente une perte de temps qui est estimée entre 45 min et 1h30 par graphiste et par jour. Pour une équipe de sept personnes, comme sur ce projet, c'est une perte de l'équivalent d'une journée de travail qui s'accumule chaque jour. Soit l'équivalent d'un temps plein perdu chaque mois. A partir de ce constat, j'ai donc proposé de consacrer du temps à créer un script qui, après plusieurs itérations, a pu réduire l'ensemble des actions citées ci-dessus à seulement deux clics. Les exports et créations de l'ensemble des fichiers nécessaires se faisant dès lors sans aucune intervention humaine. L'équipe a ainsi pu se concentrer sur des tâches plus créatives, et se permettre de faire des exports plus fréquents pour avoir un retour plus immédiat sur l'avancée de son travail. C'est grâce à la mise en place de ce type de solutions que les processus de création deviennent plus fluides et que l'on retire une masse considérable d'actions chronophages et aliénantes à l'équipe concernée.

En plus de faire gagner du temps au graphiste, ces scripts allègent les tensions sur toute l'organisation en réduisant considérablement les risques d'erreurs dans les nomenclatures. Les

structures d'organisation sont plus strictement respectées et rendent la recherche ou la communication plus simple.

III.2.3. Storymatic : au-delà du confort, une source d'informations

L'efficacité de l'automatisation des tâches démontrée ci-dessus peut également concerner de nombreuses autres applications comme la prise de relais sur des procédures d'échanges, de maintenance, de sécurité, etc.

Nous pouvons même aller encore plus loin et obtenir grâce à l'automatisation des informations cruciales pour la suite de la production. C'est le cas par exemple avec l'outil `storymatic` qui a fait ses preuves pour *Dilili a Paris*, puis pour le projet de série télé *Bonjour le Monde* (9*10 minutes, production en cours). En effet le directeur artistique Eric Serre qui était en charge de l'assemblage du story-board a été confronté au processus pénible d'assemblage des images et d'ajouts d'informations utiles comme le numéro de plan, la description de l'action, les dialogues, les informations de montage comme les cut ou les fondus... Et certains plans demandaient plusieurs vignettes pour décrire une même action. Or, il faut pouvoir mettre à jour rapidement ces story-boards, et ce sans que l'ajout d'une nouvelle vignette ne décale les pages existantes.

Eric Serre m'a demandé de chercher une solution qui lui permettrait d'avoir d'une part toutes les vignettes dans un dossier, et d'autre part les informations utiles à disposition (plans, dialogues, montage, etc.) afin de générer rapidement un pdf à jour, daté et formaté selon une grille précise. À partir de cette requête, j'ai donc établi un fichier texte avec un formatage particulier assez facile à assimiler et qui puisse être mis à jour par Eric : ajout d'une vignette, changement d'un dialogue, ou d'une action, puis lancement d'un exécutable qui génère un pdf à jour.

Les vignettes sont amassées dans un dossier (figure 14) et un script initie alors le fichier texte.

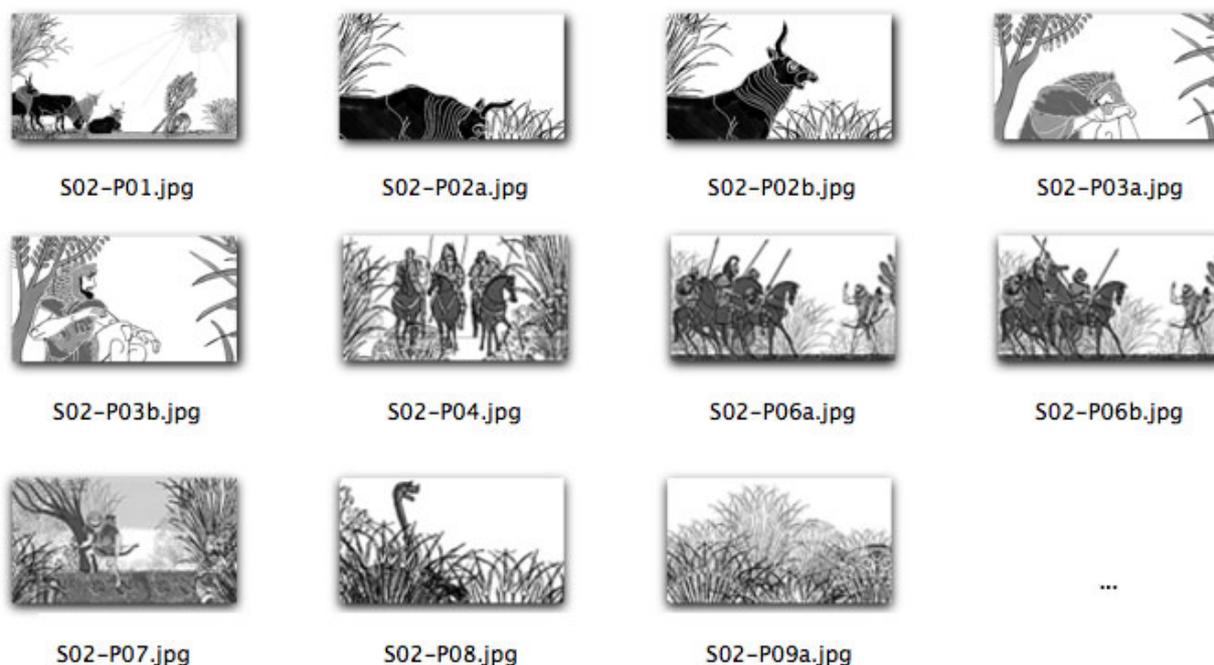


Illustration 14: Les vignettes sont assemblées dans un dossier ©Eric Serre

On peut maintenant ajouter les informations utiles ressemblent à la figure 15.

```

1 S02 L'ATTAQUE
2
3 ##### Antisèche #####
4 # nom_du_jpg
5 # ou :
6 # nom_du_jpg: texte de remplacement
7 # nom_du_jpg/ # un plan avec un cut
8 # et en dessous on peut ajouter :
9 # action: une action
10 # dialogue: un dialogue
11 # decors: un decors (en bas a gauche)
12 # orientation: une orientation
13 # echelle:
14 # cut
15 # fo # fondu d'ouverture
16 # fe # fondu enchainé
17 # ff # fondu fermeture
18 # # un commentaire - pas pris en compte
19 #####
20
21 S02-P01/
22 FO
23 action: dort, à l'ombre d'un olivier
24 orientation:
25 echelle: plan large
26
27 S02-P02a:P02
28 echelle: plan moyen
29 S02-P02b/:
30 action: sursaut d'inquiétude du boeuf
31
32 S02-P03a:P03
33 action: sort de son sommeil, avec surprise
34
35 S02-P03b/:
36
37 S02-P04/
38 action: cavaliers fixes et menaçants
39 echelle: plan large
40
41 S02-P06a:P06
42 dialogue: καλημέρα ! (Kalimera)
43 action: signe amical
44 S02-P06b/:
45 action: les cavaliers font lance un signal
46
47 ...
48
49
50

```

Illustration 15: Le fichier texte formaté contenant les informations utiles

En glissant ce fichier texte sur l'exécutable `storymatic`, on génère automatiquement un pdf, avec toutes nos vignettes et métadonnées (figure 16)

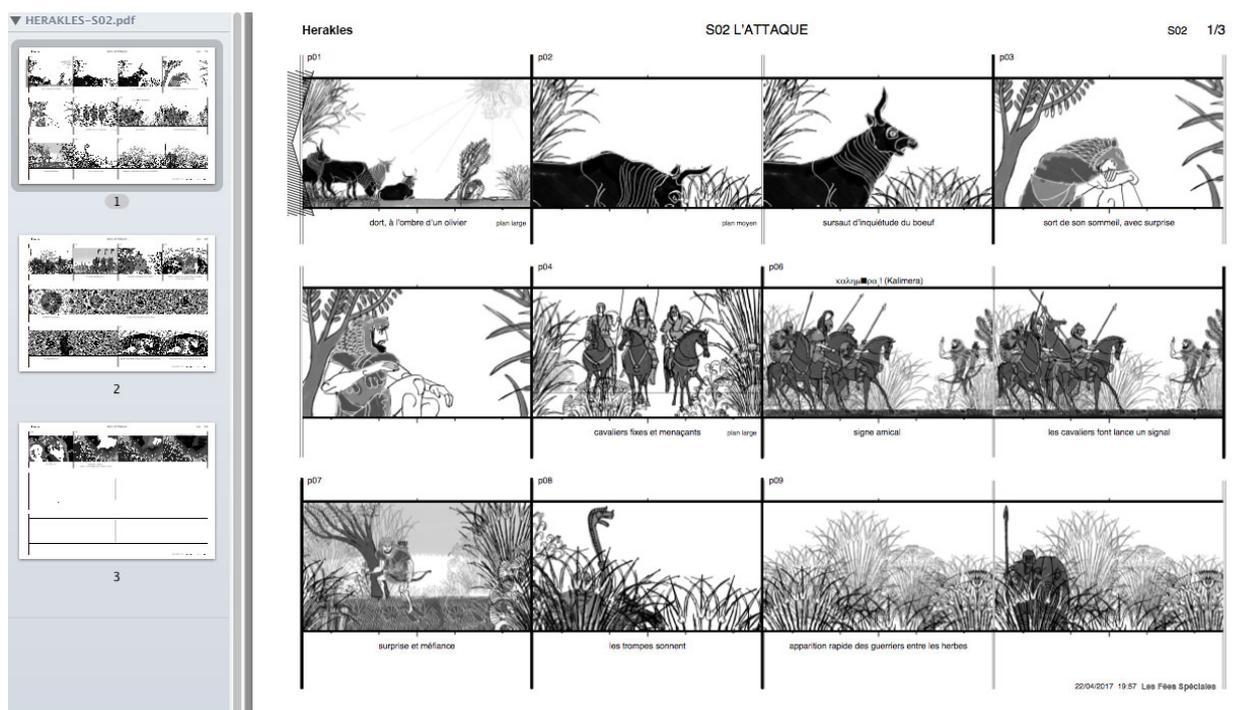


Illustration 16: Le pdf généré avec toutes les informations utiles ©Eric Serre

Avec cet outil simple, les fichiers sont bien rangés, la mise à jour est normée et rapide, et les informations utiles, y compris la date, sont toujours présentes dans le pdf sans que l'on n'ait à y penser.

Cette facilitation de la mise en place puis de la mise à jour du storyboard permet donc au graphiste de gagner un temps précieux, mais elle permet aussi de collecter une mine d'informations exploitables pour la suite de la production.

Prenons une fois encore exemple sur le long-métrage *Dilili* : les 32 séquences et 1290 plans qui composent sa version finale correspondent respectivement à 3025 vignettes de story-board. Grâce à la mise en place du fichier texte décrit ci-dessus j'ai pu récupérer toutes les métadonnées utiles au moment du dépouillement³⁷, soit : une vignette d'aperçu du plan, le nom et l'enchaînement des plans (à noter que l'ordre des plans ne correspond pas forcément à leur nom en cas d'inversement), les actions, les dialogues, des informations de montage (fondus et cuts) et d'échelles des décors, d'orientation de la caméra...

Cette récupération s'est faite automatiquement à partir des fichiers décrivant le story-board. Cela a permis de créer l'ensemble des plans du film, déjà associés aux bonnes séquences, dans notre outil de suivi de production interne *Criquet* (dont on parlera en section III.4.2). Nous avons ainsi réalisé

³⁷Lister l'ensemble des éléments que constituent le film : personnages, accessoires, décors et leurs présences dans les plans

en quelques secondes, ce que faisait auparavant une petite main qui listait fastidieusement, page par page, l'ensemble de ces informations dans un tableau ou logiciel dédié.

III.2.4. Les outils naissent de l'observation des méthodes de travail

Ces trois applications sont un échantillon des solutions que j'ai pu développer pour répondre à des situations problématiques rencontrées en production. J'ai d'ailleurs souvent été étonné par la résilience des artistes et techniciens face à des tâches d'une redondance extrême, et ce dans différentes tailles de structures.

Car en effet, quelle que soit la taille de l'équipe, petite ou grosse, l'on trouvera toujours des problématiques d'optimisation et de confort à résoudre. La différence entre une petite et une grosse structure est plutôt liée aux types de productions et aux types de problèmes rencontrés. De grosses structures auront bien évidemment des problématiques de nature différentes (grandes quantités de volumes, d'utilisateurs, de tailles de scènes, etc.).

Mais le problème récurrent est souvent le manque de connaissances des artistes de ce qui est faisable en termes d'outils. En découle une certaine difficulté pour ces derniers à formaliser une demande de solution aux équipes de support. Beaucoup d'outils sont ainsi conçus lorsqu'un TD se confronte lui-même au problème ou alors lorsqu'il a l'occasion d'être observateur auprès d'un graphiste pour découvrir les redondances de ses tâches. D'où la nécessité pour les TD d'avoir l'occasion de mener des audits auprès des graphistes et de créer des méthodes de discussion privilégiant les échanges.

On peut se poser la question de mettre en place des outils automatiques pour soulever des questions et repérer des noeuds. Par exemple les sociétés TeamTo et Mercenaries Engineering, ont développé un outil d'animation dans ce sens. Lors des Rencontres Animation Formation à Angoulême en novembre 2016 ils expliquaient être partis d'un comptage de nombres de clics associés à un certain nombre de tâches d'animation, de la même manière que nous l'avons fait sur le long métrage *Dilili*. Constatant que quasiment 30% de ces clics ne servaient qu'à effectuer une sélection préalable à une action, leur postulat était de proposer un logiciel qui économise, grâce à une assistance visuelle, cette sélection.

En conclusion, l'observation, assistée ou non, et la discussion, permettent de repérer des points de tension et de proposer des solutions.

III.3. Des outils d'assistance à la création

Ce chapitre aborde des outils qui ont un impact direct, ou en tout cas jouent des rôles de support importants, lors des phases créatives. En effet, dans les questions qui entourent la production et le pipeline il n'est pas rare d'avoir à créer ce genre de solutions. Nous pourrions presque les ranger dans les outils d'automatisation et de confort, mais je me permets de les séparer à cause de leur particularité et de l'impact direct qu'ils peuvent avoir à l'image.

III.3.1. Outil de préparation au story-board et layout

En juin 2015 je suis invité au festival d'Annecy pour parler de Criquet (cf. Partie III.4.2) lors de la conférence *Outils collaboratif et méthodes agiles*³⁸. C'est aussi l'occasion de montrer d'autres outils ou méthodes de travail et je présente la solution utilisée pour assister le travail de story-board d'une séquence test sur le long métrage *Dilili à Paris*.

L'idée est simple : permettre au storyboardeur d'utiliser une scène 3D comme base de travail. Ceci permet de prendre en compte des réalités d'espace et de cadrage. Pour l'occasion j'ai pris l'exemple d'une scène de café dont j'ai modélisé l'intérieur dans un style Art Nouveau d'une façon très sommaire à partir de photos de références.

À partir de là, l'assistant-réalisateur et moi même, créons une caméra, que nous positionnons librement. Le cadrage proposé est donc réaliste et prend en considération l'espace 3D. Nous anticipons donc des questions habituelles de compositions et de perspectives de l'image posées dans le futur espace 3D lors de la production et qui est souvent source d'erreurs dans la phase de storyboarding.

On va alors rendre séparément chaque objet vu par la caméra. On assemble ensuite ces rendus dans un fichier Photoshop avec les calques dans le bon ordre. Le story boardeur peut maintenant facilement dessiner l'action du plan, déplacer du mobilier si nécessaires, et poser des indications de lumière, directement dans l'un des outils de dessin qui lui est familier (Photoshop, Krita, ...). Avec cette procédure nous avons pu *boarder* les 70 plans de cette séquence du café Art Nouveau en seulement 2 ou 3 jours. Il n'aurait pas été possible d'obtenir cette précision en si peu de temps avec une autre façon de faire.

Après cette première version, j'imaginai une version temps réel, autonome et facile à utiliser pour le boardeur. Par pure coïncidence, dans la même conférence, juste après nous passait le grand studio

³⁸<https://www.annecy.org/a-propos/archives:fr/2015:fr/rencontres-2015/syntheses-des-conferences-2015/synthese:100000393394>

de *stop-motion* Laika. Venus parler du film *boxtroll* (2014) ils ont évoqué un outil : *Sandbox*. C'est aussi un outil d'assistance au story-board. Les décors sont modélisés tels qu'ils existent réellement sur les plateaux de tournage. Tout est entré dans le logiciel de 3D temps réel *Unity* grâce auquel le story-board artist peut se mouvoir librement dans la scène. Comme la *stop-motion* est un tournage ayant des contraintes physiques, car utilisant de vraies caméras, l'artiste est assisté par le logiciel pour prendre ces données en compte. Il peut par exemple choisir des focales parmi une sélection correspondant réellement aux objectifs à disposition du directeur de la photo. L'artiste peut aussi placer des silhouettes, à l'échelle exacte des marionnettes de tournage, dans son espace 3D pour s'en servir de référence. Une fois le cadrage validé il fait un rendu qu'il récupère dans son logiciel (Photoshop, Toonboon storyboard, ...) sur lequel il va pouvoir dessiner ses vignettes d'action. Les informations de placement et focale de caméra sont conservées pour pouvoir être facilement reproduites sur le tournage.

Laika propose donc un outil qui assiste les graphistes dans la création des actions qui prennent en compte les réalités du tournage, tout en récupérant de précieuses informations. De Laika aux Fées Spéciales, se sont donc différentes approches qui ont été présentées, mais qui répondent aux mêmes objectifs.

III.3.2. Du code au service de l'image, l'exemple des cartographies

Voici une catégorie de développement qui mériterait des sujets de recherche à part entière : ce sont les scripts et les programmes écrits pour concevoir directement une image. Ceux-ci sont créés par des artistes-codeurs et ils permettent la création d'images qui ne seraient pas possibles autrement. Toutes les promotions d'ATI ont vu ce genre de créations. On pense par exemple à l'art procédural ou aux créations de Michel Bret. On s'éloigne un peu du pipeline pur, mais la parenthèse est utile.

Dans un autre contexte que nous pouvons étudier, j'ai eu l'occasion de superviser les animations du projet *Antarctica* pour la chaîne de télévision Arte à l'automne 2016. Il s'agissait de créer 6 minutes de contenu. Une partie de ces six minutes avait une approche classique demandant le travail de graphistes, mais une autre partie nécessitait quant à elle une importante mise en place de programmation. En effet, nous devions pour ce projet manipuler des informations de différentes sortes pour créer de la visualisation scientifique. C'était notamment le cas de cartographies, comme une carte précise de l'Antarctique et des fonds marins de la région, ou encore de cartes montrant la fonte et le déplacement des glaciers.

Ce sont mes collaborateurs Damien Picard et Kevin Duy Nguyen qui ont codé différents outils pour mener ces tâches à bien. Damien Picard s'est concentré sur la récupération d'*open-data*³⁹ provenant de différents laboratoires et d'organismes comme la NASA pour recréer des modèles permettant de passer de cartes projetées de type mercator (la représentation habituelle que l'on a de la planète) à des projections centrées autour du Pôle Sud. Damien a ensuite développé l'application de données ultra-précises sur cette représentation pour détailler les profondeurs en y appliquant des matériaux qui ont donné des résultats esthétiques et justes.

Kevin quant à lui s'est focalisé sur une séquence montrant le déplacement des glaciers sur la surface du continent antarctique. Les jeux de données étaient des vecteurs de déplacement moyens sur l'année normalisés dans des fichiers .tif dont la résolution était d'un pixel pour 400m. À partir de ces vecteurs qui donnent une direction et une vitesse (de 0 à 500m/an), Kevin a utilisé le programme `processing` pour créer des systèmes de particules représentant les flux aux vitesses adéquates. Ces particules, ajoutées aux cartes de Damien ont donné des résultats lisibles et justes.

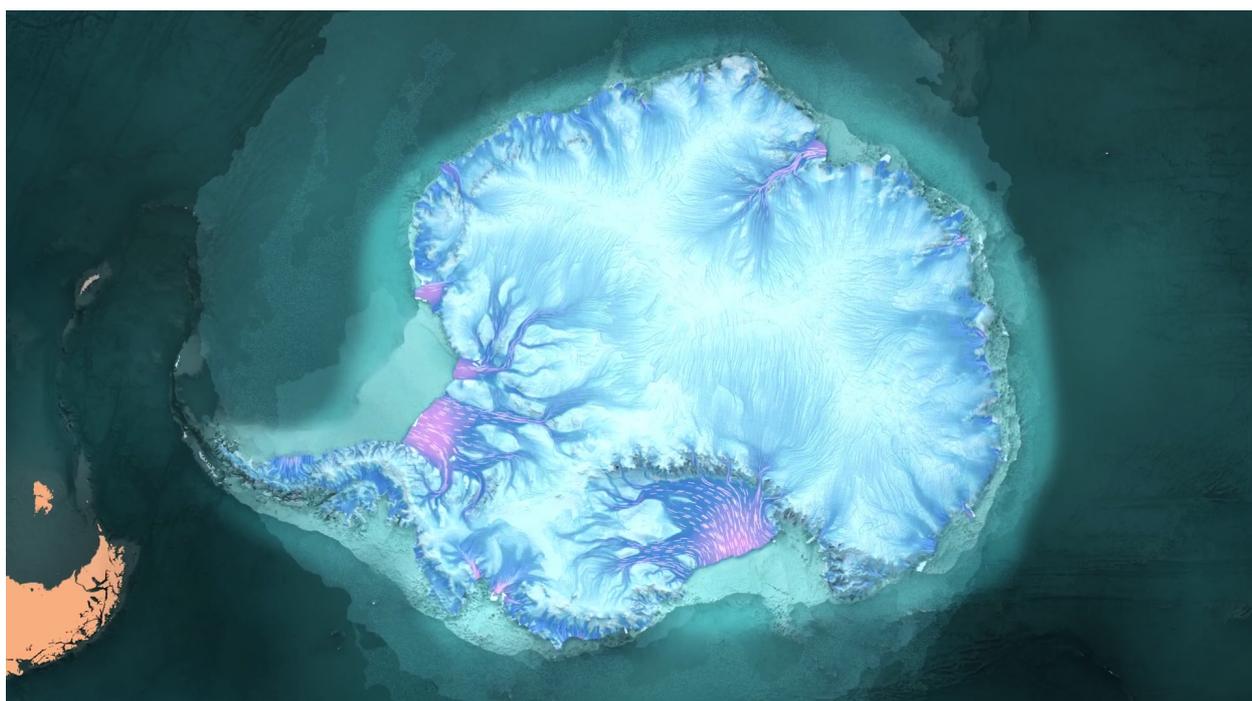


Illustration 17: Cette image, résultat du traitement de données ouvertes de sources différentes, est presque intégralement procédurale

À noter que j'ai pu présenter ce travail aux Rencontres Animation Développement Innovation 2016 par un diaporama dont les slides commentées sont disponibles sur le site de l'événement⁴⁰.

³⁹ Données numériques laissées en accès libres par une collectivité, un service public ou une entreprise, ici des données cartographiques précises

⁴⁰ http://www.rencontres-animation-formation.org/synthese_2016 Annexe 6

III.4. Les outils de suivi de production

On l'a vu en partie II., l'autre grand domaine de la création d'outils de supports est lié à la question de la gestion de production. Ce ne sont pas directement des outils de pipeline, mais leur rapprochement avec ce dernier est de plus en plus fréquent. Les outils de gestion de production centralisant en effet des bases de données dont le pipeline se sert.

Comme précisé plus tôt, je suis d'avis de garder les notions de *gestion de production* et de *suivi de fabrication* dans la définition large de pipeline.

Pour expliquer la liaison qui peut exister entre le pipeline et la gestion de production, je parlerai dans cette sous-partie de la mise en place d'un outil stratégique : la Feuille De Temps. Puis je parlerai d'un prototype devenu outil de production que j'ai développé pour le long métrage *Dilili à Paris*.

III.4.1. La Feuille De Temps

Revenons lors de la transition de Mac Guff Ligne à Illumination Mac Guff, plus gros studios français d'image de synthèse. À partir du *Lorax*, les équipes de Illumination Mac Guff ont en effet progressivement grossi pour atteindre aujourd'hui près de 900 personnes. Du jamais vu en France, ce qui a nécessité de repenser la question du suivi de temps.

Lorsque j'étais employé dans ce studio, Étienne Pêcheux, chef des TD, m'a donc chargé d'une réforme de la FDT (Feuille De Temps), pour aboutir à un outil plus moderne, mais surtout très rapide à remplir. Rappelons que la FDT est une question sensible à bien des égards : pour obtenir des artistes et techniciens qu'ils la remplissent correctement alors qu'ils ont des journées complexes et aléatoires, il faut leur proposer une solution très rapide.

Contrairement au système précédent où il fallait péniblement ajouter séparément toutes les tâches effectuées en les cherchant dans des listes et des sous-listes (ce qui pouvait déjà prendre 15 à 20 secondes), nous souhaitons que toutes les tâches associées au graphiste soient intégralement disponibles et éventuellement préremplies. C'est-à-dire que la page affichée soit unique pour chaque utilisateur et qu'elle lui offre un contenu et des suggestions adaptés à son profil. Que l'on soit animateur sur plusieurs séquences ou TD sur plusieurs projets, on ne doit donc pas avoir à chercher des éléments nouveaux, tout doit être sous nos yeux.

L'ensemble de ces tâches doit être bien classé (par projet, puis par séquence par exemple) pour les retrouver rapidement. Et pour rendre le remplissage plus pratique, nous avons donc proposé qu'une colonne regroupe les tâches les plus susceptibles d'intéresser le graphiste.

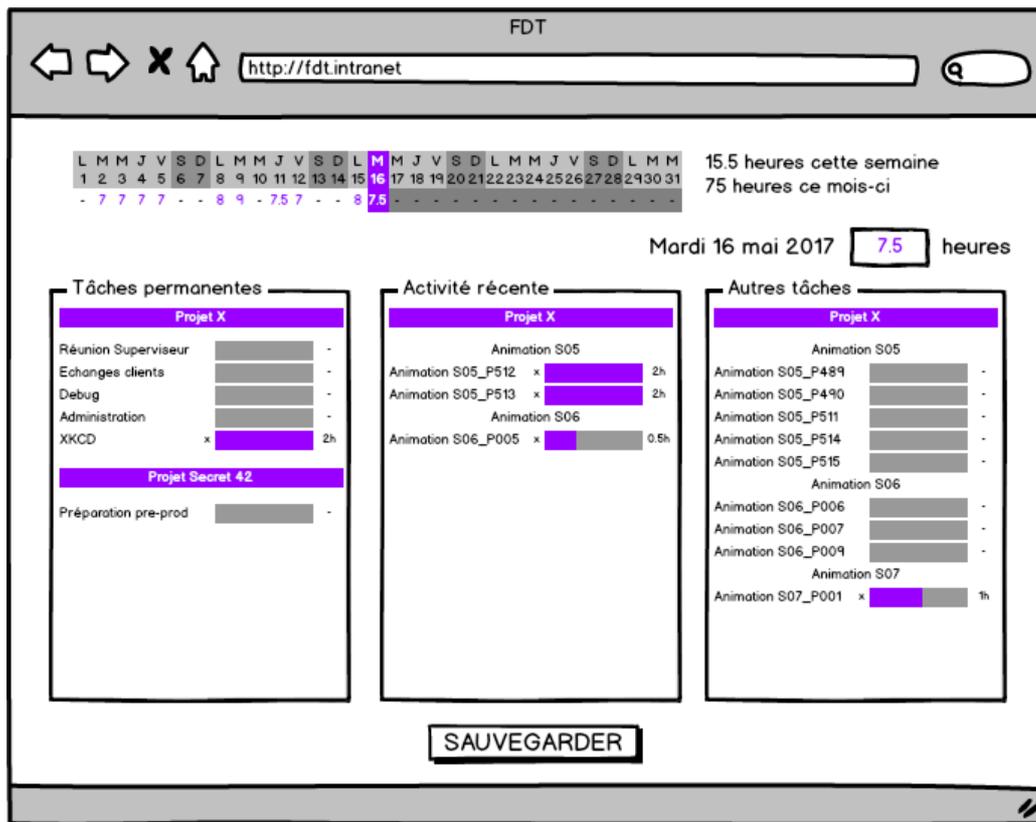


Illustration 18: Maquette de la FDT créée pour Illumination Mac Guff

L'idée de base est que le graphiste travaille et fait des sauvegardes dans InK, l'asset manager dont nous avons parlé précédemment, et ce tout au long de la journée. Il peut sauvegarder un fichier d'animation, ou une séquence d'images rendues au lighting par exemple. Chacun de ces « *publish* » (sauvegarde) pousse⁴¹ aussi vers Shotgun, l'outil de gestion de production, une série d'informations. Basé sur cet historique d'activités pour un jour donné, l'on peut alors mettre en avant dans la FDT un ensemble de tâches qui semblent s'approcher le plus de celles sur lesquelles le graphiste a travaillé.

Dans la FDT la colonne centrale affiche ces tâches. S'il a fait au moins une sauvegarde dans la journée de l'ensemble des fichiers qu'il a traités, il est fort à parier que le graphiste n'aura même pas à regarder dans la colonne de droite, celle de toutes les tâches qui lui sont assignées. À noter que la colonne de gauche représente les tâches permanentes ou récurrentes. Elles ne sont pas attachées à un plan ou à un asset en particulier. Pour un superviseur cela peut comprendre le temps de réunion inter-département, ou le temps en validation du travail de ses graphistes par exemple.

⁴¹En informatique le terme pousser, push, se dit pour des programmes qui envoient des informations vers un autre programme

L'autre fonctionnalité pour remplir rapidement cette feuille de temps est fondée sur un renversement du paradigme précédemment en place. Avant l'on indiquait le temps de travail par tâche pour obtenir une somme totale des heures travaillées dans la journée. Dans la nouvelle version, un temps de travail par défaut était déjà proposé : 8h (du lundi au jeudi, et 7h le vendredi, pour les contrats de 39h). On pouvait changer cette valeur pour indiquer qu'on avait moins travaillé ou qu'on avait fait des heures supplémentaires. Mais cette valeur n'était pas affectée ensuite par les tâches qu'on allait sélectionner.

Au contraire, on divisait ce temps parmi les tâches sélectionnées. On demandait en fait au graphiste d'indiquer s'il avait peu ou beaucoup travaillé sur les activités listées. Et la répartition du temps se faisait en fonction de ces choix. L'interaction se faisait via une petite barre de quantité à côté de la tâche, comme une jauge qu'on remplit plus ou moins.

Ainsi, même pour quelqu'un qui avait une dizaine de tâches différentes dans la journée, le remplissage était rapide. Tout était disponible d'un coup d'œil et le tri des activités était prémâché en fonction de l'activité du graphiste.

Néanmoins, malgré les efforts de simplification qui devaient permettre un suivi plus efficace, les dérives étaient courantes. Entre des chargés de production qui demandent des temps plus précis de travail, et des graphistes maniaques qui pensent que 1 millième à une valeur définie (15mn par exemple), nous avons passé beaucoup de temps à former les équipes et à leur expliquer la valeur et l'efficacité d'un décompte des temps par des jauges « imprécises ». Ceci démontre peut-être que des recherches d'ergonomie sont encore à faire ou bien que nous sommes formatés par certaines façons de faire.

De plus, beaucoup de graphistes ne remplissent pas la FDT au fur et à mesure. Il était commun de les voir la remplir le vendredi soir, voire le lundi quand le chargé de production se rendait compte que le temps n'avait pas été rempli. Cela diminuait la qualité de l'information, car se souvenir précisément du temps passé à faire quelque chose devient difficile le temps passant. Ce constat peut correspondre à une forme d'échec de la réflexion que nous avons déployée pour encourager un suivi plus ponctuel des équipes sur leur temps de travail...

Pour finir, ces valeurs sont validées par le chargé de production. Les chiffres sont automatiquement assemblés pour connaître la consommation du budget sur un plan ou un asset permettant ainsi de détecter une avance ou un retard. Et dans tous les cas, ces chiffres sont cruciaux pour la comptabilité afin qu'elle puisse effectuer la paye des salaires.

III.4.2. Créer un outil d'aide au devis

Fin 2014, studio O, le studio de Michel Ocelot, mandate Eric Serre et moi même pour fabriquer une animatique, en couleur et en 3D, d'une séquence du long métrage sur lequel Michel travaille : *Dilili à Paris*. Le projet est en train d'être signé avec Nord Ouest Films, et nous faisons ce test pour convaincre d'autres partenaires financiers (comme des chaînes de télévision), mais aussi afin de définir les grandes lignes de fabrication. Ce test nous a permis de valider des méthodes, notamment le mélange des photos du décor et de la 3D (pour les personnages ou certains mouvements de caméra).

Au printemps 2015, sur la base du story-board quasi complet, nous sommes donc chargés d'établir un dépouillement complet pour estimer les temps de fabrication afin d'établir un devis.

Dépouiller un projet consiste à décortiquer le story-board (ou a défaut le scénario) afin d'énumérer l'ensemble des éléments qui seront à fabriquer (telle quantité de décors ou de personnages par exemple). À l'échelle d'un long métrage, l'exercice est compliqué. Nous voulons par exemple lister le nombre d'utilisations d'un personnage pour savoir quelle méthode de fabrication appliquer pour être le plus efficace (un personnage peu vu peut être fabriqué en « pantin » plutôt qu'en 3D complète par exemple). De même, un décor très utilisé (recadrages ou réutilisation), prendra plus de temps à être fabriqué, car il est plus présent à l'écran.

Traditionnellement pour mener ce travail à bien l'on fait de grands tableaux, parfois très complexes, dans un tableur (Excel ou équivalent). Il faut imaginer croiser 1400 plans, avec plusieurs centaines de personnages et près de 1000 décors. Or, je ne souhaitais pas utiliser ces méthodes limitées et laborieuses qui rendent la tâche très complexe.

Il se trouve qu'à cette époque je travaille au cahier des charges d'un outil de gestion de production dénommé *Ricochet*, en me penchant plus particulièrement sur la question du choix des technologies de bases de données. Les bases de données SQL⁴² semblent trop limitées pour le cahier des charges établi et le choix du NoSQL⁴³ semble plus porteur. Je profite donc de l'occasion qu'offre la mission de dépouillement du story-board de *Dilili* pour développer un prototype de Ricochet qui nous permettra d'indiquer :

- tous les plans, en indiquant des informations importantes comme leurs durées, la présence d'animation 2D ou d'effets spéciaux, et d'autres métas-données utiles

⁴²Le SQL est un langage informatique normalisé pour manipuler des données dans des bases de données relationnelles

⁴³Par opposition aux bases de données SQL celles-ci se démarquent des paradigmes classiques des bases relationnelles

- tous les personnages, par famille, et leur présence dans les plans (ce qu'on appelle le casting)
- tous les décors, en fonction des réutilisations et recadrages estimés
- une partie des accessoires importants

Nous avons donc des informations précieuses sur le nombre d'utilisations de tel ou tel élément, qui, une fois croisées aux durées de leur fabrication, permettront de faire une estimation du coût en temps de projet.

On applique également des quotas en fonction du type d'élément à fabriquer et de sa difficulté et on utilise des bonus/malus en fonction de paramètres précis. Cette procédure est appliquée à chaque département de fabrication (Layout, Modélisation, Rig, Animation, Compositing, ...). Par exemple, on applique un quota de 2 secondes par jour en animation si le plan est moyennement difficile; si ce plan fait 4 secondes alors il y a 2 jours d'animation.

On va rapidement obtenir les masses de travail, par plan, par asset et par étapes de fabrication, et l'on va commencer à savoir combien de jours sont nécessaires en modélisation, en animation, en compositing, et ce séquence par séquence.

Nous y intégrons des méthodes de fabrication que nous préconisons, notamment sur les décors et les foules, afin de réduire le budget total et pour faire en sorte que le projet puisse se faire à son coût le plus optimal.

Puis on élabore des scénarios, selon différentes tailles d'équipes, pour estimer sur quelles durées le projet pourrait s'étaler. On obtient alors un budget et un planning prévisionnels.

Toutes ces informations sont remises à Nord Ouest Film et Studio O, qui ont maintenant une idée du coût global du projet. Ces informations les aideront ensuite à discuter avec le prestataire Mac Guff Ligne (qui affinera le devis pour ses spécificités) et à boucler le plan de financement qui permettra au film d'être fabriqué en France. Car l'objectif était alors que le film se fasse sans faire appel à une coproduction internationale avec le Canada⁴⁴, ce qui aurait conduit à une trop grande complexification de la fabrication du film.

L'exercice est un succès. Ce prototype nous a permis d'extraire l'ensemble des chiffres utiles. Et il nous a fait découvrir une technologie de bases de données NoSQL graph⁴⁵ prometteuse dans le cahier des charges : Neo4J.

Une interface web simple avait permis à Eric d'entrer toutes les informations utiles sans mon aide, pendant que je traitais les scénarios chiffrés.

⁴⁴Alors présenté comme seule solution avec ses crédits d'impôts et compétences pour mener à bien le projet

⁴⁵La base de données est un graph avec des noeuds qui contiennent des données et qu'on lie à d'autres noeuds avec des connexions

Le projet Dilili a donc bien démarré en février 2016 avec le scénario initial et la répartition suivante pour la fabrication : Les Fées Spéciales au layout (mise en scène) et aux foules, Nord Ouest Film aux décors et enfin Mac Guff Ligne (France et Belgique) à l'animation principale, au rendu et au compositing.

III.4.3. Criquet : du provisoire qui dure

Le développement de Ricochet prenant du retard, c'est sur la base de ce prototype utilisé pour le dépouillement de *Dilili* que j'ai développé *Criquet*, l'outil de suivi de fabrication que nous avons en production sur ce même projet.

Dépourvu de fonctionnalités de suivi de temps, il nous permet de suivre l'état de fabrication de tous les éléments et d'assister les graphistes. Par exemple, les informations de casting permettent aux layout-artist de créer des scènes préfabriquées avec les personnages déjà présents.

Les informations de timing sauvegardées dans *Criquet* sont aussi cruciales. Quand un playblast est fabriqué depuis une de nos scènes de layout ou d'animation, un script demande à *Criquet* quel est le timing prévu et le compare avec le timing de la scène. S'il y a une différence, si le graphiste a changé le timing, l'information est indiquée directement sur la vidéo, ce qui permet d'éviter qu'on l'oublie, et *Criquet* est averti du changement (figure 19).

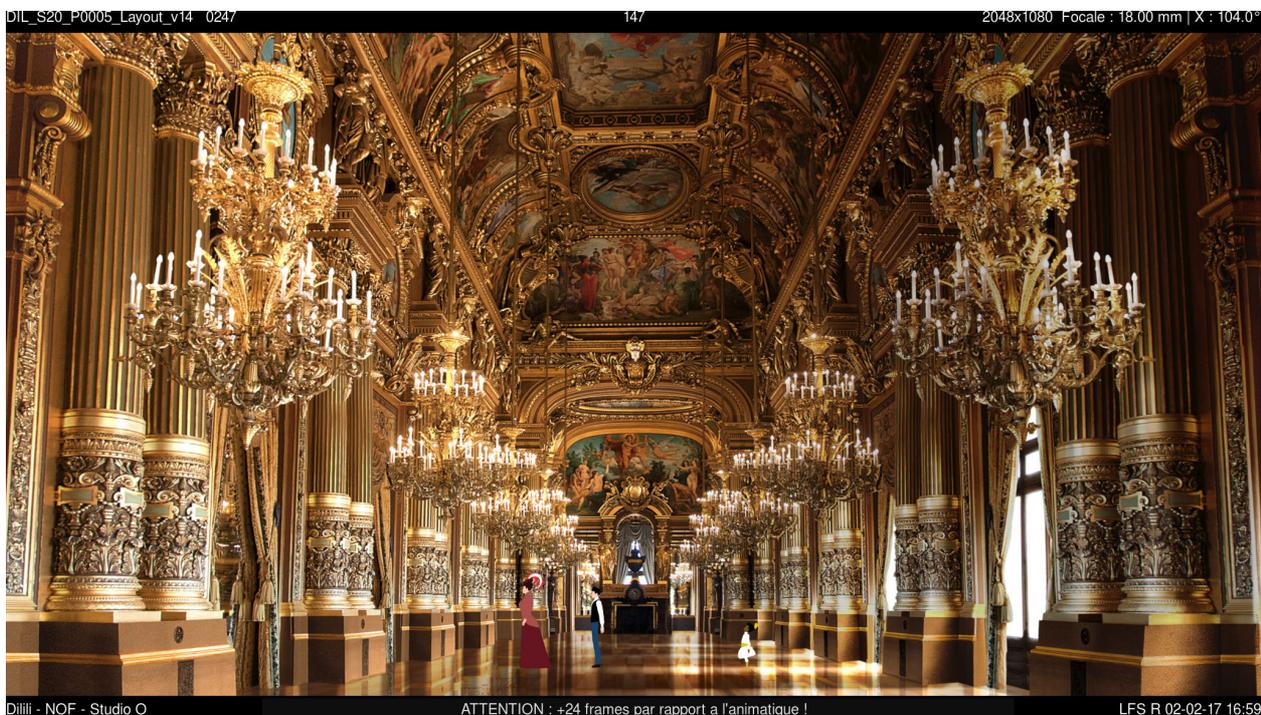


Illustration 19: Exemple de playblast standardisé où toutes sortes d'informations utiles sont écrites sur la vidéo : séquence-plan, version, timecode, résolution, focale, angle de la caméra, date, avertissement de timing, ... ©Nord-Ouest-Studio O

Criquet nous assiste également pour suivre les 500 pantins que nous avons fabriqués et permet de les retrouver facilement grâce à un trombinoscope. De là, un simple clic importe le pantin dans la scène d'animation (c'est un outil de confort).

Criquet a gagné beaucoup de fonctionnalités pendant la production (une API, un système de messagerie et d'historique sur les assets, des automatisations, ...) et a servi aux équipes des Fées Spéciales mais aussi de Nord Ouest pour suivre la fabrication des décors. Mac Guff Ligne étant quant à lui toujours sur Shotgun.

Même si le développement de la v1 de Ricochet est sur le point de commencer avec l'embauche d'un développeur dédié, c'est encore Criquet qui servira au prochain gros projet des Fées Spéciales. De nouvelles fonctionnalités sont en cours de développement pour faciliter la communication et la validation d'éléments avec notre nouveau client. Toutes ces recherches de fonctionnalités étant bien sûr un terrain d'expérimentations utiles pour la future première version de Ricochet.

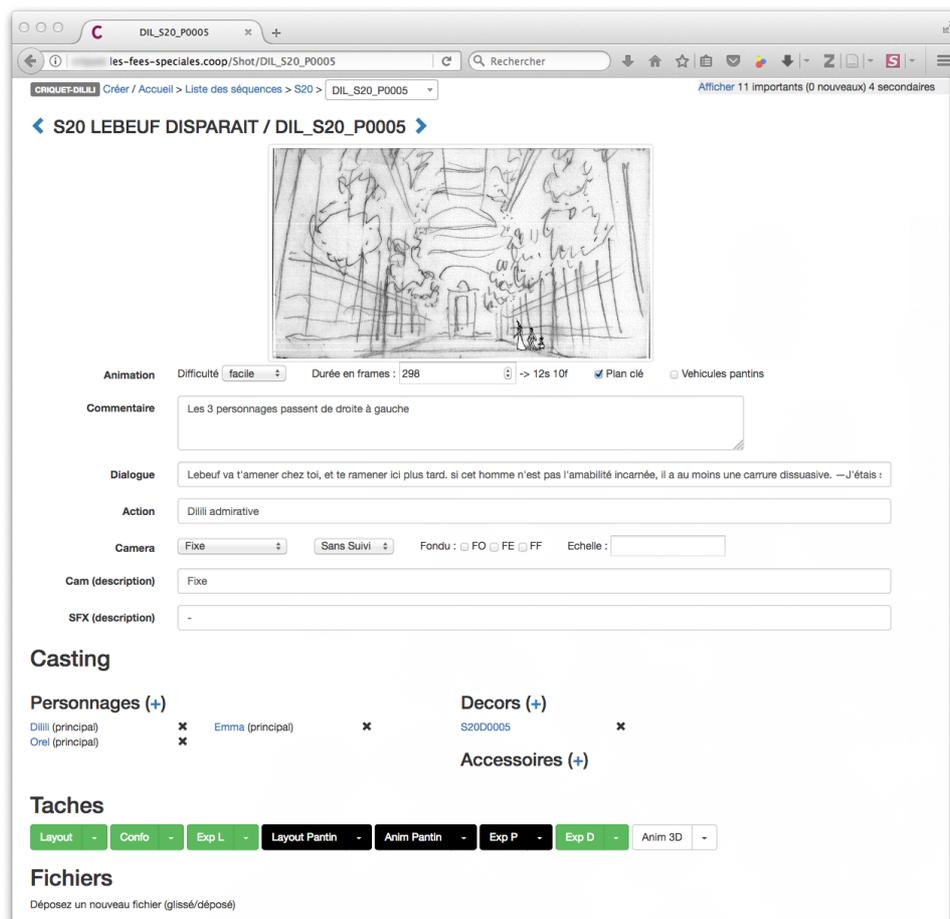


Illustration 20: Criquet, page typique d'informations d'un plan du film avec les métadonnées utiles, le statut des départements et le casting

Criquet, en tant qu'outil d'aide au dépouillement, avait été présenté au festival d'Annecy 2015 puis aux premières Rencontres Animation Développement Innovation à Angoulême la même année. Criquet n'est pas disponible en l'état, car il reste un prototype « *quick and dirty* »⁴⁶, entré en production par la force des choses. Néanmoins Ricochet a vocation à être distribué sous licence libre. Dans les deux cas, la base de données est et devrait rester Neo4J. Le backend⁴⁷ tourne aujourd'hui en du python et le serveur web Tornado. Le frontend⁴⁸ est une application web utilisant l'html5 et des frameworks javascripts.

⁴⁶Sa traduction la plus simple, "à la va-vite", illustre le travail souvent dans l'urgence et sans prendre trop de précaution des infographistes scripteurs. C'est la devise du département TD d'Illumination Mac Guff

⁴⁷En information le backend est la couche invisible d'un logiciel qui gère l'accès aux données

⁴⁸Le frontend est la couche visible, ou couche de présentation, d'un logiciel ou service et sert d'interface avec l'utilisateur

III.5. Outils fondamentaux : Une question de long terme

Les outils qui vont cimenter et structurer le pipeline sont à regarder du côté du *filesystem*⁴⁹.

Mettre en place de tels outils demande plus qu'un profil de TD. Ce sont des questions d'architectures de fichiers et de droits utilisateurs qui demandent des connaissances et des développements poussés d'IT⁵⁰.

Leurs spécificités, leur échelle et leur environnement informatique sont nécessaires pour créer ces outils, et c'est ce qui fait qu'ils sont souvent réservés aux gros studios.

Les Fées Spéciales n'a encore rien fait dans cette direction, privilégiant plutôt une triple sauvegarde (deux locales, dont une incrémentielle qui garde des versions sur plusieurs jours, associées à une sauvegarde distante) pour revenir en arrière facilement en cas de mauvaise manipulation.

Ma seule expérience directe de mise en place d'un asset manager remonte à ma participation au programme EUCROMA (European Cross Media Academy) en Allemagne et au Danemark en 2012. Un IT avait pris les choses en mains, partant de ma proposition de nomenclature et avait basé le système de versionning sur le logiciel *Perforce*. Le projet était trop petit pour tirer des conclusions sur le long terme et la viabilité d'un tel choix. Mais je ne recommande pas d'utiliser ce genre de solutions, comme *SVN* ou *GIT*, logiciels de versionning et contrôle des versions, pour l'asset management de l'intégrité d'une production audiovisuelle.

Ces logiciels ont leur utilité reconnue dans le monde de la programmation. J'utilise *git* tous les jours et je m'en suis même servi pour sauvegarder l'avancée de ce mémoire. Mais alors que ces solutions gèrent très bien des fichiers textes, comme peut l'être du code source y compris de dizaines de milliers de fichiers (comme un système d'exploitation), leur gestion des fichiers binaires n'est pas idéale. Et quasiment tous les fichiers générés par une production sont des binaires, comme les images ou les caches. Le *repository*, la structure centrale de sauvegarde, peut être alors vite dépassé.

À la fin de *Moi, Moche et Méchant* InK était encore basé sur *SVN* et semblait arriver aux limites possibles alors qu'il gérait des centaines de milliers d'assets. En fin de journée les publish pouvaient créer de long embouteillages tellement le serveur était sollicité. Il arrivait que le système s'emballé et provoque des erreurs à la chaîne. On pouvait entendre des assistants de production passer de salle en salle demandant aux équipes de ne plus sauvegarder pendant qu'une partie de l'équipe de

⁴⁹ Système de gestion de fichiers : c'est la façon de stocker des informations dans des fichiers comme sur un disque dur

⁵⁰ Pour rappel, l'équipe d'Information et Technologie est en charge des questions matérielles comme le réseau et le stockage

développeurs corrigeait les conflits. La première correction importante d'InK, avant de commencer *Le Lorax*, le projet suivant, a donc été de retirer SVN et passer à un système basé sur le *filesystem*.

À l'heure d'établir un système de contrôle de version ou de sauvegarde, il est utile de faire un audit pour étudier quels types de données on génère. On peut établir une liste des données essentielles, à toujours sauvegarder, et les données secondaires. On peut par exemple partir du principe que tout travail humain a trop de valeur pour risquer de le perdre. Alors que tout fichier généré automatiquement et qui résulte de temps machine, comme du calcul d'images, peut être refait. Il mérite alors une sauvegarde plus légère si les ressources sont limitées.

III.6. Le futur des pipelines

Dans cette dernière sous partie, je propose d'ouvrir la porte aux probables évolutions à venir.

III.6.1. Des offres commerciales et des mutualisations

La professionnalisation du marché présentée en partie I. a abouti à ce que quasiment toutes les problématiques logiciels aient trouvé des réponses sur le marché. Que ce soit des éditeurs proposant des programmes adaptés ou les branches R&D des studios proposant des offres. Toutes ? En vérité la réponse est non, car le pipeline, tel un village de Gaulois, n'a pas bien été servi au vu de toutes ses particularités.

Mais depuis peu et surtout pour accélérer le développement de nouveaux studios ou de studios éphémères, émergent des solutions commerciales. J'évoquerai trois solutions qui semblent sortir du lot.

Shotgun, déjà bien présenté pour la question de suivi de production, était le plus à même d'étendre sa stratégie à la gestion même des fichiers tels qu'un pipeline le demande. La base de données connaissait déjà tous les éléments à fabriquer, il n'y avait qu'un pas à faire pour faciliter la gestion de fichiers associés. C'est ainsi que les développeurs ont commencé à travailler sur `Tank` depuis renommé `Shotgun Toolkit` un framework complet de gestion des fichiers basé sur Shotgun. Pour avoir vu Toolkit en production, je trouve que ce système a encore une marge de progression devant lui. La dépendance à Shotgun, souvent installé sur des serveurs aux États-Unis, rend le système particulièrement lourd et lent⁵¹. Ceci dit, l'intérêt vient de son éditeur, Autodesk, qui propose une chaîne de fabrication complète *clés en main* : la production (shotgun), le pipeline (toolkit) et tous les logiciels 3D nécessaires pour faire des images de synthèse. Les services étant particulièrement bien intégrés et connectés entre tous ces logiciels.

À côté de ça, le studio français Circus a créé une filiale, Texels, qui propose le logiciel `Jarvis`. Jarvis est une sorte de super outil d'automatisation plus qu'une base de pipeline. Ici on reprend un modèle en parti présent dans des solutions comme InK ou encore plus dans le logiciel dédié de BUF Company, [BProd](#). C'est un système Nodal composé de noeuds qui correspondent à des actions précises. Un fichier 3D à éditer est un noeud par exemple. L'avantage de ce système est la réaction en chaîne qu'il développe : en effet toutes modifications faites sur un de ces éléments, ou noeud, peut entraîner l'activation des noeuds suivants. Par exemple une fois une

⁵¹Néanmoins Shotgun [a annoncé](#) fin avril l'ouverture de serveurs sur tous les continents pour mieux répondre à la demande)

animation mise à jour il est possible de générer un playblast, d'exporter un cache, de relancer des rendus, etc.

Jarvis n'amène pas vraiment de nouveautés dans ce que l'on voyait dans certains gros studios, mais il a la particularité d'être disponible pour toutes les structures y compris de moindre ampleur.

Tout début mai 2017, l'incontournable éditeur The Foundry (Nuke, Mari, ...) annonce⁵² Elara dont le but est de proposer le pipeline comme un service *cloud* pour les équipes décentralisées (cf Partie II.). Peu d'informations existent à l'heure de la rédaction de ce mémoire, les tarifs et la date de disponibilité ne sont pas encore connus. Il faudra surveiller de près cette solution que propose un tel éditeur.

Ce n'est que le début de ces solutions commerciales et il reste beaucoup à faire pour standardiser le pipeline lui-même. Mais il semble qu'en attendant beaucoup de studios auront encore leurs propres solutions maison.

III.6.2. Les nouveaux standards: USD, EXR, ABC, ...

Depuis quelques années, nous avons vu émerger 4 standards qui jouent un rôle important dans les pipelines. 3 d'entre eux sont déjà largement implémentés dans les logiciels et studios et ils ont joué un fort rôle structurant des pipelines. Le dernier est en passe de devenir aussi important :

- `openExr` est devenu le format d'image par excellence et a permis de structurer les couches de travail
- `Alembic` s'est imposé comme le format d'échange des données 3D
- `OpenSubdiv` standardise la représentation finale des géométries
- et l'`Universal Scene Description (USD)` semble devenir le standard de structuration même des données de tout type dans une scène 3D (gestion des dépendances extérieures et des particularités d'une scène).

Ces formats sont nés chez des géants de la 3D : ILM, Sony et Pixar, largement suivis et aidés par d'autres géants et laboratoires de recherches. Le choix d'une distribution libre aidant grandement. D'autres formats libres et standards, non listés ici sont spécifiques à des tâches particulières, comme `OpenVDB` pour la standardisation des volumes type fumée, feu, eau, etc.

Il est intéressant de remarquer que c'est la distance, de plus en plus marquée, entre l'outil 3D et la donnée manipulée, qui est à l'origine de ces nouveaux formats. Au contraire du modèle classique où

⁵²www.3dvf.com/actualite-20047-the-foundry-presente-en-video-elara-son-futur-service-dans-cloud.html

un format était lié à un logiciel précis ces standards permettent aux données de circuler plus facilement d'un logiciel à l'autre profitant des fonctionnalités de ces derniers.

Les fichiers structurés avec des standards libres et ouverts permettent que n'importe quel outil puisse le lire et l'écrire. L'interopérabilité est ainsi de plus en plus assurée et il devient plus facile de structurer un pipeline efficace en venant greffer n'importe quel logiciel nécessaire à la fabrication du projet. On a alors de moins en moins de formats exotiques à gérer. Le tout sans se poser trop de questions de conversions ou de passerelles d'un programme à l'autre.

C'est sans surprise que l'industrie des effets spéciaux s'est appropriée très vite ces standards. Car ces formats permettent aussi de simplifier les coproductions, facilitant les échanges entre studios qui n'ont pourtant pas les mêmes pipelines. Et, rappelons-le, rares sont les studios qui fabriquent l'intégralité des effets d'un film.

III.6.3. Pérenniser son pipeline

La question de la pérennisation du pipeline dans le temps et de l'harmonisation des méthodes peut sembler évidente. Pourtant, un pipeline est organique comme nous l'avons vu en partie I. : il évolue au gré des méthodes, des outils et des technologies pour mieux supporter les projets en cours. De plus, les projets changent, les technologies évoluent, les gens pour les maintenir partent pour de nouveaux horizons et leurs remplaçants viennent avec de nouvelles envies (cf. III.2.3.5. Le défi de la capitalisation des savoirs et des savoir-faire ALGM). On peut aussi noter la distinction entre les studios qui se survivent dans le temps et les studios éphémères, faits pour supporter un projet particulier avant disparaître. Ces éléments remettent donc en perspective la question de la nécessité de pérennisation d'un pipeline.

Un TD qui sortait de la production d'un long métrage m'expliquait avoir discuté de cette pérennisation avec le producteur du film. Il voulait expliquer le gain, pour un prochain projet, à maintenir une petite équipe en place pour mutualiser et réutiliser les acquis. Mais le producteur avait un argument financier simple : « si j'ai pu faire ce film pour X millions d'euros, je sais que je pourrai refaire le même film pour X millions d'euros en recommençant à zéro. Pourquoi voudrais-je me rendre la vie plus compliquée ? ».

Dès lors, nous pouvons nous demander si la pérennité des pipelines est l'apanage des équipes techniques ? D'autre part, il est intéressant de noter ici l'usage du terme « compliqué » au regard des enjeux de simplification, de mutualisation et de standardisation que nous avons précédemment associés à la création d'un pipeline.

Heureusement ce discours n'est pas celui de tous les producteurs. Beaucoup de studios ont misé sur le long terme avec leurs équipes de R&D et les équipes de TD.

Néanmoins, fatigué de recommencer les mêmes outils à chaque fois que je change d'emploi, lorsque s'est posée la possibilité de cofonder le studio Les Fées Spéciales, une de mes conditions était de publier le code source des logiciels que nous développons sous licence libre. Et ceci afin de préserver ces développements et rendre à la communauté un peu de ce qu'elle nous apporte. Combien d'entreprises, même de grandes entreprises, ont fermé, coulant avec toutes leurs connaissances, car le code appartenait à la structure ? Combien de TDs réinventent la roue tous les jours ? Alors que si nous partageons et mutualisons les solutions trouvées, nous sortirions d'un éternel recommencement pour nous concentrer un peu plus sur des choses innovantes.

C'est pour se positionner par rapport à ces constats et questions que le code développé chez les Fées Spéciales est libre par défaut. Certes il doit être nettoyé et commenté pour être distribué, mais il pourra survivre à l'histoire de la structure et de ceux qui la composent.

III.7. Conclusion de la troisième partie

Dans cette dernière partie, j'ai pu décrire quelques-uns des outils que j'ai pu mettre en place dans différents contextes. Ces outils s'inscrivaient souvent dans un cadre de travail et des projets bien définis, sans lesquels je n'aurais pas pu expérimenter.

L'héritage des pipelines, ces écosystèmes mis en place dans les studios au cours du temps sont autant d'aides pour progresser et aller plus loin, comme un chemin balisé. Lorsque j'ai quitté Mac Guff pour Eucroma, puis de nouveau pour In Efecto, la sensation de tout recommencer sur un terrain vierge n'était vraiment pas agréable. Habitué à avoir des outils sur lesquels me reposer, il fallait reprendre les fondamentaux. Mais faire table rase permet de recommencer sur nouvelles bases et proposer des solutions utilisant des technologies et des procédures plus modernes.

Conclusion

Comme il a été annoncé en introduction de ce mémoire, j'ai souhaité prendre à travers lui du recul sur les questions qui m'animent depuis bientôt dix ans. Cette première analyse aura en effet été pour moi l'occasion de formaliser à quel point un pipeline peut être pluridisciplinaire, plurivoque, polysémique et protéiforme.

En effet, il me semble qu'en faire une définition générale n'a pas grand sens face à la variété des cas de figure que l'on peut découvrir au sein des projets que les pipelines servent.

Chacun d'entre eux est une possibilité d'étudier une configuration particulière de pipeline, une opportunité de comprendre la richesse des applications de ce concept simple au départ. Par la même occasion, cette démarche d'étude de « cas » permet également de mettre en lumière de nombreuses problématiques du métier qui restent encore mal étudiées et analysées, comme la façon dont nous définissons certains termes du jargon professionnel. Cette question mériterait d'ailleurs clairement d'être creusée dans des recherches ultérieures tant la question reste ouverte.

La notion de protéiforme évoquée ci-dessus est aussi incarnée par la nécessaire adaptation du pipeline aux différents départements de production ainsi qu'aux nouvelles situations, l'urgence étant souvent le maître mot des pipelines TD. D'autre part, pour faire en sorte que le projet se termine bien, les pipelines doivent faire communiquer des gens qui n'ont rien à voir les uns avec les autres. En effet, tous les acteurs de la fabrication d'une œuvre de collaboration audiovisuelle, que ce soit les techniciens, les artistes ou les producteurs, tous ont besoin d'outils communs de discussion et de transmission.

Il est aussi un autre facteur qui demande toujours plus de polyvalence et d'optimisation aux pipelines : ce sont les budgets alloués aux films d'animation. En effet, ces derniers ont tendance à se réduire aujourd'hui, comme nous le montre notamment l'exemple de Michel Ocelot, auteur de *Kirikou*, qui a moins du budget pour son prochain *Dilili à Paris*, lequel est pourtant plus ambitieux que son précédent *Azur et Asmar*. Or, il est donc souvent nécessaire dans ce contexte de réduire la voilure pour pouvoir réaliser des projets. Dès lors, avoir un bon pipeline, c'est-à-dire un pipeline qui permet les diverses optimisations nécessaires à la gestion du temps, des coûts, mais aussi d'éventuels multisites de production, devient crucial.

Les entreprises n'ont donc pas tort d'investir et de réunir les bonnes personnes sur ces questions. Illumination Mac Guff illustre bien cet enjeu en étant passé de 2 TD en 2010, à près d'une vingtaine

en 2017. Des événements comme les Rencontres Animation, Développement & Innovation montrent eux aussi à quel point les besoins en TD se font de plus en plus importants. Quant aux Fées Spéciales, nous avons fait le choix d'allouer d'importantes ressources (3 postes à temps plein pour une équipe totale d'une quinzaine de personnes) sur le projet Dilili à Paris afin de pallier son budget réduit. Bien que cela paraisse beaucoup pour le démarrage d'un jeune studio comme le nôtre, nous n'aurions pas pu utiliser les méthodes et ressources classiques, il nous a fallu nous adapter.

Pour conclure ce mémoire, je tiens également à aborder la question de la formation. Même si nous ne l'avons pas abordée, il est à noter qu'elle n'en est pas moins une question fondamentale au regard des enjeux toujours plus importants dont les pipelines sont les dépositaires. Je donnerai simplement ici quelques exemples prouvant que l'offre scolaire s'intéresse de plus en plus à des compétences liées aux pipelines.

Ainsi, des écoles privées, comme ArtFx, ont récemment créé des départements TD. L'école Méliès avait tenté d'ouvrir une licence pro DAAJV (Technique et Activités de l'Image et du Son Spécialité Développement d'Applications pour l'Animation et le Jeu Vidéo) en partenariat avec l'Université d'Evry. C'est une démarche intéressante, mais qui n'avait pas trouvé d'écho dans les studios. À l'ESMA, les étudiants en animation sont de plus en plus demandeurs pour apprendre des langages de programmation. Depuis 3 ans je collabore donc avec l'école en proposant des cours d'introduction à Python. Aujourd'hui l'ESMA cherche à pérenniser ce cours en créant un poste à plein temps qui serait reparti entre la formation et un poste de TD transversal aux projets étudiants. ATI fait quant à elle clairement figure de précurseur avec sa formation hybride atypique. Sa longueur d'avance et sa philosophie en font un terreau particulièrement fertile pour de futurs TD.

Maintenant que ce tour d'horizon des formations est fait, et malgré la pertinence de ces démarches, j'ajouterais cependant que le métier de TD ne s'apprend pas vraiment sur les bancs de l'école. Il me semble en effet, d'une part, que l'on devient TD grâce à une curiosité un peu touche-à-tout (le code, la fabrication d'images, la production et les équipes techniques et artistiques) qui peut dépasser le cadre des formations. Et que d'autre part, ce sont surtout les expériences réelles sur le terrain qui forgent la réactivité, la prise de recul et la créativité si précieuses pour les Technical Directors.

Cette nécessité du terrain engage donc le milieu professionnel. Il en va en effet de la nécessité et de la responsabilité des studios de faire confiance aux jeunes recrues. D'autant plus que c'est une aventure humaine qui garantit à ces mêmes studios de continuer à expérimenter et à chercher de nouvelles façons de faire.

Ce sont en effet les ressources humaines qui sont le garant d'une réflexion proactive et innovante, la force vive des pipelines conçus. Je souhaite insister de nouveau sur ces points et ajouter que, selon moi, la question pour les studios devrait être systématiquement : j'investis sur « qui » plutôt que sur « quoi » ? Car il me semble que l'investissement ne doit pas se faire sur un modèle x ou y de pipeline, mais plutôt sur une équipe efficace.

Car les modèles choisis, aussi parfaits soient-ils à un instant T, risquent fort d'être obsolètes avant la fin du prochain projet. Ce sont typiquement le genre de pipelines détestés par les graphistes en production, car trop rigides. Par contre, une équipe capable de s'adapter fera en sorte de maintenir et de transformer au mieux le pipeline au projet qu'il supporte.

Si nous regardons à présent l'avenir de la profession, nous pouvons noter que les prochains défis se concentreront très certainement sur la gestion d'une globalisation toujours plus grande des productions avec le lot de contraintes économiques et humaines que cela sous-entend. La décentralisation et l'arrivée du *cloud* comme une première réponse à ces contraintes et comme une réalité tangible des productions à venir seront, elles aussi, au cœur des questionnements des équipes de support. Les TD ne vont pas manquer de travail.

D'ici là les studios doivent continuer à choisir leurs chantiers en essayant d'appliquer une règle simple : développer des solutions pour les cas les plus génériques. Car ce sont ces solutions qui ne vont pas dans le cas particulier (les micropipelines), qui seront les fondations à un pipeline protéiforme. Si l'on crée un outil pour une tâche qui ne se présente qu'une fois sur cent, n'est-on pas en train de mal utiliser les ressources ?

Enfin, ce mémoire est aussi l'occasion de se rappeler que, quoi qu'il en soit, nous développons des outils en nous reposant sur les briques solides que nous offrent les langages de programmation, les systèmes d'exploitation et les logiciels que nous utilisons tous les jours. Sans eux nous ne ferions pas grand-chose, ce qui me pousse à conclure en citant Newton :

« Si j'ai vu plus loin, c'est en montant sur les épaules de géants. »

Bibliographie

- Academy Originals. (s. d.). *Credited As: Stop Motion Animator*. Consulté à l'adresse <https://www.youtube.com/watch?v=ij3IbplMisA>
- Butler, G., Langlands, A., & Ricklefs, H. (2008). A pipeline for 800+ shots (p. 72). Présenté à ACM SIGGRAPH 2008 talks, ACM. <https://doi.org/10.1145/1401032.1401125>
- Catmull, E., & Wallace, A. (2014). *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True Inspiration*. Transworld Publishers Limited.
- Chuang, R., & DeBry, D. (grue). (2009). Creative collaboration: effective CG pipelines: any size, any place (p. 6). Présenté à ACM SIGGRAPH ASIA 2009 Courses, ACM. <https://doi.org/10.1145/1665817.1665823>
- Dunlop, R. (2014). *Production Pipeline Fundamentals for Film and Games*. CRC Press.
- George-Molland, A.-L. (2007, décembre 12). *La collaboration au cœur du processus de création des œuvres audiovisuelles numériques : analyse des transformations apportées par le développement des technologies et par l'évolution des savoir-faire*. Paris 8.
- Grant, D., Stein, M., Lavietes, S., Gibbs, J., & Ford Grant, B. (2014). *SIGGRAPH University : « The Digital Production Pipeline »*. Consulté à l'adresse <http://www.siggraph.org/content/siggraph-university-digital-production-pipeline-0>
- Iwerks, L. (2007). *The Pixar Story*.
- Kenneally, C. (2012). *Side by Side*.
- Lewis, J. P. (2001). Limits to software estimation. *ACM SIGSOFT Software Engineering Notes*, 26(4), 54-59. <https://doi.org/10.1145/505482.505490>
- O'Neill, R., Mavroidis, P., & Ho, M.-H. (2007). openPipeline: teaching and implementing animation production pipelines in an academic setting (p. 33). Présenté à ACM SIGGRAPH 2007 educators program, ACM. <https://doi.org/10.1145/1282040.1282075>
- Pintea, P. (2003). *Effets spéciaux: un siècle d'histoires*. Minerva.
- Polson, B. (2014). CG pipeline design patterns (p. 29-29). Présenté à Proceedings of the Fourth Symposium on Digital Production, ACM. <https://doi.org/10.1145/2633374.2637729>

- Polson, B. (2015). A conceptual framework for pipeline (p. 51-52). Présenté à Proceedings of the 2015 Symposium on Digital Production, ACM. <https://doi.org/10.1145/2791261.2791272>
- Ross, S., & Squires, S. (2013). *The State of the VFX Industry and where do we go from here*. NAB SuperMeet. Consulté à l'adresse <https://vimeo.com/66487005>
- Schwank, A., James, C. J., & Micilotta, T. (2016). The trees of The Jungle Book (p. 21). Présenté à ACM SIGGRAPH 2016 Talks, ACM. <https://doi.org/10.1145/2897839.2927428>
- Vanns, J., & Carey, A. (2016). A fully cloud-based global visual effects studio (p. 72). Présenté à ACM SIGGRAPH 2016 Talks, ACM. <https://doi.org/10.1145/2897839.2927432>
- Welker, C. (2015, juin 11). *La fabrique des "nouvelles images" : l'émergence des images de synthèse en France dans la création audiovisuelle (1968-1989)*. UNIVERSITÉ SORBONNE NOUVELLE – PARIS 3. Consulté à l'adresse <https://tel.archives-ouvertes.fr/tel-01488940>

Remerciements

Anne Laure George Molland et Damien Picard, pour leur temps et nos matinées recherches,
Marie Saby pour son précieux temps et ses remarques avisées,

Étienne "Chex" Pêcheux, pour son amitié et son mentorat,

Virginie Guilminot, Ève Machuel et Éric Serre pour leur confiance bienveillante et leur amitié qui
m'ont permis de suivre une nouvelle aventure féerique,
Mes collaborateurs aux Fées avec qui je continue d'apprendre,
Les Fées Spéciales pour avoir soutenu cette recherche,

Nord Ouest, Studio O et Mac Guff ligne qui me poussent tous les jours depuis plus de 2 ans à
chercher et trouver des solutions qui alimentent ma réflexion,
Mais aussi mes anciens et premiers employeurs, Les Armateurs, Autour de Minuit, Mac Guff Ligne,
Illumination Mac Guff, qui m'ont fait confiance et m'ont permis de tant apprendre,

Jean Francois Jego pour son suivi malgré mes contraintes, et aussi Cedric, Chu Yin, Nadia et
l'ensemble l'équipe enseignante pour m'avoir permis de ne jamais quitter ATI

Damien "Dee" Coureau, pour m'avoir confié kabaret.naming,
Dorian Fevrier, pour sa folie qui inspire bien des TD,

Kevin et Maxime pour leurs conseils,
Laure pour ses encouragements et sa patience,
ma famille qui ne comprends pas toujours de quoi je parle mais que ça n'ennuie pas,

ma team ATI 2006-2009, ceux que j'ai pu oublier dans cette version...,
et la promo d'ATI 2014-2017, ... ma troisième promo d'ATI.

À vous tous : MERCI !