

Université Paris 8

Master Création Numérique

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

**Création et développement d'outils de rendu
pour l'expression artistique en animation 3D
temps-réel**

Sophia Mouajjah



Mémoire de Master 2

2018 - 2019

« L'expression recrée et métamorphose »

Maurice Merleau-Ponty

Résumé

Alors que la recherche du photoréalisme en images de synthèse revêt une place importante au sein du champ de développement technologique et technique actuel, d'autres manières d'utiliser et d'exploiter les possibilités permises par le rendu 3D pour le film restent encore à explorer. L'enjeu de mes recherches sera ainsi d'expérimenter et de développer différentes techniques de rendu dit « expressif » en animation temps-réel afin de permettre aux artistes de s'approprier plus librement le médium numérique et de transmettre le plus intuitivement possible leur subjectivité. Pour mettre à bien ces outils, il s'agira d'abord de s'interroger sur les divers procédés mis en œuvre par les artistes au cours de l'histoire via l'emploi des médias traditionnels pour exprimer une vision ou une émotion singulière. Nous étudierons notamment comment les effets de matière, de couleur et de lumière sont régulièrement employés par les peintres pour inventer et développer un style visuel propre. On posera ensuite la question de la spécificité du médium numérique, en particulier dans le domaine du temps-réel, dans le but d'exploiter au mieux les possibilités offertes par celui-ci. Nous tenterons ainsi de produire de nouveaux outils utilisables pour l'animation 3D temps-réel afin donner un maximum de libertés aux artistes pour qu'ils puissent faire passer un message ou une émotion de manière efficace.

Abstract

While the research of photorealism in computer graphics takes an important place in the actual technical and technological development field, there are other ways of using and taking advantages of the possibilities offered by 3D rendering for movies that are still waiting to be investigated. The main issue of these studies will be to experiment and develop various expressive render technics destined to realtime animation in order to allow the artists to appropriate the digital medium in a more freely way and to convey their subjectivity as intuitively as possible. To implement those tools, I will firstly question the various processes undertaken by artists throughout history by the use of traditionnal medias to express a vision or a singular emotion. In particular, we will study the way material, color and light effects are regularly used by painters to invent and develop their new proper visual styles. Next, we will raise the question of the specificity of the digital medium, more particularly in the field of realtime rendering, in order to make the best use of the possibilities it can offer. Thus, we will try to implement tools that can be used for realtime 3D animation in order to provide maximum freedom to artists so that they will be able to express an emotion or a message in an efficient way.

Remerciements

A ma famille pour tout son soutien,
A Gaëtan Henry pour son merveilleux travail lors du projet intensif,
A Chu-Yin Chen pour son suivi et ses précieux conseils,
A l'ensemble de l'équipe ATI pour l'enseignement et les
nombreuses connaissances qui m'ont été transmises

Table des matières

Introduction	11
I. Etude des procédés expressifs employés par les artistes à l'aide des médias traditionnels et numériques	13
1. L'expression artistique : une forme de communication particulière	13
1.1 Au delà de la représentation : l'expression comme prise de conscience et clarification de l'émotion	14
1.1.1 Expression et représentation	14
1.1.2 Remise en ordre, prise de conscience et clarification	15
1.1.3 Le rôle de l'intuition	16
1.2 L'expression artistique peut être vue comme une forme de langage	17
1.2.1 L'importance de l'appropriation et de l'interprétation dans le domaine de l'expression	17
1.2.2 L'utilisation des signes et des symboles pour exprimer	18
1.2.3 La particularité de l'expression artistique comme moyen de communication	19
1.3 L'importance de l'expérience et la place du corps comme interface dans le processus d'expression artistique	20
1.3.1 Corps, sens et motricité	20
1.3.2 L'œuvre d'art comme résultat d'une action sur la matière	21
1.3.3 L'expression comme expérience et rencontre entre l'œuvre et le public	21
2. Générer une œuvre expressive à l'aide des médias traditionnels et numériques	23
2.1 L'expressivité dans l'art traditionnel	23
2.1.1 Contexte historique de l'expressivité dans l'histoire art et émergence de l'individu artiste	23
2.1.2 La couleur comme puissance expressive	24
2.1.3 La simplification et la stylisation	28
2.1.4 L'emploi du matériau, l'importance de la matière et du médium	30
2.2 Les spécificités du médium numérique	33
2.2.1 Simulation et transposition	33
2.2.2 L'automatisation, un gain de temps pour les artistes	34
2.2.3 L'essor de la génération procédurale comme terrain d'expérimentation et d'hybridation	35
2.2.4 La question de la matérialité du numérique et de la technicité	36
2.3 Le style et le rendu artistique dans les réalisations numériques aujourd'hui: exemples de diverses œuvres s'inspirant des techniques traditionnelles dans un but expressif	37
2.3.1 Film	38

2.3.2 Jeu-vidéo	40
2.3.3 Installations interactives	43
3. Les techniques de rendu expressif	43
3.1 Le rendu en images de synthèse	44
3.1.1 Définition et fonctionnement global	44
3.1.2 Le choix du rendu temps réel	45
3.1.3 Le pipeline graphique et la méthode de rendu par rasterisation	46
3.2 Le rendu expressif	54
3.2.1 Définition.....	54
3.2.2 Caractéristiques relatives au rendu expressif.....	55
3.2.3 Apports techniques et utilisations	58
3.3 Tour d’horizon de techniques de rendu expressif et étude du lien entre l’outil et l’utilisateur	59
3.3.1 Filtres et traitement d’image	59
3.3.2 Rendu à base de marques ou SBR	62
3.3.3 Transfert de style par deep learning.....	66
3.3.4 Méthode par création de shell ou coquille.....	66
3.3.5 Rendus expressifs basés sur la lumière.....	68
II Expérimentations et créations d’outils de rendu expressif aux effets variés, s’inspirant de techniques traditionnelles.....	70
1. Permettre l’expression artistique à travers divers matériaux virtuels grâce aux outils de traitement d’image	71
1.1 Généralités et principes du <i>post-processing</i> par traitement d’image en temps réel	71
1.1.1 Rôle et apports dans le domaine du temps réel	71
1.1.2 Précisions et descriptions techniques: comment développe-t-on un shader dans Unity?	72
1.1.3 Comment réaliser un effet de post processing via Unity?	73
1.2 Développement de la palette de matériaux	73
1.2.1 Touches de peinture	74
1.2.2 Aquarelle	77
1.2.3 Ajout de contour, crayonné.....	80
1.2.4 Turbulences	81
1.3 Retour d’expérience et efficacité	82
1.3.1 Apports de la méthode: un outil léger qui aide l’artiste à générer une variété de styles visuels expressifs	82

1.3.2 Les limites de l'outil: un manque de contrôle au niveau local	82
1.3.3 Appréciations sur la méthode employée et définition des nouveaux enjeux	83
2. Algorithmes pour générer des peintures de manière procédurale	83
2.1 Choix de la technique	83
2.1.1 Principe du rendu basé sur strokes.....	83
2.1.2 Problématiques et enjeux.....	84
2.1.3 Intérêt pour la création artistique	85
2.2 Première variante de l'outil.....	86
2.2.1 Gagner en performances et en maniabilité avec les compute shaders	86
2.2.2 Structure principale de la stroke.....	88
2.2.3 Transformations et placement des touches de peinture	89
2.3 Deuxième variante: Wind Painter.....	101
2.3.1 Retrouver la trace de l'artiste par l'évocation du mouvement.....	101
2.3.2 Comportement des strokes au cours du temps.....	103
2.3.2 Une peinture générative guidée par le son ou la musique	106
2.4 Retour d'expérience et efficacité	107
3. Peindre une géométrie animée de manière procédurale.....	107
3.1 Expérimentations en précalculé	108
3.1.1 Méthode manuelle par placement de plans.....	108
3.1.2 Méthode par motion blur	109
3.1.3 Méthode par utilisation de techniques de <i>fur</i>	110
3.2 Instanciation sur mesh 3D	111
3.2.1 Récupérer les positions des points sur un mesh après déformation du skinning	111
3.2.2 Répartir les instances de manière efficace, scattering	112
3.2.3 Orientation des <i>strokes</i>	114
3.2.4 Apparence des <i>strokes</i>	115
3.2.5 Gérer la lumière.....	116
3. Retour d'expérience	117
III Réalisation artistique : travail sur la matière céleste par la couleur et la lumière comme sources d'expression et d'immersion.....	119
1. Un film immersif.....	119
1.1 Présentation du projet	119
1.2 Références artistiques	120
1.3 Objectifs techniques et esthétiques	120

2. Le rendu de ciel étoilé comme exercice de style	121
2.1 Historique et traitement du ciel de nuit en art traditionnel	121
2.2 Création du ciel procédural	123
2.3 Gestion de variantes de ciel afin de produire des effets visuels accompagnant la narration et exprimant les états mentaux de l'astronaute	124
3. Mettre en valeur les personnages grâce au shading et les FX	128
3.1 Identifier l'astronaute par la couleur	128
3.2 Constellations	129
3.3 Silhouettes	130
3.4 Flat shader	131
4. Retour sur les problèmes rencontrés, appréciation globale et améliorations à apporter	132
4.1 Rendu et <i>shading</i>	132
4.2 Post-processing	132
4.3 Particules	133
Conclusion	134
Bibliographie	138
Webographie	141
Filmographie	144
Ludographie	145
Table des illustrations	146

Introduction

La notion d'expression dans le domaine de la création artistique s'est continuellement transformée au fil des époques, suivant l'apparition de mouvements artistiques et de découvertes scientifiques et techniques. Une des fonctions couramment attribuées à l'art est de transmettre une vision intime et personnelle du monde en apportant une certaine subjectivité qui est propre à chaque artiste. Cette subjectivité se traduit généralement par la manière avec laquelle celui-ci emploie les outils et travaille la matière pour produire une œuvre originale, ce qui permet ainsi d'engendrer un style particulier et reconnaissable. Dans les productions numériques actuelles, la touche personnelle de l'artiste peut parfois avoir tendance à s'effacer au profit d'un réalisme exacerbé. Le but de cette recherche sera de tenter différentes approches afin de préserver au mieux cette touche propre à l'artiste. Cela va ainsi nous conduire à différents questionnements.

Comment les techniques de rendu expressif peuvent permettre de transmettre une vision ou une émotion en conservant la sensibilité esthétique de l'artiste au sein d'une production en animation 3D temps-réel ? Par quels moyens les outils numériques peuvent-ils être mis au service des artistes pour leur permettre d'inventer de nouvelles esthétiques ? Comment s'approprier les médias traditionnels tels que l'art pictural afin d'apporter une matérialité au sein des œuvres d'art numérique ?

Pour répondre à ces problématiques, nous allons d'abord étudier certaines théories esthétiques portant sur le rôle de l'expression dans le domaine de la création artistique en nous interrogeant sur la différence entre expression et représentation et sur le concept d'expression artistique comme moyen de communication. Nous évoquerons ainsi la place qu'occupent les corps au sein du processus d'expression artistique et l'importance de la réception chez le spectateur pour que l'œuvre, qui sera ici traitée comme une expérience, puisse effectivement être qualifiée d'expressive.

Par la suite, certaines recherches théoriques seront effectuées afin de déterminer les différents procédés employés par les artistes dans le but d'exprimer une émotion, une idée ou une vision du monde subjective et personnelle. Pour cela, nous allons porter notre attention spécifiquement sur certains mouvements de la fin du XIXe et du XXe siècle tels que

l'expressionnisme, le fauvisme ou encore l'impressionnisme. Nous nous intéresserons aux mécanismes qui permettent à l'artiste de développer un style propre, c'est à dire sa manière particulière d'utiliser les outils pour produire une œuvre. L'enjeu sera de repérer les éléments qui peuvent être appropriés et transposés vers le médium numérique en prenant en compte les spécificités de ce dernier. On pourra ainsi exploiter certaines dimensions apportées par l'outil numérique telles que l'automatisation et la génération procédurale afin de créer de nouvelles formes esthétiques. Cela passera notamment par l'élaboration d'un état de l'art non exhaustif de certaines techniques et utilisations d'outils de rendu expressif dans différents domaines tels que le film d'animation, le jeu-vidéo ou encore les installations interactives. Il faudra également s'intéresser à ce que le numérique a changé en terme de processus de création ainsi qu'au rapport entre l'artiste et les outils de création artistique.

En seconde partie, plusieurs expérimentations exploitant différentes techniques de rendu expressif seront présentées. Il s'agira de permettre à l'utilisateur de contrôler un certain nombre de paramètres de rendu en temps réel de manière totalement procédurale. Le but sera de générer une grande diversité d'effets visuels non réalistes par le biais d'une interface, en facilitant le travail de l'artiste pour qu'il puisse se préoccuper uniquement de l'aspect visuel et artistique de l'œuvre. On verra par exemple comment ajouter de la matière au rendu final grâce à au traitement de l'image ou par instanciation de touches de peinture virtuelle. Ces procédés pourront être employés dans le but de développer une ambiance ou une atmosphère particulière via le rendu et donc d'exprimer ou de provoquer des sensations et des émotions chez le spectateur. Il sera ainsi possible de bénéficier de certaines dimensions spécifiques au médium numérique telles que la présence du temps pour engendrer des visuels dynamiques et traduire une intention spécifique, propre à l'artiste. Certaines contraintes et difficultés liées à l'utilisation du temps réel pourront être ainsi explicitées.

Ces recherches se termineront par la rétrospective d'un court-métrage d'animation immersif réalisé au cours de l'année. Je présenterai ma démarche artistique et m'interrogerai sur les différentes problématiques liées au rendu temps réel pour le film et à l'immersion. On verra ainsi quels moyens peuvent être développés pour faire vivre une expérience au spectateur en le transportant dans une narration guidée par divers effets visuels stylisés. Cette réalisation consistera en une forme d'exercice de style dans lequel

chaque scène devra exprimer une intention précise par le biais du travail sur le *shader* et la lumière.

I. Etude des procédés expressifs employés par les artistes à l'aide des médias traditionnels et numériques

1. L'expression artistique : une forme de communication particulière

Nous allons d'abord tenter de définir le concept d' « expression artistique » afin de mieux cerner les enjeux du sujet et de pouvoir par la suite élaborer des outils adaptés à cette fin. En premier lieu, intéressons nous à l'idée même d'expression pour ensuite étudier son rôle dans le domaine de la création artistique. L'Encyclopédie Universalis la définit comme étant un « ensemble de données perceptives offertes par un être à un moment donné du temps et qui fonctionnent comme des signes manifestant, révélant ou trahissant les émotions, les sentiments et les intentions de cet être »¹. Elle consiste donc en une organisation d'éléments portant une signification intentionnelle ou émotionnelle pouvant être transmise au monde extérieur par le biais de la perception. Cette dimension de « signe » est importante car on verra plus tard que les formes d'expression partagent toutes une certaine similarité avec le langage. Le mot expression a des origines étymologiques latines: « *expressio* », terme qui définit l'acte d'extraire en pressant. L'expression consisterait alors en une forme de mouvement, avec une notion de sortie, de passage de l'intérieur vers l'extérieur, ou encore de manifestation, « ce par quoi quelque chose se manifeste »². Qu'en est-il alors de l'expression artistique ? Le Larousse définit l'expression artistique comme une « manière de s'exprimer, de représenter quelque chose par une technique artistique »² et une « qualité de force, de vivacité par laquelle un artiste représente ses sentiments ; style »².

¹ Françoise ARMENGAUD, « **EXPRESSION, philosophie** », *Encyclopædia Universalis* [en ligne], consulté le 6 mars 2019. URL : <http://www.universalis.fr/encyclopedie/expression-philosophie/>.

² *Larousse* [en ligne], consulté le 2 mars 2019.

URL : <https://www.larousse.fr/dictionnaires/francais/expression/32326>.

On peut ainsi la voir comme étant une façon personnelle et propre à l'artiste de rendre une émotion ou un message au sein de ses œuvres. Nous pouvons par ailleurs noter cette idée de « manière » ou de « style » qui sera importante dans notre analyse. On verra en premier lieu que l'expression peut être vue comme une manière de prendre conscience, de dénouer et d'éclaircir une émotion, puis nous étudierons son lien avec le langage ainsi que l'importance particulière qui doit être accordée au rôle du spectateur. Enfin, tout en nous appuyant sur différentes théories esthétiques, nous aborderons la place de l'expérience et du corps dans l'expression artistique.

1.1 Au delà de la représentation : l'expression comme prise de conscience et clarification de l'émotion

1.1.1 Expression et représentation

Les conceptions sur l'art et l'esthétique ont évolué au cours des siècles. Durant l'antiquité, Platon considérait l'œuvre d'art comme étant une copie trompeuse de la réalité. Pour lui, l'œuvre imite le monde sensible qui lui-même est une représentation factice de la réalité, autrement dit du monde des Idées. Il percevait également l'art comme étant une forme d'artisanat, l'artiste étant à cette période considéré comme étant un « dêmiourgos », « demos » signifiant peuple et « ergos » le travail. Au Moyen-âge, en occident, l'art était d'abord vu comme un métier : les artistes appartenaient à une confrérie et travaillaient sur commandes. On avait alors une vision très technique de l'art: l'œuvre devait avoir une fin. C'est ce à quoi s'oppose le philosophe et historien Robin George Collingwood³. Il s'interroge sur la définition de l'art comme représentation. Collingwood distingue notamment l'artisanat de l'art : pour lui, ce n'est pas la qualité avec laquelle l'artiste parvient ou non à mimer la nature qui permet de définir le fait qu'elle puisse être ou non œuvre d'art. Ces caractéristiques ne permettent pas de déterminer ce qu'est l'art puisqu'elles n'offrent qu'une vision incomplète de celui-ci. De la même manière, le philosophe Maurice Merleau-Ponty est en opposition avec la thèse selon laquelle un peintre peut figurer uniquement ce

³ Stecker, R., et T. Gracyk. *Aesthetics Today: A Reader*. Aesthetics today. Rowman & Littlefield Publishers, 2010. https://books.google.fr/books?id=UgL_QQAACAJ.

qui existe, la peinture étant donc seulement réduite à un art de représentation. D'après lui, au contraire, l'artiste n'a pas pour objectif unique de représenter ou de mimer.

C'est ici qu'entre en jeu le concept d'expression. L'expression, d'après Merleau Ponty, est une « structure ontologique »⁴ qui consiste en un mouvement d'entrée ou de sortie d'une intériorité vers l'extérieur, soit: « une opération primordiale de signification où l'exprimé n'existe pas à part de l'expression »⁴. Selon Collingwood, la qualité de l'œuvre tient dans le fait que l'émotion qu'elle produit soit à l'image de l'émotion générée par le motif original. Il distingue ainsi trois différents niveaux de représentation⁵: la représentation littérale, la représentation de ce qui est perçu en arrangeant, transformant et en ajoutant des éléments qui ne font pas directement partie du modèle, et enfin, la représentation émotionnelle d'une intériorité. Pour lui, l'artiste ne sait pas encore quelle expression se dégagera de son travail avant de l'avoir réalisé tandis que dans la représentation le résultat final est prévu à l'avance. L'expression concerne l'individuel et le subjectif, quand la représentation a à voir avec le général, l'universel. Par exemple, si un artiste souhaite exprimer de la tristesse, celle-ci sera une tristesse spécifique, propre à son état mental et au contexte extérieur qui l'a provoquée, et non le sentiment de tristesse de manière générale et absolue.

1.1.2 Remise en ordre, prise de conscience et clarification

D'après Collingwood, l'expression permet une clarification et une organisation de l'émotion. Avant l'acte d'expression, l'individu ressent une « perturbation »⁵, ou une « excitation »⁵ qui le pénètre. Il se sort de cet état en s'exprimant, c'est à ce moment qu'il prend conscience de cette émotion. Il dira ainsi: « C'est aussi quelque chose qui a à voir avec la conscience: l'émotion exprimée est une émotion de la nature de la personne qui sent qu'elle n'est plus inconsciente »⁶. Il en résulte une émotion « esthétique »⁵ qui n'est pas une « émotion préexistante à son expression »⁵. Elle provient de la concentration et la clarification de la personne qui exprime: « L'acte d'exprimer l'émotion est ainsi une

⁴ Dupond, Pascal. *Le vocabulaire de Merleau-Ponty*. Vocabulaire de. Paris: Ellipses, 2001.

⁵ Kemp, Gary. « Collingwood's Aesthetics ». In *The Stanford Encyclopedia of Philosophy*, édité par Edward N. Zalta, Fall 2016. Metaphysics Research Lab, Stanford University, 2016.
<https://plato.stanford.edu/archives/fall2016/entries/collingwood-aesthetics/>.

⁶ Collingwood, R. G. *The Principles of Art*. Read Books Ltd, 2016.

exploration de ses propres émotions »⁷. Ainsi, elle ne provient pas d'un processus de déduction mais d'une mise en forme. Pour lui, la création artistique ne doit pas dépendre d'un plan, contrairement par exemple à l'artisanat. L'œuvre d'art n'est pas productrice d'émotion délibérée : il s'agit d'une expression pour et par elle-même. Ainsi, c'est le processus de création qui possède une fin et non l'œuvre elle-même. L'émotion permet de structurer une expérience et de la synthétiser en reliant les éléments qui la constituent. Il faut préciser cependant qu'exprimer n'est pas extérioriser ou laisser sortir une émotion de manière incontrôlée. L'expression doit avant tout passer par l'élaboration d'un objet qui peut être par exemple une peinture, une danse, des notes de musique, qui vont servir à transmettre ce qui est exprimé. D'après Dewey, cela ne peut se faire sans que le sujet inhibe en premier lieu cette émotion pour pouvoir la structurer et lui donner une forme. Il faut ainsi qu'il effectue un travail sur les expériences passées.⁸

1.1.3 Le rôle de l'intuition

Dans son ouvrage *Esthétique comme science de l'expression et linguistique générale*, Benedetto Croce lie expression et intuition. Pour lui, l'intuition est une connaissance différente de la perception, de la sensation et du concept qui se réfèrent au général. L'intuition a besoin de l'expression pour être transmise, il dit ainsi : « Tout ce qui est véritablement intuition ou représentation est aussi expression »⁷. L'intuition représente ce qui survient avant tout jugement et raisonnement. Certains scientifiques comme par exemple Henry Poincaré, pensent également que l'intuition a une fonction antérieure aux autres modes de connaissance. Il dit d'ailleurs de l'intuition qu'elle est le « résultat d'une sensibilité esthétique subconsciente qui nous fait deviner des relations cachées »⁹. D'après Croce, l'expression est nécessaire à l'intuition car celle-ci « fait, forme, exprime »⁷. C'est ensuite par la production d'un objet physique qu'il est possible de communiquer cette intuition. Nous allons donc tenter de voir en quoi l'expression artistique, comme moyen de communication, peut être vue comme une forme de langage et ainsi déterminer la place du spectateur au sein de cette expérience.

⁷ Kobayashi, Chinatsu. « Bosanquet, Collingwood et l'esthétique idéaliste britannique ». *Philosophiques* 36, n° 1 (2009): 149-82. <https://doi.org/10.7202/038014ar>.

⁸ Quéré, Louis. *Le modèle esthétique de la communication de John Dewey*. Occasional Paper 19, 2014

⁹ Poincaré, Henry

1.2 L'expression artistique peut être vue comme une forme de langage

1.2.1 L'importance de l'appropriation et de l'interprétation dans le domaine de l'expression

Une des notions les plus importantes au sujet de l'expression est que celle-ci doit impliquer une communication. L'expression se base à la fois sur l'artiste auteur et à la fois sur la communication qui se crée entre lui et le public récepteur. Jenefer Robinson distingue les deux termes « expression » qui se réfère au travail de l'artiste et « expressivité » qui qualifie la part de réception du spectateur¹⁰. Lors d'un acte d'expression, l'individu qui vit et expérimente une émotion est amené à manifester celle-ci à autrui afin que cette émotion puisse être recueillie, assimilée et perçue par des individus extérieurs. On a d'abord vu que l'expression permettait à la personne qui exprime de remettre de l'ordre dans le chaos afin de prendre conscience et de reconnaître cette émotion. On peut alors voir l'art comme un langage: il élucide et dénoue une expérience en la rendant claire et consciente pour l'individu. Merleau-Ponty parle ainsi de l'expression comme une « appropriation »¹¹ de la pensée. Pour cela, fait une analogie entre le langage et la création artistique¹². La peinture se constitue d'un système de règles et de notions qui sont à la fois culturelles et techniques. De la même manière, la langue contient un ensemble de signes et de structures permettant aux individus de communiquer. Ainsi, chaque personne devra s'approprier la langue pour y faire naître une parole unique qui est sienne et exprimer sa pensée. Il en est de même pour le peintre qui va s'approprier la peinture afin de produire une œuvre personnelle et originale, celle-ci étant différente d'une simple imitation ou reproduction. C'est cette façon de s'approprier un médium qu'on appellera le style de l'artiste, l'interprétation jouant un rôle majeur dans cette appropriation.

¹⁰ Robinson, Jenefer. *Expression and Expressiveness in Art*, Postgraduate Journal of Aesthetics, Vol. 4, No. 2, August 2007

¹¹ Merleau-Ponty, Maurice. *Phénoménologie de la perception*. Editions Gallimard, 2013.

¹² Guilloux, Robin. « Maurice Merleau-Ponty, Art et Langage ». Le blog de Robin Guilloux. Consulté le 6 avril 2019. <http://lechaturmonepaule.over-blog.fr/2018/05/maurice-merleau-ponty-art-et-langage.html>.

1.2.2 L'utilisation des signes et des symboles pour exprimer



Illustration 1 - Vieil homme dans la tristesse - Vincent Van Gogh (1890)

Illustration 2 - New York movie - Edward Hopper (1939)

Illustration 3 - Jeune homme triste dans un train - Marcel Duchamp (1911)

Henri Matisse distingue deux manières d'« exprimer les choses »: « l'une est de les montrer brutalement, l'autre de les évoquer avec art »¹³. Par exemple, si un artiste souhaite exprimer de la peine ou de la tristesse, il devra user de différents raccourcis et de symboles permettant de la suggérer. On peut se référer à la manière avec laquelle Van Gogh, dans son tableau *Vieil homme dans la tristesse*, joue sur les distorsions et les déformations pour renforcer la posture prostrée du personnage et amplifier l'émotion de celui-ci, tandis que Marcel Duchamp avec *Jeune homme triste dans un train*, utilise la décomposition du mouvement ainsi que qu'une palette de couleur réduite pour décrire son attitude. Hopper se sert du vide et la disposition de l'espace pour créer une sensation d'isolement qui renforce l'état d'âme du personnage présent dans la scène dans *New York movie*. Ce n'est pas la même tristesse qui est exprimée dans ces trois œuvres, mais bien à chaque fois une forme de tristesse singulière, inextricablement liée à celui ou celle qui la ressent. Il apparaît alors une importance du subjectif dans l'expression artistique, celle-ci étant une forme de communication à part entière. Kandinsky voit également une analogie entre l'art et le langage. Il considère qu'il doit exister une forme logique en plus de l'intuition dans la création artistique qui sert à construire des règles¹⁴. Par exemple, la couleur a une place importante dans l'expression: elle permet d'engendrer un mouvement et des forces au sein du tableau. Les différents éléments de la composition se répondent, s'opposent, se lient

¹³ *Ecrits et propos sur l'art* (1972), Notes d'un peintre, la Grande Revue, 25 décembre 1908.

¹⁴ Kandinsky, Wassily. *Du spirituel dans l'art et dans la peinture en particulier*. Éditions de Beaune, 1951.

comme un ensemble organisé. Ce macrocosme constitué d'interactions génère différentes impressions sur le public. Elle fonctionne par association: elle rappelle des éléments du réel, par exemple le rouge évoquera la chaleur ou la passion par son lien symbolique avec le feu.



Illustration 4 - Jaune - rouge - bleu, Vassily Kandinsky (1925)

1.2.3 La particularité de l'expression artistique comme moyen de communication

John Dewey considère l'art comme étant la forme de communication la plus puissante. Il permet d'aller au delà de ce qui est possible par le langage pour partager et transmettre une vision et des idées. On dit alors de l'art qu'il est extatique: il révèle ce que la raison et les sens seuls ne peuvent montrer. Cette communication se base sur l'émotion et sur le médium, il dira ainsi que « La science a des significations, l'art les exprime »¹⁵. L'œuvre d'art est l'objet qui donne la possibilité de mettre en lien un destinataire et un émetteur. Elle attend une action des deux parties qui se traduit sous la forme d'une coopération en vue d'une conséquence entre celui qui exprime un message et celui qui le reçoit et l'interprète. Nous allons ainsi voir en quoi l'expression artistique permet de constituer une expérience pour l'artiste et le spectateur ainsi que la place du corps et de l'objet physique dans celle-ci.⁸

¹⁵ « Data Art : une œuvre montre comment s'agite votre cerveau... et c'est beau ! » L'ADN, 20 avril 2018. Consulté le 29 avril 2019. <https://www.ladn.eu/mondes-creatifs/traduire-des-souvenirs-en-oeuvres-dart/>.

1.3 L'importance de l'expérience et la place du corps comme interface dans le processus d'expression artistique

1.3.1 Corps, sens et motricité

D'après Merleau-Ponty, le corps joue un rôle primordial dans l'expression artistique puisqu'il consiste en une interface par laquelle les perceptions permettent au sujet d'appréhender et d'expérimenter le monde sensible. C'est par et à travers lui que l'on accède aux impressions originaires du monde extérieur. Le corps est une « manière d'être au monde »¹⁶. On retrouve une forme d'inversion de statut entre le peintre et ce qui est perçu par le peintre. En effet, à la manière d'une surface sensible, celui-ci se laisse imprégner par le monde réel, et c'est de là qu'en ressort une expression. Il dira « le peintre reprend et convertit justement en objet visible ce qui sans lui reste enfermé dans la vie séparée de chaque conscience: la vibration des apparences qui est le berceau des choses »¹⁷. On peut prendre en exemple la peinture impressionniste par laquelle les artistes disent peindre ce qu'ils perçoivent, c'est à dire les apparences de la réalité sensible. L'artiste impressionniste traduit ce qu'il voit en une matière visible, il y a donc un lien très important entre l'artiste et les sens, notamment les sensations kinesthésiques et les sensations motrices. Collingwood parle de « sensas »⁵: informations provenant d'une expérience sensorielle, que l'on pourrait traduire par impressions. Il oppose ces sensas aux pensées qui font appel à la logique et à la dualité telles que le vrai et le faux ou le bien et le mal. Au contraire, les sensas « ne peuvent se contredire » car ils possèdent tous une propriété émotionnelle qui permet une expérience particulière en fonction d'un contexte donné. Ces sensas permettent d'engendrer de nouvelles idées qui imprègnent l'esprit de celui qui ressent. Une fois la sensation devenue consciente, celle-ci se transforme en une forme d'imagination et il est possible de reproduire ses effets à la demande: « L'imagination est donc la nouvelle forme que prend le sentiment lorsqu'il est transformé par l'activité de la conscience »⁵.

¹⁶ Hubeny, Alexandre. « Le sens de l'histoire dans la philosophie de Merleau-Ponty ». *Annales de l'École pratique des hautes études* 117, n° 113 (2004): 415-20. <https://doi.org/10.3406/ephe.2004.12396>.

¹⁷ Pons, Gilbert, et Maurice Merleau-Ponty. *Le doute de Cézanne : Extrait*. Rochefort: Les petites allées, 2016.

1.3.2 L'œuvre d'art comme résultat d'une action sur la matière

Lors d'un acte d'expression artistique, un « matériau intérieur »⁸ en interaction avec son environnement extérieur engendre l'expression. Ce matériau consiste en un ensemble de dispositions et d'émotions liées à cet environnement. Au même moment, un matériau extérieur va devenir le véhicule de l'émotion: il est modelé, transformé, organisé. John Dewey dira que ces deux matériaux sont « en relation organique l'un avec l'autre »⁸. Il accorde une importance à ce qu'il appelle l'objet expressif. Il s'agit de créer un lien entre général et individuel. Pour Kandinsky, l'expression consiste à passer de l'émotion immatérielle à l'œuvre qui s'inscrit dans la matière. L'enjeu est de retrouver une harmonie entre le fond et la forme. De la même manière, Collingwood dira que l'artiste manifeste des émotions au moyen de la réalisation d'un objet ou « artefact »⁵. L'art amène donc à « appréhender et exprimer une émotion considérée comme un motif pour l'action »¹⁸. Il considère que l'œuvre d'art est d'abord une chose issue de l'imagination, présente dans l'esprit de l'artiste, et que la création permet à la chose de s'incarner dans la matière et d'être partagée à un public. Il y a donc une dimension imaginative: la réalisation d'une œuvre est en fait une « instance »⁵ de la création imaginée. L'expression artistique se caractérise par l'existence d'un processus de création et de réalisation, elle appartient donc au domaine du faire et de l'agir et son résultat s'inscrit ainsi dans un objet physique matériel. Lorsqu'un support est employé à des fins artistiques, celui-ci devient médium de l'œuvre. Dewey expose ainsi que: « Ce qui fait d'un matériau un médium, c'est qu'il est utilisé pour exprimer une signification différente de ce qu'il est du seul fait de son existence physique : la signification, non pas de ce qu'il est physiquement, mais de ce qu'il exprime »¹⁹. Il sera ainsi important dans la deuxième partie de se préoccuper de la spécificité du médium numérique et de la manière avec laquelle on peut l'exploiter à des fins expressives.

1.3.3 L'expression comme expérience et rencontre entre l'œuvre et le public

L'œuvre d'art ne consiste cependant non pas en l'objet incarné par l'œuvre mais en la réalisation et la rencontre avec celle-ci. Elle permet de lier les individus: l'état d'âme de l'artiste ainsi partagé contamine le public. Tolstoï dira ainsi: « L'art est une activité humaine

¹⁸ Collingwood, R. G. *The Principles of History: And Other Writings in Philosophy of History*. Édité par W. H. Dray et W. J. van der Dussen. Oxford: Clarendon Press, 2001.

¹⁹ Dewey, John. *L'art comme expérience*. Folio, 2010.

consistant à ce qu'un homme, consciemment, au moyen de certains signes extérieurs, transmette à d'autres des sentiments qu'il a vécus et que ces autres personnes soient infectées par ces sentiments et les éprouvent également »²⁰. Il est donc important de considérer la rencontre avec une œuvre d'art non pas comme une simple contemplation mais également comme une expérience. En effet, c'est l'expérience qui permet d'élaborer puis d'apprécier une œuvre. John Dewey reprend ainsi le concept d'« expérience pure »²¹ de William James pour décrire l'expérience esthétique car elle se suffit à elle-même et n'a pas d'autre fin qu'elle-même. L'expression artistique provient d'une expérience qui est en plus intimement liée à la perception: on voit avec le tableau, en fonction de lui. Le spectateur est nécessairement lié à l'existence de l'œuvre car il doit aller à la rencontre de celle-ci pour qu'elle puisse exister. L'artiste est également observateur lorsqu'il réalise son œuvre car il interagit continuellement avec celle-ci lors de son exécution. Chaque spectateur qui expérimente l'œuvre devra faire appel à sa propre imagination pour reconstituer la sensation et l'émotion qui s'en dégage. Celle-ci est profondément liée au processus de création. De plus, l'œuvre évolue perpétuellement puisqu'elle ne sera jamais expérimentée de la même manière en fonction du lieu et de l'époque durant laquelle elle sera exposée. Ce qu'exprime l'œuvre est en partie dépendant du contexte et son impact change selon le lieu et la culture du spectateur qui l'éprouve. La relation entre l'œuvre et le spectateur joue ainsi un rôle dans ce qu'elle exprime.

On a donc vu que l'expression artistique est un moyen de communication particulier, qui consiste en une clarification et une mise en ordre d'émotions dans le but de les transmettre au monde extérieur. Cette transmission s'effectue par le biais de l'appropriation des codes et d'un médium spécifique afin d'inscrire matériellement la vision subjective de l'artiste dans une œuvre d'art, son expressivité dépendant ensuite de la réception du public. L'œuvre peut être vue comme une forme d'expérience sensible destinée à un public qui doit aller à la rencontre de celle-ci afin de pouvoir l'éprouver. Avant de nous intéresser à l'art numérique, nous allons à d'abord nous baser sur plusieurs mouvements artistiques utilisant les médias traditionnels pour tenter de déterminer les différents moyens par lesquels les

²⁰ Tolstoy, graf Leo. *What Is Art?* Crowell, 1899.

²¹ Deledalle, Gérard. « II. William James (1842-1910). L'expérience pure ». *Le Point philosophique* 3e éd. (1998): 133-45.

artistes parviennent à communiquer une émotion ou une vision du monde et génèrent ainsi une expressivité au sein de leurs œuvres.

2. Générer une œuvre expressive à l'aide des médias traditionnels et numériques

Nous allons à présent tenter de déterminer les différents moyens utilisés par les artistes afin de réaliser des œuvres expressives. Pour cela, nous commencerons par nous intéresser à la manière dont l'expression a pris de l'importance dans l'histoire de l'art, notamment au tournant du XVIII^e siècle. Nous étudierons les procédés utilisés par des artistes traditionnels de plusieurs mouvements, puis nous nous intéresserons à la problématique de l'art numérique et de ses spécificités.

2.1 L'expressivité dans l'art traditionnel

2.1.1 Contexte historique de l'expressivité dans l'histoire art et émergence de l'individu artiste

La place de l'expression en art s'est fortement accentuée durant la révolution romantique, vers la fin du XVIII^e siècle. A cette époque, suite à la montée de la bourgeoisie, on a pu assister à une libéralisation du marché de l'art. L'individualité et le désir d'avoir une existence propre et singulière a permis d'offrir une place à la subjectivité de l'artiste. Celui-ci acquiert une liberté mais aussi une précarisation. On parlera ainsi d'art autonome: l'œuvre d'art se suffit à elle-même et adopte désormais un rôle exclusivement esthétique. Elle est considérée comme singulière, l'originalité et l'invention étant privilégiées par rapport à l'imitation ou la mimésis. L'œuvre ne dépendant plus de commanditaires, on constate également une dimension de subversion de l'autorité car l'artiste ne doit plus obéir à une académie mais développer sa propre manière de peindre ou de sculpter, ce qu'on appellera le style. Ces évolutions permettent ainsi l'émergence nouveaux modes de création artistique. Les conceptions sur l'art ayant évolué vers une vision plus personnelle et singulière de l'œuvre d'art tout en offrant une place privilégiée à l'expression de soi, on peut constater l'apparition de nouveaux styles qui impliquent une manière inédite et subversive d'employer la peinture.

2.1.2 La couleur comme puissance expressive

On va alors assister à une remise en question du statut de la couleur. Auparavant, celle-ci était considérée comme secondaire. Avec notamment le *prima del disegno*, la peinture était d'abord vue comme un art du dessin. Jacqueline Lichtenstein, dans *La Couleur éloquente : rhétorique et peinture à l'âge classique*²², explique ainsi que la couleur était condamnée en raison de sa matérialité et de son caractère éphémère et superficiel. On la considérait comme trompeuse car faisant partie du domaine de l'apparence et de la séduction nous éloignant de la vérité. Elle était également liée aux arts mécaniques et à l'artisanat tels que la teinturerie et donc vue comme non intellectuelle. C'est à partir du XVIIIe siècle que son statut change. Le fait de concevoir l'œuvre comme l'expression d'une intériorité va en partie mener à une revalorisation de la couleur. Les artistes vont dès lors s'attarder à représenter la nature comme une entité vivante et en perpétuel changement. La couleur est ainsi utilisée comme moyen d'exprimer une forme de dynamique, de mouvement. L'aspect inachevé d'une œuvre et l'esquisse sont valorisés par rapport au projet fini. On souhaite ainsi se rapprocher de l'immédiateté. La couleur, par son aspect vif et tumultueux, offre ainsi de nombreuses possibilités expressives.

Elle revêt une fonction symbolique car elle n'évoque pas la même chose selon le lieu et l'époque. Pour illustrer ceci, Michel Pastoureau nous donne l'exemple de la mer qui était représentée un certain nombre de couleurs différentes durant l'antiquité²³ (rouge, blanc, noir...), pour ensuite être symbolisée par la couleur verte en occident au Moyen-âge. Cette couleur est aujourd'hui finalement remplacée par le bleu. Il y a également une intellectualisation de celle-ci avec la découverte de certaines théories scientifiques et optiques comme la théorie du contraste simultané des couleurs énoncée par Michel-Eugène Chevreul en 1839. La couleur permet enfin de rapprocher l'œuvre du spectateur en lui accordant plus d'importance. Kandinsky distingue deux effets lorsque celui-ci est confronté à la couleur d'une œuvre d'art. Le premier est « purement physique »¹⁴, il correspond à la stimulation et à la satisfaction momentanées que les couleurs peuvent provoquer sur notre corps via nos sens. Il peut mener à un second effet, qu'il qualifie de « psychique »¹⁴, responsable de la « vibration de l'âme »¹⁴ lorsque celle-ci est parvenue à toucher l'esprit du

²² Lichtenstein, Jacqueline. *La couleur éloquente : Rhétorique et peinture à l'âge classique*, Nouvelle édition, 2013

²³ Pastoureau, Michel, et Dominique Simonnet. *Le Petit livre des couleurs*. Paris: Points, 2014.

spectateur. Cézanne parle également de « sensations optiques »¹⁴. Dans sa peinture, les objets se définissent ainsi grâce à la couleur et non plus aux contours. Celle-ci joue un rôle important dans ses toiles car c'est elle qui exprime la profondeur et qui sert de structure au tableau, rôle autrefois attribué au dessin. Notre intérêt se porte particulièrement sur l'expressivité engendrée par l'emploi de la couleur et la manière dont celle-ci est employée dans différents mouvements artistiques.



Illustration 5 - Impression, soleil levant - Claude Monet (1872)

Par exemple, les impressionnistes réalisent leurs tableaux en se basant sur la théorie de Chevreul. Elle va leur permettre selon Bernard Dorivalls d' « exalter les tons »²⁴ grâce aux complémentaires. Ils ne mélangent plus les couleurs mais les juxtaposent sur la toile et jouent sur leurs connaissances en optique pour produire l'effet souhaité chez le spectateur. Bien qu'ils saisissent la lumière à partir d'une scène réelle, la couleur n'est pas utilisée de manière réaliste. Elle permet une recomposition de la nature en ne servant plus uniquement à identifier en se référant à la couleur réelle émise par l'objet, mais aussi à exprimer une vision et une impression. La lumière a généralement pour rôle de donner une intention ou de créer l'ambiance ou le sentiment global de la scène. Ici, travail de l'ombre et de la lumière se fait par la couleur.

²⁴ Alliez, Éric. *L'oeil-cerveau: nouvelles histoires de la peinture moderne*. Vrin, 2007.

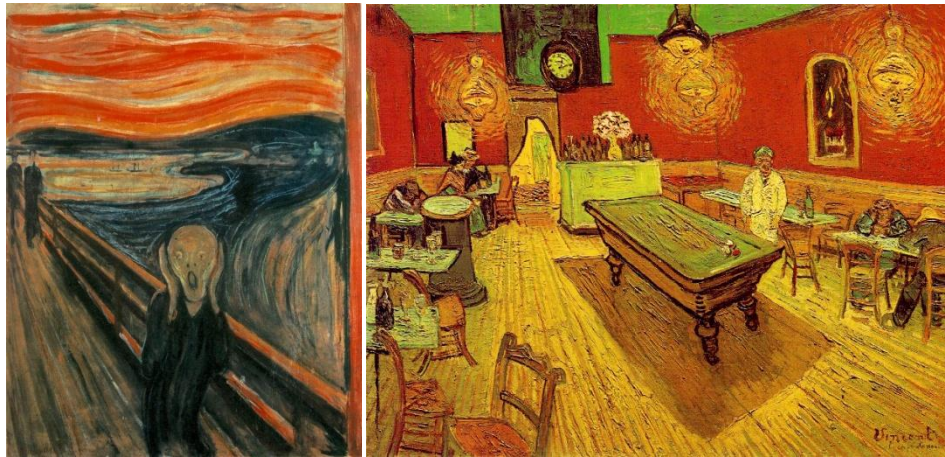


Illustration 6 - Le Cri - Edvard Munch (1893-1917)

Illustration 7 - Le Café de nuit - Vincent Van Gogh (1888)

Dans l'expressionnisme, la couleur sert à exprimer les tourments des artistes, les tableaux traduisant généralement certains états psychologiques tels que l'angoisse, la peur et le trouble. Par exemple, dans *le Cri* d'Edvard Munch, la couleur apporte un caractère au tableau: les touches rouge orangées ardentes et presque agressives ajoutent une tension dans l'image. On peut également relever la façon avec laquelle les couleurs sont posées sur la surface du tableau: il peut s'agir de petites touches, de courbes, d'aplats. Ici, le contraste et la discordance des couleurs ajoutés aux ondulations induisent une déstabilisation chez le spectateur et communiquent ainsi un sentiment d'inquiétude et d'effroi. L'application des couleurs en lignes obliques et courbes sert à indiquer la perspective de la scène qui semble nous inclure dans le tableau et qui trouble l'équilibre. Il y a en plus une dimension symbolique dans l'utilisation de la couleur, le tableau semblant se diviser en deux mondes différents: l'un est plus réaliste et se constitue de couleurs froides, sombres et de lignes droites. Il semble renvoyer au monde organisé et civilisé. Le second est représenté dans les tons chauds et criants appliquées par traînées ondulantes qui serpentent à droite et en haut du personnage. Ces éléments peuvent à la fois évoquer la nature et à sa puissance menaçante, mais aussi le monde du rêve et de la folie. On retrouve ce type de procédé chez Van Gogh, qui dira dans une lettre en parlant de son tableau *Le Café de nuit*: « j'ai cherché à exprimer l'idée que le café est un endroit où l'on peut se ruiner, devenir fou, commettre des crimes. Alors j'ai cherché, par des contrastes de rose tendre et de rouge sang, de doux vert Louis XV et Véronèse, contrastant avec les jaune et les vert-bleu durs, tout cela dans une

atmosphère de fournaise infernale, de soufre pâle, à exprimer comme la puissance des ténèbres d'un assommoir »²⁵.

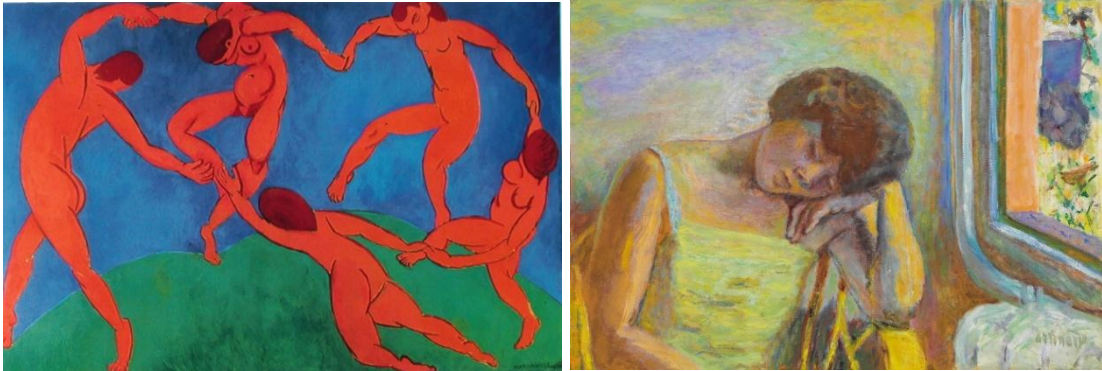


Illustration 8 - La Danse - Henri Matisse (1910)
Illustration 9 - Femme endormie - Pierre Bonnard (1928)

Chez les artistes fauves comme Henry Matisse, on observe une simplification passant par des aplats de couleur. Celle-ci devient petit à petit plus importante que le dessin. Son application permet de créer une dynamique au sein du tableau. Les artistes ouvrent ainsi une porte vers l'abstraction. Ce mouvement se caractérise par l'emploi de couleurs vives et pures car très peu mélangées. Dans les œuvres de Bonnard également, des amas de couleurs emplissent l'espace de la toile. Il peint généralement d'après souvenir et note ses intentions et ses ressentis dans un carnet puis dit prendre le temps de songer avant de réaliser ses œuvres. Les couleurs vives et contrastées génèrent une atmosphère flamboyante et produisent une sensation d'exaltation. Il peint l'intimité tout en donnant sa propre vision du quotidien par le travail du coloris. La subjectivité de l'artiste transparaît ainsi dans le choix des couleurs, il affirme notamment qu'il souhaite: « Qu'on sente que le peintre était là, voyait consciemment les objets dans leur lumière déjà conçue dès le début »²⁶.

²⁵ Van Gogh, Vincent. Lettres de à Théo Van Gogh, des 8 et 9 septembre 1888.

²⁶ Bonnard, Pierre, Alain Lévêque, et Antoine Terrasse. Observations sur la peinture. Strasbourg: L'Atelier Contemporain, 2015.



Illustration 10 - Around the blues - Sam Francis (1957, 1962)

Enfin, dans le domaine de la peinture abstraite, on peut évoquer le mouvement d'abstraction lyrique des années 1940. Cette forme de peinture s'oppose à l'abstraction constructiviste ou géométrique en ayant pour enjeux l'expression d'une réalité intérieure ainsi que l'exaltation d'une sensualité. En effet, les artistes de ce mouvement voulaient rendre la peinture abstraite moins froide et impersonnelle en employant la couleur afin d'apporter plus de lyrisme et de transmettre des émotions. Il en ressort une forme d'harmonie et d'effervescence pouvant être ressentie par le spectateur. Le corps prend une place importante, certains de ces artistes emploieront notamment l'action painting, ou la peinture gestuelle comme Jackson Pollock. Par exemple dans *Around the Blues* de Sam Francis, œuvre s'apparentant au tachisme, des projections et des coulures peinture colorée produisent un effet vibrant et organique. Ce sont elles qui, assemblées par couches, constituent l'espace de la toile et enivrent le spectateur.

2.1.3 La simplification et la stylisation



Illustration 11 - Sans titre - Yukio Karaki

D'autres paramètres permettent de rendre une œuvre plus expressive. On peut ainsi évoquer la simplification des formes et la stylisation. Toutes deux permettent d'amplifier ou de mettre en valeur certains éléments pour y apporter une signification et donner à voir. L'artiste va ainsi sélectionner certaines dimensions du réel et les représenter d'une manière atypique, en ayant généralement un but démonstratif ou expressif. Par exemple dans les tableaux de Yukio Karkai, un artiste japonais de l'art brut survivant des bombardements nucléaires survenus à Hiroshima et Nagasaki, on peut observer l'emploi de raccourcis et d'emphases, certains éléments étant par exemple représentés plus grands sans tenir compte de la perspective. Il y a une certaine spontanéité dans sa manière de représenter des scènes vécues et il tente d'aller à l'essentiel. L'art brut se caractérise par l'utilisation de l'art à des fins exclusivement expressives sans réel but d'atteindre une quelconque supériorité technique ou artistique. Ses œuvres ont un rôle de témoignage: l'enjeu n'est pas de peindre une scène objectivement réaliste à l'aide de moyens techniques avancés mais au contraire de laisser transparaître la vision de l'auteur. Son graphisme unique et les imperfections du dessin apportent de la subjectivité et donnent à voir son expérience personnelle.



Illustration 12 - Portrait d'un homme - Erich Heckel (1919)

Dans l'expressionnisme, on a également des déformations, une forme de stylisation et de simplification. Elles permettent de renforcer une idée ou une émotion par exemple par exagération. C'est un procédé récurrent comme on peut le voir par exemple dans le tableau d'Erich Heckel, *Portrait d'un homme*. De la même manière, dans *le Cri* de Munch, le visage du personnage au premier plan est représenté avec une expression de peur exacerbée. Ses proportions ne sont pas réalistes. L'ensemble des éléments du tableaux est simplifié et a une

apparence floue, les personnages ne sont pas représentés de manière distincte et il n'y a pas d'évolution de la netteté selon la profondeur. L'artiste souhaite par cela nous plonger dans un état de confusion. Ainsi, l'expressivité provient également d'une part d'aléatoire, d'erreurs et d'imprécisions avec souvent la non obéissance à des règles strictes de composition ou de la perspective.

2.1.4 L'emploi du matériau, l'importance de la matière et du médium

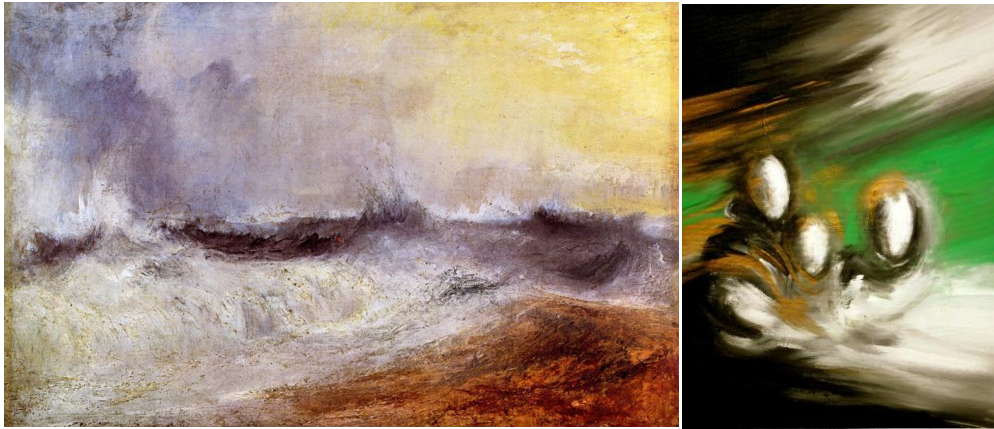


Illustration 13 - Waves breaking against the wind - Joseph Turner (1835)

Illustration 14 - Tableau de Corinne Lamoine

Le mot matière vient du latin « materia » qui désignait autrefois le « bois », un matériau perçu comme une substance qui engendre. En art, la matière peut se référer au support ou au matériau, le support étant lui un « subjectile ». Le matériau est la matière qui est déformée, modifiée par l'artiste durant la réalisation d'une œuvre et qui la constitue. La façon dont l'artiste travaille la matière peut générer différents effets. Par exemple, certains peintres aiment laisser apparaître la trace de la brosse, ce que les impressionnistes appellent la « virgule ». D'autres jouent sur l'altération, l'empâtement ou le mélange. Ils peuvent aussi se servir un rapport d'analogie entre certains aspects de la peinture et la matière réelle de l'objet représenté. C'est le cas par exemple avec l'aspect flottant et nébuleux que donne la peinture dans le tableau *Waves breaking against the wind* de Joseph Turner, qui évoque le caractère turbulent, spumeux et léger de l'écume. La manière avec laquelle les matériaux sont appliqués sur la toile font partie du mode d'expression des artistes. Cet aspect est particulièrement visible dans tableaux d'Iris Scott, une artiste qui peint avec ses doigts: toute

l'énergie qui se dégage de ses œuvres passe par la direction des touches de peinture ainsi que les couleurs appliquées. Le corps revêt ici un rôle fondamental.



Illustration 15 - Peinture au doigt - Iris Scott

C'est ce qu'on voit dans l'expressionnisme moderne: les artistes utilisent le corps comme sujet d'expérimentation. Pierrette Cornu dira qu'elle cherche à « mettre en scène la matière »²⁷: celle-ci permet d'apporter une dimension organique, presque vivante. Dans les œuvres de Corinne Lamoine, on peut sentir des vibrations des couleurs sur le fond sombre, une forme de dynamisme passe par des traînées de peinture. Celles-ci donnent un aspect tangible et mouvant à l'œuvre. Les artistes effectuent un travail de déconstruction : « Chaque dessin, chaque peinture, chaque volume réalisé est une pièce minuscule d'un puzzle - morceaux de vie, vie en morceaux - qui ne serait jamais terminé, un morceau de l'infini d'un univers sans limites comme le cosmos ou l'empire des rêves »²⁷ dira Corinne Le Guhenec. Cela permet de laisser transparaître un intérieur, une réalité sous-jacente.



Illustration 16 - Walls Bear the Scars of War - Antoni Tàpies

²⁷ Campo, Antoine. *L'Expressionnisme d'hier à aujourd'hui - l'expressionnisme contemporain*.

Pour aller encore plus loin, on peut prendre pour exemple le matiérisme, courant pictural informel apparu à la fin des années 1940 en Europe. Il consiste en l'expérimentation et l'utilisation de matériaux non conventionnels tels que le verre, le plâtre, le sable... L'artiste exploite la nature brute de ces matériaux à des fins expressives. Une certaine importance est accordée à la trace laissée par l'auteur de l'œuvre: elle peut être entaillée, raclée, grattée, déchirée, elle se compose généralement de plusieurs couches de matière. Ces artistes souhaitaient inscrire l'art abstrait dans une forme de matérialité. Antoni Tàpis par exemple emploie la matière pour exprimer la souffrance et représenter l'horreur de la guerre. Dans *Walls bear the sky of war*, les traces de griffures évoquent une forme de lutte et semble symboliser les empreintes laissées par les hommes qui forgent la mémoire collective.

On notera enfin l'importance du choix du médium et de la technique dans ce que l'artiste souhaite exprimer. Mc Muhan dira dans *Understanding Media: The extensions of man* : « le médium, c'est le message »²⁸. Le message n'est pas un simple contenu et il dépend directement de la manière avec laquelle il est transmis et donc de son support. Il n'est pas possible de le séparer du médium car celui-ci façonne notre rapport à la société. En art, le choix du médium a un impact direct sur le résultat et l'effet produit par une œuvre. Il agit comme une extension de nos sens en faisant passer l'information. Le support va par exemple définir un format, une texture, une consistance, qui vont influencer la manière avec laquelle l'artiste compose l'image, la façon dont s'étalent, se posent ou s'appliquent les matériaux. L'artiste Auguste Macke le disait ainsi dans son essai *Les Masques* dit à ce propos: « les différences d'expression viennent du matériau, du mot, de la couleur, du son, de la pierre, du bois, du métal »²⁹. Le médium est donc directement responsable de l'aspect final de l'œuvre.

Nous avons ainsi effectué une étude non exhaustive des divers moyens expressifs offerts par l'art traditionnel et de leur utilisation dans l'histoire de l'art: l'expression peut ainsi passer par le biais du travail de la couleur, de la lumière, de la matière mais également par la composition, la déformation l'exagération et la stylisation. Il va falloir à présent

²⁸ McLuhan, Marshall, et Lewis H. Lapham. *Understanding Media – The Extensions of Man*. Reprint. Cambridge, Mass: MIT Press, 1994.

²⁹ Macke, August. *Les masques*. S.l.: Mélanges Editions, 2005.

étudier les caractéristiques du médium numérique dans le but d'élaborer au mieux nos outils en respectant les spécificités de ces techniques.

2.2 Les spécificités du médium numérique

L'art numérique, apparu au cours des années 1960, englobe toute création artistique reposant sur l'utilisation des ordinateurs et du langage informatique. Le processus de création se retrouve ainsi profondément modifié. Quelles possibilités supplémentaires offre le virtuel? Comment traiter la matière digitale afin d'engendrer de nouvelles formes d'expression? On va ainsi voir que les techniques numériques, notamment dans le domaine de l'image de synthèse, comportent à la fois des atouts et des inconvénients concernant les possibilités expressives offertes aux artistes.

2.2.1 Simulation et transposition

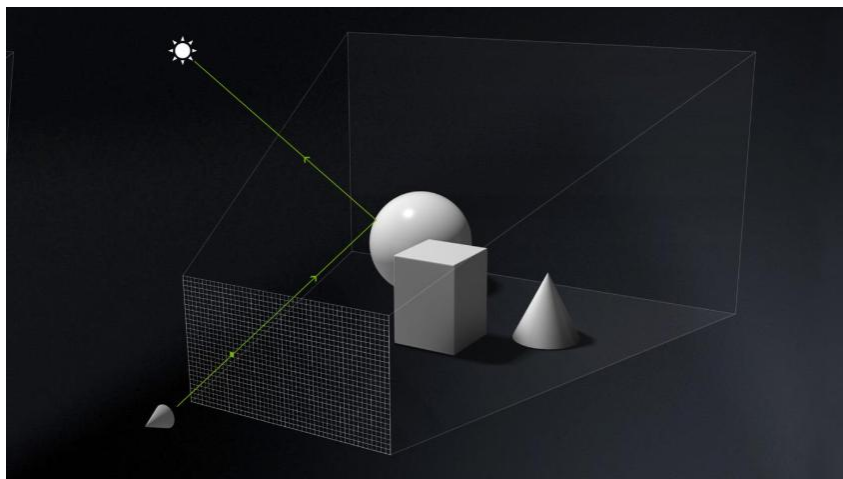


Illustration 17 - Schéma représentant l'étape de raytracing

Contrairement aux techniques traditionnelles, l'outil numérique se base sur un langage de programmation qui traite des données quantifiées et dépend d'un certain formalisme. Il permet de recréer des systèmes organisés en s'inspirant ou non de la nature. On retrouve ainsi ce lien privilégié entre l'art et la science. Dans le domaine de la 3D, l'espace est simulé à partir de données et de lois généralement basées sur des phénomènes physiques. La simulation obéit à des règles définies afin de reproduire des processus similaires à ceux qui engendrent les phénomènes naturels. Par exemple, les moteurs de

rendus permettent d'effectuer des calculs de lumière en créant virtuellement son comportement lorsque celle-ci se diffuse et rebondit sur des surfaces aux propriétés physiques diverses. On peut ainsi s'interroger sur la manière avec laquelle il est possible de produire des œuvres d'art à partir de modèles basés sur la rationalisation pure et comment retrouver une forme de sensibilité exploitable par les artistes. L'outil numérique peut, comme on le verra par la suite, simuler des propriétés d'autres médiums. Il serait intéressant d'exploiter cette dimension et de s'inspirer du fonctionnement de certaines techniques traditionnelles pour en tirer l'aspect expressif et la mettre à la disposition des artistes. Le recours au numérique pourrait ainsi nous faire tendre vers une forme de transposition qui consiste en l'utilisation de procédés, ici informatiques, afin de faire transparaître certains éléments stylistiques d'autres formes d'art tels que le dessin ou la peinture, l'enjeu n'étant pas ici d'imiter ou de reproduire à l'identique les médiums traditionnels mais d'en tirer la force expressive pour produire de nouveaux modes de création artistique.

2.2.2 L'automatisation, un gain de temps pour les artistes

Un des principaux avantages du numérique, notamment dans le domaine du film d'animation, est l'automatisation. L'utilisation des ordinateurs permet de calculer et d'effectuer d'un grand nombre d'opérations complexes, ce qui constitue un gain de temps considérable et permet d'obtenir des résultats qui ne seraient pas calculables à l'échelle humaine. Dans le domaine artistique, il évite ainsi un travail manuel qui peut être long, répétitif et fastidieux pour que les graphistes puissent se consacrer à des tâches plus artistiques requérant des compétences humaines. On laisse ainsi beaucoup plus la main à la machine qui obéit à un certain nombre de règles déterminées par l'utilisateur. Le point négatif de ce procédé est qu'il peut arriver que l'on perde le contrôle de celle-ci et que l'on n'atteigne pas le résultat escompté. De plus, l'automatisation mène parfois à une perte de sensibilité dans le sens où l'on perd la trace de l'artiste en se retrouvant par exemple avec des rendus ou animations trop parfaits, trop lisses voir totalement impersonnels. L'impact esthétique se retrouve ainsi diminué. L'utilisation de l'aléatoire peut permettre d'ajouter de l'erreur et de briser le déterminisme du programme informatique en apportant par exemple plus de matière et d'organique à celle-ci. On pourra voir cela par exemple avec l'emploi de fonctions aléatoires telles que les bruits ou noises en seconde partie.

2.2.3 L'essor de la génération procédurale comme terrain d'expérimentation et d'hybridation

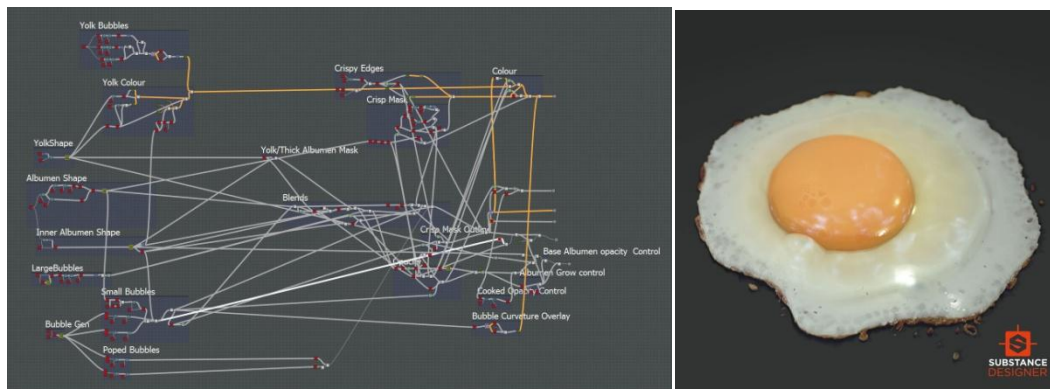


Illustration 18 - Oeuf généré procéduralement dans le logiciel de texturing Substance Designer

Un autre intérêt dans l'emploi des techniques numériques repose dans ses capacités d'expérimentation. Edmond Couchot et Norbert Hillaire disent d'ailleurs de l'art numérique qu'il aurait « une vocation expérimentale »³⁰. Le développement d'algorithmes et de logiciels permettant d'effectuer certaines étapes de manière procédurale, en passant notamment par des systèmes de graphes, laisse l'utilisateur tester rapidement un grand nombre de possibilités de rendu et d'animation. Il peut passer de l'un à l'autre facilement pour comparer ses résultats. On a également la possibilité de revenir en arrière, de conserver simultanément plusieurs versions différentes d'un même travail, de les assembler... Cela modifie considérablement le processus de création puisque dans l'art traditionnel, l'ajout et la transformation du matériau se fait généralement par couches de manière linéaire et continue, il n'y a pas vraiment de modularité. L'artiste acquiert donc de nouvelles capacités expressives. Il est également possible de combiner ou d'hybrider certains effets et de créer ses propres systèmes et algorithmes à des fins esthétiques. Avec l'avènement de l'art génératif, les artistes réalisent leurs œuvres de manière indirecte: ils établissent différentes règles en fonction du résultat souhaité et ces règles seront ensuite appliquées via un algorithme. C'est une manière différente de réaliser des projets artistiques, il peut arriver que le résultat soit différent de ce à quoi la personne s'attendait. On peut ainsi

³⁰ Couchot, Edmond, et Norbert Hillaire. *L'art numérique*. Paris: Flammarion, 2009.

qu'une place est accordée à l'aspect recherche, l'artiste pouvant parfois se qualifier d'artiste-chercheur. Il faut cependant constater que la méthode procédurale, par opposition au procédé manuel, peut en partie effacer et diminuer l'importance du geste de l'artiste.

2.2.4 La question de la matérialité du numérique et de la technicité

Il se pose ainsi la question de la matérialité des œuvres numériques. L'artiste travaille une matière "virtuelle" qui passe souvent inaperçue au regard des spectateurs. L'œuvre est en effet constituée d'un ensemble de données, généralement contenues dans des pixels, qui ont besoin d'un support matériel comme un écran ou un projecteur pour être rendues perceptibles pour le public. Elles sont donc reproductibles et ubiquitaires car elles peuvent être présentes dans un nombre indéfini de lieux physiques au même moment. Elles dépendent du support physique qui les rendent visibles et sont donc contraintes par l'obsolescence des machines. De plus, il existe un écart parfois important entre les désirs de l'artiste utilisateur et les possibilités qu'offrent les programmes ou logiciels de création graphique qui peuvent limiter son champ d'action. En effet, le travail de la matière numérique nécessite souvent des connaissances de programmeur. Une forme d'opacité s'élève ainsi entre l'artiste et la machine. En art traditionnel, au contraire, l'artiste est plus proche de la matière qu'il manipule et a un lien plus direct avec celle-ci. Bien que des compétences techniques soient aussi nécessaires, il a la possibilité d'agir sur la matière sans intermédiaires autres que son corps et ses instruments. Avec les outils numériques, le nombre d'intermédiaires augmente. Michel Guérain dira à ce sujet qu'« avec le fonctionnement des appareils techniques et, plus encore, avec les incontournables technologies numériques contemporaines, les nouveaux médiums ont vu s'épaissir les couches de médialité, s'accroître la part déléguée à des programmes qui « marchent tout seuls », s'imposer les rapprochements induits par l'arborescence des outils numériques et la prolifération des *écrans* ». ³¹ Si l'artiste souhaite créer un effet trop spécifique qui n'est pas réalisable avec les programmes existants, un travail de développement de fond peut être requis pour fabriquer de nouveaux outils dédiés à cet effet. Ce travail ne relève plus vraiment du domaine de l'artiste. A l'inverse, il peut arriver que l'outil propose un nombre trop élevé d'options, ce qui risque de perdre et d'embrouiller l'utilisateur. Il faudra ainsi

³¹ Guérain, Michel. « Qu'est-ce qu'un médium artistique ?. Intention et condition ». *Appareil*, n° 17 (29 juin 2016). <https://doi.org/10.4000/appareil.2308>.

trouver un juste milieu afin d'offrir un maximum de libertés aux artistes. Le but est qu'ils puissent le plus facilement possible transformer cette matière numérique à des fins expressives, sans avoir à se préoccuper des systèmes sous-jacents liés au monde informatique.

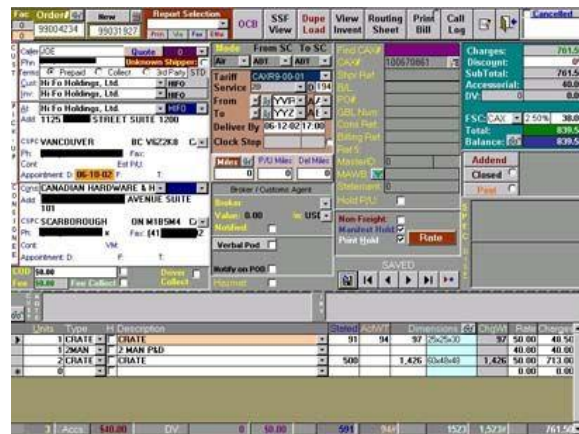


Illustration 19 - Exemple de GUI mal adapté à l'expérience utilisateur

Les outils numériques ont ainsi considérablement le travail de l'artiste. On peut aussi ajouter que l'emploi du médium numérique propose également une ouverture vers de nouvelles dimensions expressives telles que l'immersion ou l'interaction où la place du spectateur se retrouve une fois de plus transformée. La question de l'immersion sera plus précisément abordée en dernière partie.

2.3 Le style et le rendu artistique dans les réalisations numériques aujourd'hui: exemples de diverses œuvres s'inspirant des techniques traditionnelles dans un but expressif

Nous allons voir comment diverses formes d'expressivité se manifestent aujourd'hui au sein d'œuvres numériques dont le but n'est pas de produire un rendu réaliste. On va voir que ces réalisations s'inspirent souvent de méthodes picturales ou traditionnelles mais les exploitent d'une manière différente, propre à son médium et qu'elle reste toujours liée à certains codes hérités de l'histoire de l'art. Il n'y a ainsi pas de réelle discontinuité entre technique traditionnelle et numérique.

2.3.1 Film



Illustration 20 - Le tableau - Jean François Laguionie (2011)

Certains films d'animation s'inspirent du monde pictural pour créer des styles originaux et faire passer une vision au public tout en racontant une histoire. Par exemple, le long métrage d'animation *Le Tableau*³² est rempli de références à l'histoire de l'art et à ses symboles. On a dans ce film deux manières radicalement différentes d'utiliser le rendu 3D: le monde réel est représenté de manière hyper réaliste, avec des décors complexes, très détaillés. Au contraire, l'intérieur des tableaux est stylisé: il se constitue d'un mélange de décors en 2D et de personnages 3D. La différence de style entre le monde pictural des toiles et l'extérieur permet de séparer les deux univers. Un travail a été effectué pour que les personnages semblent être constitués de peinture à l'huile. La matière picturale joue un rôle essentiel dans le scénario et sert à exprimer une hiérarchie sociale entre les individus: ceux qui détiennent le pouvoir ont une apparence achevée, tandis que les autres, exclus de cette société, ne sont pas totalement peints. Certains personnages n'ont même pas de couleur et sont simplement esquissés. Ils peuvent quitter leur support pour voyager de tableau en tableau. La couleur employée revêt différentes symboliques. Le château dans lequel vivent les Toupins, personnages les plus privilégiés du monde du tableau, est lumineux, teinté de couleurs vives et vibrantes dans les tons chauds, tandis que les exclus sont relégués aux parties les plus sombres et ternes du tableau. Cette tendance s'inverse lors que les personnages inachevés décident de se peindre par eux mêmes et arborent des couleurs plus flamboyantes et resplendissantes encore que celles des Toupins. Dans les décors, elle sert également à indiquer la lumière: l'emploi de complémentaires permet de mettre en valeur

³² Laguionie, J.F.,(2011),*Le Tableau*

certaines zones en les rendant plus éclatantes. De plus, les références à des peintures existantes sont nombreuses: on peut ainsi noter par exemple une composition similaire à *l'Odalisque Rouge* de Matisse. Les traits et le style de certains personnages évoquent également la peinture de Modigliani et certaines scènes s'inspirent de la perspective impossible présente dans les œuvres d'Escher.



Illustration 21 - Comparaison de style entre le Tableau et la peinture de Modigliani

D'autres réalisations reprennent des caractéristiques de l'art pictural à des fins expressives comme par exemple le long métrage *Croc-Blanc* sorti en 2018³³. Si les personnages ont des proportions et une animation globalement réalistes, le rendu et la lumière imitent en partie le comportement de la peinture pour en faire une forme de tableau animé. La peinture numérique permet de gérer les différentes ambiances du film en les rendant plus intenses et en soulignant ainsi l'immensité des espaces d'Amérique du Nord³⁴.

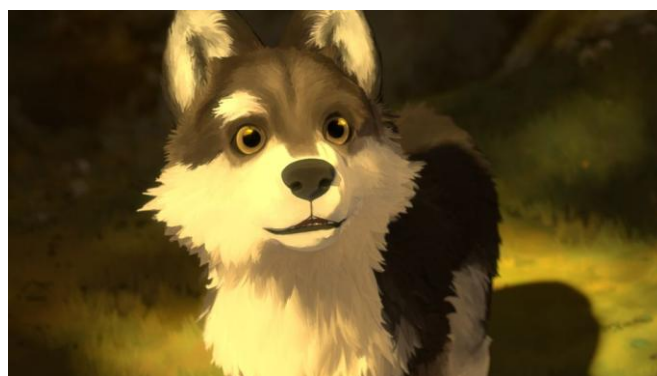


Illustration 22 - Croc-Blanc - Alexandre Espigares - 2018

On peut également citer certains films d'animations qui reprennent des caractéristiques stylistiques d'artistes pour y ajouter une dimension immersive: on retrouve

³³ Espigares,A.,(2018),*Croc-Blanc*

³⁴ AlloCine. *Croc-Blanc BONUS « La fabrication du film »*, réalisé par la Fondation Gan pour le Cinéma. Consulté le 25 avril 2019. http://www.allocine.fr/video/player_gen_cmedia=19577509&cfilm=248271.html.

ce type d'environnement dans certains films courts en 360° tels que *The starry night* de VR Motion Magics qui nous plonge dans le monde des tableaux de VanGogh. Il est possible d'explorer un univers reproduisant la manière de peindre de l'artiste. Les textures donnent un aspect pictural proche de ses œuvres. La dimension temporelle apportée par le film permet de générer du mouvement sur certains éléments, notamment le ciel, avec des petites touches qui suivent la direction des coups de pinceau.



Illustration 23- The Starry night - 360° VR short movie

2.3.2 Jeu-vidéo

Dans le domaine du jeu vidéo, il existe également de nombreuses innovations d'un point de vue de l'expression graphique. Si d'un côté le photoréalisme semble se développer de plus en plus grâce à la puissance croissante des cartes graphiques, il y a également chaque année une apparition de jeux présentant un style visuel plus original et personnel. La démocratisation des outils et moteurs de jeu permet à de plus en plus de studios et de concepteurs de réaliser des jeux de manière indépendante et d'y apporter une touche plus intime et artistique en expérimentant des techniques qui ne sont pas exclusivement tournées vers le photoréalisme. Cette production de jeux indépendants est aussi liée au nombre croissant de graphistes et de programmeurs sur le marché qui n'accèdent pas à des postes de création et finissent par former des studios de taille plus restreinte afin d'avoir plus de liberté et de moyens d'expression que dans l'industrie triple A.



Illustration 24 - Gris - Studio Nomada (2018)

Par exemple, dans le jeu *Gris*³⁵, sorti en fin 2018 par le studio Nomada, c'est un peintre, Conrad Roset, qui crée lui même des aquarelles qui servent de décor pour le jeu. Le spectateur se retrouve plongé dans un monde stylisé, composé de formes hybrides entre abstraction et figuration. L'environnement se déconstruit et se reconstruit grâce à des effets spéciaux 2D qui créent des impressions de matière et donnent une réelle consistance à l'œuvre: taches d'aquarelle qui se diluent, coups de crayon, morphing... Le compositeur de musique, Marco Albano dira ainsi : « L'idée était de développer un récit dans lequel le joueur puisse se perdre et avoir sa propre idée de ce qui se passe à l'écran mais aussi de combiner des éléments artistiques, comme lorsque vous êtes debout dans une galerie d'art en regardant un tableau, avec des éléments de jeu. »³⁶ Il en résulte une œuvre complexe, mêlant peinture, musique et jeu. Un certain nombre jeux vidéos de ce type sont sortis ces dernières années, arborant tous un style unique comme *Botanicula*³⁷ et *Journey* en 2012, ou *Machinarium* en 2009. Dans *Gris*, le travail artistique est majoritairement effectué manuellement: les décors sont peints, l'animation est faite en 2D « traditionnelle ». Il serait intéressant de voir s'il est possible d'obtenir des résultats aussi expressifs de manière plus procédurale.

³⁵ Cuevas.A,Roset.A, (2018), *Gris*.

³⁶ « 50 shades of Gris: How Nomada Studio created 'a game that evolves visually, not only mechanically' ». MCV (blog). Consulté le 2 janvier 2019. <https://www.mcvuk.com/50-shades-of-gris-how-nomada-studio-created-a-game-that-evolves-visually-not-only-mechanically/>.

³⁷ (2015), *Beyond Eyes*.

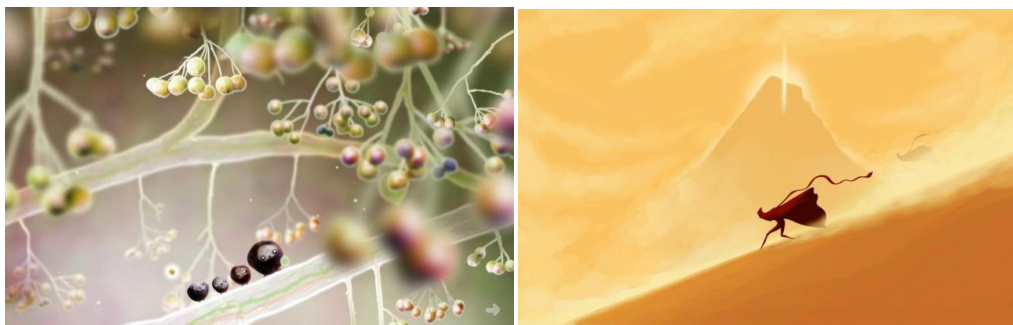


Illustration 25 - Journey - thatgamecompany (2013)

Illustration 26 - Botanicula - Amanita Design (2012)

Les effets artistiques permettent parfois d'apporter des éléments de gameplay. Par exemple, dans le jeu *Beyond Eyes* du studio Tiger & Squid où l'on incarne une fille aveugle, la représentation de son environnement revêt deux rôles différents. Elle permet d'abord d'imager et d'apporter des informations sur son intériorité et sa manière de percevoir le monde puisqu'elle consiste en un mélange de représentations imaginées à partir des sons et de souvenirs que la petite fille a accumulés avant de perdre la vue. Les tons pastels diffus apportent une certaine douceur et un aspect légèrement confus et nébuleux à l'ambiance du jeu qui s'accordent ainsi à l'état mental du personnage. Elle sert en plus à contraindre notre manière de jouer: les parties blanches représentent les zones inconnues de son environnement. La matière visuelle du jeu apparaît ou s'estompe en fonction de ce que le personnage touche, sent ou entend et se dilue autour d'elle, ce qui crée un effet graphique original, ressemblant un peu au comportement de l'aquarelle.



Illustration 27 - Beyond Eyes - Tiger and Squid (2015)

2.3.3 Installations interactives



Illustration 28 - Koro - Damien Traversaz
Illustration 29 - Body Paint - Mehmet Akten (2010)

En allant voir du côté des installations artistiques, on observe également la présence de nombreuses formes expressives qui sont amplifiées par les possibilités offertes par l'interactivité et l'immersion. Dans *Koro*³⁸, de Damien Traversaz, une installation interactive, deux personnes face à un écran génèrent des effets visuels et de la musique en fonction des battements de leur cœur et du contact physique. Les images s'animant sur l'écran évoluent en temps réel et semblent constituées de traces de peinture abstraite colorée. Ici, c'est le spectateur qui participe à l'aspect final de l'œuvre grâce à une forme d'expression inconsciente liée au corps. Les sens du spectateur se mêlent grâce à un ensemble de musique, de son, d'image et de toucher. On retrouve ce lien entre corps et expression dans l'installation *Body Paint*³⁹ de Mehmet Akten où le spectateur utilise le mouvement de son corps et le geste pour peindre sur un tableau virtuel. Ce procédé, qui n'est pas sans rappeler celui de l'action painting, permet de générer une infinité de formes abstraites vibrant et s'étalant sur la surface.

3. Les techniques de rendu expressif

Nous allons à présent exposer diverses méthodes programmables permettant de générer des rendus induisant une forme d'expressivité. Ces techniques sont généralement

³⁸ « Koro Installation Interactive Battements de Coeur Damien Traversaz ». damien-traversaz. Consulté le 6 mars 2019. <https://www.damientraversaz.com/koro>.

³⁹ « Body Paint ». *Memo Akten* (blog). Consulté le 7 avril 2019. <http://www.memo.tv/portfolio/bodypaint/>.

destinées soit au rendu d'images fixes, de vidéos, de scènes 3D et peuvent ou non fonctionner en temps réel. On notera également que l'expressivité d'une œuvre numérique ne provient pas seulement du rendu, mais aussi de l'animation qui peut être plus ou moins stylisée ou encore de la modélisation, qui peut avoir différents degrés de réalisme ou d'abstraction. Il est également possible de jouer sur les styles en mélangeant différents types de rendus. Ces aspects étant trop en dehors de notre sujet principal, ils ne seront pas abordés dans ce mémoire mais demeurent des pistes éventuelles à explorer pour aller plus loin dans l'étude des moyens expressifs offerts par les outils numériques.

3.1 Le rendu en images de synthèse

3.1.1 Définition et fonctionnement global

Avant de parler de rendu expressif, il nous faut commencer par définir en quoi consiste l'étape de rendu dans le pipeline de création 3D. Le rendu en image de synthèse constitue l'ensemble des opérations permettant de générer des images à partir des informations contenues dans une scène 3D telles que des modélisations, des lumières, des particules... Il s'agit ainsi d'obtenir une représentation en deux dimensions d'un ensemble d'éléments se trouvant dans un espace 3D virtuel. Cette technique date des années 1960 et avait été inventée à la base pour l'aviation par William Fetter qui avait pour la première fois généré l'image d'un humain en vue orthographique. A l'époque, il avait appelé procédure « Computer Graphics »⁴⁰.

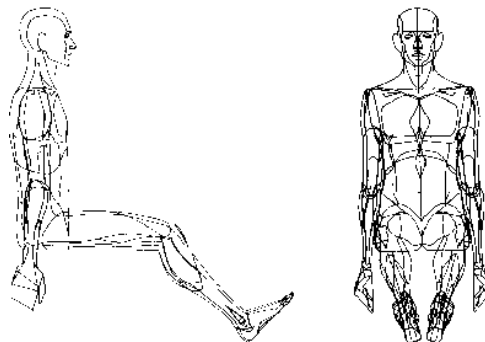


Illustration 30 - First human in orthographic view - William Fetter

⁴⁰ « L'Histoire du Rendu 3D ». Consulté le 2 novembre 2018. <http://www.3dvh.com/dossier-383-1-histoire-rendu-3d.html>.

Il existe aujourd'hui deux principales techniques de rendu: le temps réel et le pré-calculé. Le rendu pré calculé, qui jusqu'à maintenant était généralement utilisé dans le domaine du film d'animation, permet de générer des images demandant des calculs complexes et importants. Cela permet par exemple d'obtenir des rendus de haute qualité et physiquement réalistes grâce à certaines techniques telles que le raytracing. Cependant, les temps de calculs étant longs, il ne permet pas l'interactivité et n'est donc pas utilisable pour réaliser un jeu-vidéo ou une application interactive.

Dans ce cas, on est amené à utiliser les techniques de rendu temps-réel. Comme son nom l'indique, le rendu temps-réel permet de rendre une grande quantité d'images en un temps très restreint, généralement entre 30 et 60 images par secondes pour un jeu vidéo. Ce type de rendu apporte un certain nombre de contraintes: il demande d'abord un grand travail d'optimisation. Par exemple, il faut modéliser avec un nombre restreint de polygones, éviter d'user de textures trop lourdes. Il n'est pas possible d'obtenir la même qualité qu'avec le pré calculé et le réalisme des images s'en trouve ainsi atteint, notamment au niveau des lumières ou des réflexions, puisque pour l'instant, la plupart des moteurs temps réels ne gèrent pas le rebond de lumière. Bien qu'il existe aujourd'hui des raytracers dédiés au temps réel comme le RTX de NVIDIA, le nombre de cartes graphiques pouvant le supporter est encore restreint et il est coûteux d'obtenir des machines suffisamment performantes pour permettre ce type de rendu.

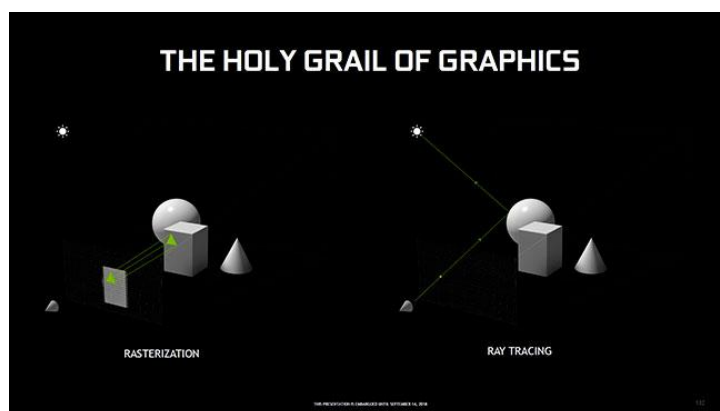


Illustration 31 - Schéma représentant les deux types de rendu: Ray tracing et Rastérisation

3.1.2 Le choix du rendu temps réel

Ce mémoire porte principalement sur l'utilisation du rendu temps réel à des fins expressives pour le film d'animation. Bien que certains tests aient été effectués en utilisant

des raytracers en seconde partie, la majorité des expérimentations sont réalisées à l'aide de moteurs temps réel. Nous allons donc nous concentrer plus particulièrement sur ce type de rendu et expliciter son fonctionnement. Le choix s'est porté sur cette technique car elle permet d'expérimenter plus rapidement et de manière beaucoup plus directe. Cette forme d'immédiateté rend le processus de création plus intéressant pour les artistes qui peuvent entrevoir directement le résultat de leur travail. Etant donné que cette étude porte sur le rendu artistique expressif et non sur la recherche du photoréalisme, il est ainsi pertinent de privilégier une plus grande aisance pour les artistes afin de leur offrir de multiples libertés artistiques au détriment de la puissance de calcul qui aurait été permise par le pré calculé et qui n'est pas forcément utile en ce qui concerne l'effet recherché. De plus, lors de la réalisation de créations artistiques, on obtient au commencement même des résultats visuels. La plupart des éléments peuvent être changés à tout moment sans engendrer de problèmes dans l'organisation du pipeline. A l'échelle individuelle, par exemple pour un projet étudiant, l'emploi du temps réel offre un gain de temps considérable puisqu'il n'est plus nécessaire de patienter durant le temps de rendu. Il est ainsi possible d'apporter des modifications au film jusqu'au dernier moment, et il nous n'avons plus à recourir à un matériel onéreux tel que les renderfarms. Pour une production à grande échelle, l'utilisation de ces techniques permet une forme de parallélisation des étapes et ainsi une plus grande collaboration entre les artistes de différentes équipes.

3.1.3 Le pipeline graphique et la méthode de rendu par rasterisation

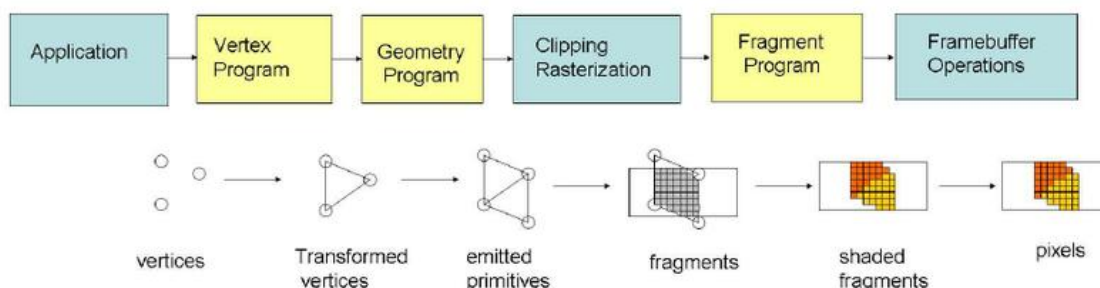


Illustration 32 - Schéma représentant le pipeline graphique pour OpenGL

Afin de mieux saisir les enjeux liés à la création d'outils de rendu, nous allons effectuer une brève introduction au pipeline graphique 3D dans le but de déterminer et de

comprendre les étapes standards liées au rendu temps réel par rasterisation. Le pipeline est une architecture de logiciel contenant une suite d'opérations exécutées en série. Pour chaque opération, le résultat devient une donnée d'entrée pour l'opération suivante. Il est ainsi divisé en différentes étapes que l'on appelle stages, qui s'exécutent l'une à la suite de l'autre. Ces étapes se calculent en parallèle via le processeur graphique. Le processeur graphique ou GPU pour « graphic compute unit » est un composant constitué d'un grand nombre de micro-processeurs qui permettent de réaliser un nombre important de calculs et d'opérations en parallèle. Comme l'indique son nom, il est historiquement dédié au calcul d'images pour l'affichage sur un écran et s'est spécialisé par la suite dans le rendu d'images 3D. En effet, si l'on souhaite, par exemple, calculer la couleur en sortie de chaque pixel de l'écran, il est beaucoup plus simple et rapide de traiter chaque pixel simultanément, plutôt que l'un après l'autre comme on aurait pu le faire avec le CPU, l'unité centrale de traitement, qui n'a pas la possibilité d'effectuer un aussi grand nombre de tâches simultanément. De plus, certaines fonctions sont implémentées directement dans l'hardware du GPU et donc s'effectuent plus rapidement, comme par exemple les transformations matricielles qui sont complexes à calculer. Il existe également des circuits qui sont programmables par l'utilisateur. Ils exécutent des programmes que l'on appelle *shaders* qui possèdent des processeurs dédiés. Il peut ainsi y avoir plusieurs types de *shader*: *vertex shader*, *fragment shader*, *geometry shader*... Une partie d'entre eux seront plus particulièrement détaillés dans la deuxième et troisième partie du mémoire. La carte graphique dispose d'une architecture hybride, consistant en un mélange entre des unités SIMD ou SIMT et MIMD selon les fabricant. Le but du pipeline graphique est de rendre un ensemble de données issues d'une scène 3D grâce à différentes étapes de transformation que nous allons à présente expliciter.

Application stage

Lorsqu'une application s'exécute en temps réel, il faut à chaque frame remettre à jour les informations des différents éléments constituant la scène 3D. C'est à cette étape que se calculent par exemple la physique, le son, les détections de collision, ou encore l'animation. Il faut également récupérer les données entrées par l'utilisateur. Ainsi, diverses informations telles que les rotations, positions ou échelle sont mises à jour. Ces opérations s'effectuent via le CPU.

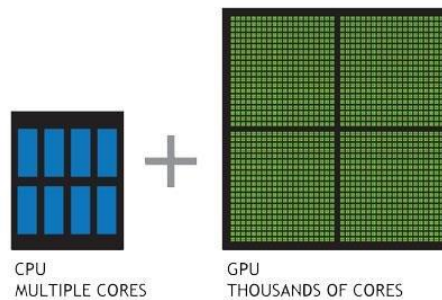


Illustration 33 - Le GPU possède des milliers de micro processeurs tandis que le CPU en possède moins mais avec une plus grande puissance de calcul

La première étape consiste à récupérer les données de la scène à partir du CPU. Il s'agira de par exemple de géométries 3D, de lumières, ainsi que de différents attributs liés au matériau des objets de la scène. On récupère également les attributs, la position et l'orientation de la caméra qui va définir le point de vue. Ces informations sont chargées à partir du disque sur la RAM. Le but est de convertir ces données pour qu'elles soient utilisables par le GPU. Il faut donc pouvoir gérer l'interaction entre CPU et le GPU: cette communication se fait par ce qu'on appelle le *command buffer* ou ring buffer. Les données sont ainsi stockées sur la mémoire vidéo ou VRAM qui est la mémoire dédiée au GPU à laquelle il peut accéder très rapidement. L'accès à la mémoire vidéo se fait par différents tampons: les *vertex buffers*, qui contiennent les vertices ou sommets, les *frame buffers* qui comprennent les pixels à afficher à l'écran dans un tableau à deux dimensions, et les textures qui représentent des images et existent sous formes de tableaux de une à trois dimensions. L'utilisateur peut envoyer des directives au GPU via des APIs telles que DirectX, Vulkan ou encore OpenGL. Pendant ce temps, le CPU est disponible pour effectuer d'autres calculs. C'est à la fin de cette première étape que l'on va pouvoir exécuter le pipeline graphique.

Quand le CPU envoie une requête au GPU, il se produit deux étapes. D'abord, il configure l'état du rendu, c'est à ce moment qu'il va aller chercher le *vertex shader*, le *pixel shader*, les textures et les paramètres de lumière et de rendu liés à l'objet. Il s'agit ensuite de rendre cet objet, c'est ce qu'on appelle un *draw call*. Les informations sont transférées à la mémoire vidéo pour créer un état ou *render state*. On peut ainsi dessiner toutes les géométries qui utilisent le même état. Pour cela, on crée *batch* qui représente tout ce qui doit être modifié pour configurer l'état de rendu. Par exemple, pour gagner du temps, le

moteur de jeu Unity connecte tous les objets ayant les mêmes matériaux, textures ou *shaders*. Ceux-ci sont alors tous rendus via un unique *batch*.

Géométrie stage ou vertex processing

Durant cette étape, on récupère les données en entrée et on les analyse pour obtenir un tableau de vertices. Les vertices sont tout simplement des structures de données contenant généralement des informations telles que les coordonnées d'un point dans l'espace 3D, des couleurs, ou bien encore des coordonnées UV, des tangentes ou des normales.

On commence par la phase **modeling transformation** : les coordonnées des vertices sont initialement placées selon le repère local de l'objet. C'est à dire que le modèle possède un point, par exemple au centre de l'objet, qui représente l'origine du repère. Il faut convertir ces coordonnées en coordonnées globales dans ce qu'on appelle le repère monde ou world space qui contient tous les objets de la scène. C'est également lors de cette étape que l'on effectue des calculs sur les vertices en fonction de leurs attributs comme de l'animation procédurale.

Vient ensuite la phase d'**illumination** ou de **shading**. Il s'agit de calculer une couleur pour chaque vertex en se référant aux positions et propriétés des sources de lumière ainsi qu'aux positions, normales et propriétés du matériau assigné au vertex courant. A ce moment, il peut aussi y avoir un calcul de texture pour chaque vertex. Il est également possible dans certains cas que cette opération s'effectue après l'étape de *view transformation* que nous allons à présent présenter.

Ensuite, durant l'étape de **view transformation**, on passe les coordonnées du repère globale au repère de la caméra, aussi appelé *view space*. Le *view space* est un repère qui a pour origine la position de la caméra. On effectue cette transformation grâce à une matrice de changement de repère. Généralement, l'axe de la caméra est orienté vers l'axe z ou -z.

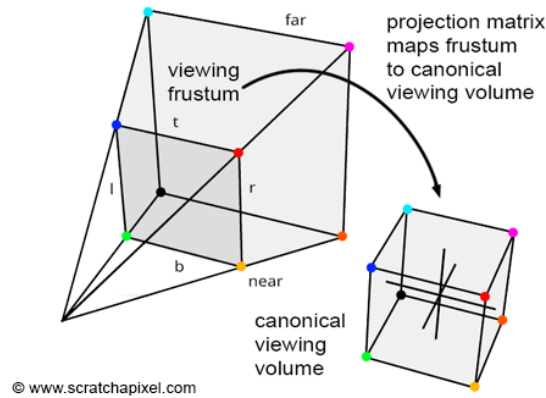


Illustration 34 - Transformation de la vue frustum

L'étape de **projection** a pour but de projeter des objets 3D sur un plan 2D. Il est possible d'effectuer une projection orthographique, dans ce cas il n'y aura pas de déformation par la perspective et tous les objets auront la même taille peu importe leur distance par rapport à la caméra, ou une projection en perspective qui sera alors définie avec une longueur de focal. Pour cela, on crée un *view volume* qui effectue une projection devant la caméra. En projection orthographique, ce *view volume* aura une forme rectangulaire alors qu'en perspective il sera pyramidal. Il permet de définir un *frustum*, une pyramide tronquée par deux plans : le *near plane*, proche de la caméra et le *far plane*, plan le plus éloigné de la caméra. Ensuite, il va falloir passer les coordonnées dans ce qu'on appelle le *clip space*, un espace de projection dans lequel le *frustum* est transformé en un cube dont les coordonnées des sommets varient de -1 à 1 dans chacun des axes, qu'on appellera le *canonical view volume*. Cette opération s'effectue grâce à une matrice de projection. On obtient alors des coordonnées homogènes, c'est à dire normalisées qui vont rendre plus simples les opérations sur le GPU en les linéarisant. Il est ainsi possible par exemple d'effectuer une rotation, translation et modification d'échelle en une seule étape au lieu de trois.

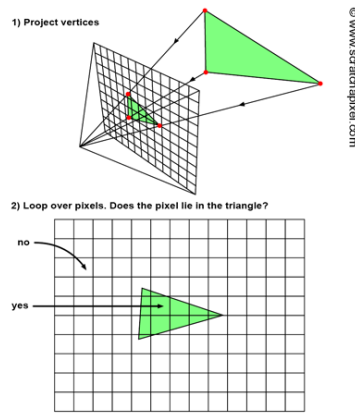


Illustration 35 - Projection de vertices sur un plan 2D

Aujourd’hui, la plupart de ces opérations sont effectuées via un programme que l’on appelle **vertex shader**, qui s’exécute en parallèle sur chaque vertex à calculer. D’autres opérations comme la **tessellation** peuvent également être effectuées. Cette étape, non obligatoire, se compose généralement de trois sous étapes: le *control*, le *tessellator* et l’*evaluator*. Elle permet de créer de nouvelles primitives, de subdiviser le *mesh*, et d’y appliquer du *displacement* afin de déplacer la géométrie et de gérer les niveaux de détail. Les données peuvent également passer par le **geometry shader**. Ce programme, qui est aussi optionnel, peut générer nouveaux vertices à partir de ceux donnés en entrée. Cela permet d’effectuer diverses opérations utiles via le GPU, comme par exemple extruder. Cette étape est fréquemment utilisée pour générer un grand nombre de formes simples telles que de l’herbe ou de la végétation.

Il faudra ensuite assembler les vertices pour former des primitives. Cette étape s’appelle **primitive assembly**: elle permet de relier les vertices afin de créer des primitives qui peuvent être des triangles, des quads, des lignes ou des points. On génère par exemple une interpolation entre les vertices constituant l’arête d’un triangle.

L’étape de **clipping** a pour but premier de repérer les primitives qui se situent à l’intérieur ou à l’extérieur du frustum grâce à une opération appelée frustum culling. Si elles sont en dehors, elles seront simplement ignorées. Sinon, si elles entrent en intersection avec le frustum, il faudra couper la primitive et créer deux nouveaux vertices au point de l’intersection.

Enfin, durant l’étape de **screen mapping**, les coordonnées des primitives projetées qui sont à trois composantes (X_p, Y_p, Z_p) sont transformées en coordonnées écran ou screen coordinates à deux composantes (X_s, Y_s). Si on ajoute la valeur de profondeur

comme troisième composante, on obtient ce qu'on appelle les window coordinates (Xs, Ys, Zw). Cette valeur sera utilisée plus tard, dans le fragment shader.

Rasterisation

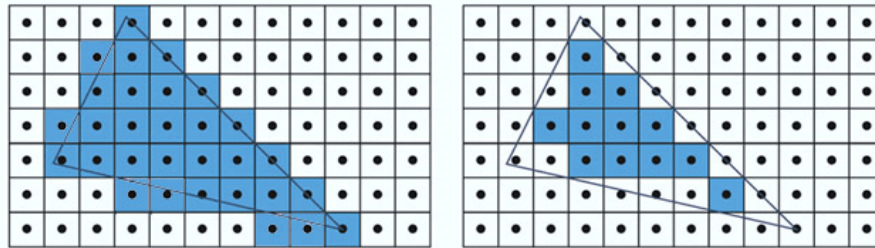


Illustration 36 - Etape de rasterisation d'un triangle

Vient après la rasterisation: les primitives, qui sont des données vectorielles, sont mappées pour passer en données matricielles, c'est à dire en un tableau de fragments afin de pouvoir les afficher sur un écran. Le fragment est une donnée permettant de générer un pixel. En se basant sur le *screen space normalized*, on détermine quel pixels vont dessiner la primitive sur l'écran. Si le pixel est en dehors de l'écran, il sera ignoré. On interpole l'information du *vertex shader* avant de l'envoyer dans le pixel shader. Il est également possible d'effectuer du *backface culling*, une opération qui permet d'ignorer les faces dont la normale n'est pas orientée face à la caméra.

Fragment processing

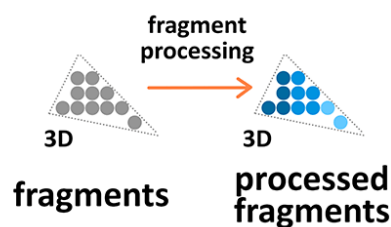


Illustration 37 - Etape de fragment processing

Durant l'étape du *fragment processing*, on génère la couleur finale des pixels grâce au *fragment shader* ou *pixel shader*. Pour chaque fragment, il y a au moins deux buffers : le *color buffer* ou *frame buffer* et le *Zbuffer* qui contient les informations de profondeur. Il est possible, grâce au *fragment shader*, d'ajouter des effets de *post processing* comme des

corrections colorimétriques, ou encore un vignettage. Les données en sortie contiennent généralement la couleur du pixel ainsi que l'alpha et la profondeur ou *zdepth* du pixel.

Visibilité

Il faut ensuite effectuer un certain nombre de tests de visibilité.

L'*alpha test* est utilisé pour pouvoir afficher ou ignorer un fragment en fonction d'une valeur seuil de l'alpha. Si son alpha est en dessous de cette valeur seuil, le fragment ne sera pas pris en compte.

Le *stencil test* permet à partir d'un masque de réduire une aire de rendu en ignorant certains fragments.

Le *depth test* consiste, à la position actuelle du fragment, à évaluer la profondeur du pixel par rapport celle du frame buffer. C'est une manière d'effectuer de l'*occlusion culling*. Il s'agit d'un test qui vérifie si un objet est en partie ou totalement masqué par des objets qui se situent devant lui. Si la profondeur du fragment est plus élevée que celle du buffer, alors il sera ignoré. Ainsi, les parties des objets situés derrière un autre objet ne seront pas rendues.

D'autres opérations peuvent également s'effectuer comme le *blending* qui permet grâce à l'alpha de combiner la couleur du fragment courant à celle du *framebuffer*, ou le *color masking* qui sépare les composantes RGB pour que l'utilisateur puisse choisir quelle *channel* il souhaite afficher.

Frame buffer

Les pixels sont envoyés dans le *frame buffer* qui permet, à partir d'un emplacement dans la mémoire, d'envoyer les directives d'affichage au moniteur. Pour afficher une image, on utilise généralement deux buffers ou tampons au minimum: un *front buffer* qui comprend les pixels affichés à l'écran et un *backbuffer*. Cela permet d'éviter d'avoir l'image suivante qui se forme par dessus l'image précédente au moment de rafraîchir le buffer. Pour passer de l'un à l'autre et ainsi rafraîchir l'image, plusieurs méthodes sont possibles: les pixels peuvent être copiés d'un tampon à l'autre durant la période de rafraîchissement ou bien être échangés, ce qu'on appelle le *swapping*, ce qui signifie que une fois sur deux le front buffer devient *backbuffer* et ainsi de suite.

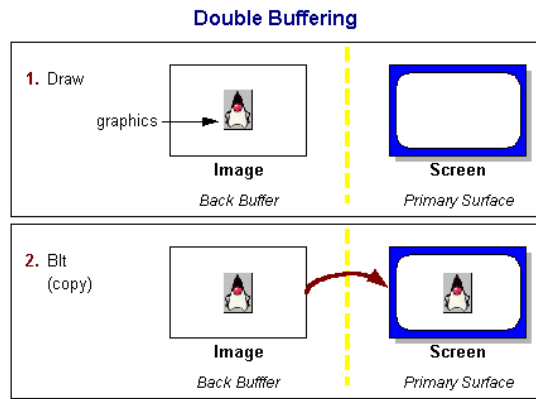


Illustration 38 - Schéma représentant le fonctionnement du double buffering

On a donc vu que le rendu par rasterisation consistait à générer une image à partir de données 3D et d'une vue définies par l'utilisateur. Ensuite, ces données sont transposées dans ce qu'on appelle le *view space*. On cherche ensuite quel pixel de l'écran sera rempli par la primitive, il s'agit de la rasterisation. Enfin, on détermine la couleur du pixel dans le *fragment shader* et on effectue différents tests tels que le *zTest* pour déterminer si l'objet est caché derrière un autre. Les pixels sont finalement affichés sur l'écran via le frame buffer.

3.2 Le rendu expressif

3.2.1 Définition

Le rendu expressif est une catégorie regroupant diverses techniques de rendu dont l'objectif est de permettre à l'utilisateur de donner à voir une forme de subjectivité par le biais de procédés d'amplification, de simplification et de détournement du réel. Elle accorde une importance à la manière de représenter, autrement dit au style de l'artiste. On emploie souvent le terme de rendu non photoréaliste ou NPR pour décrire cet ensemble de techniques mais je n'apprécie pas vraiment cette expression car elle les définit négativement sans décrire précisément en quoi elles consistent et demeure donc incomplète.

En rendu expressif, on est souvent amené à s'inspirer de mediums traditionnels tels que le crayon, les hachures, la peinture ou encore l'aquarelle. Ce procédé est couramment destiné à la communication et est utilisé pour faire passer un message ou une intention. Il pose également la question de comment l'artiste veut représenter le monde et accorde une importance à la réception, en se questionnant notamment sur la manière avec laquelle le

spectateur interprète l'œuvre et quels procédés artistiques permettent d'engendrer certains effets sur le public.

Ce type de rendu revêt diverses fonctions: il permet d'abord de développer et de créer de nouveaux styles graphiques dans le but d'exprimer une vision, une subjectivité en passant par l'abstraction et la stylisation. L'image peut faire ressortir une humeur, une ambiance ou une émotion propre à l'artiste. Il offre également la possibilité de donner à voir en mettant en valeur certains éléments d'une manière spécifique. C'est d'ailleurs pour cette raison qu'il peut être utilisé en ingénierie ou en architecture car il aide à simplifier le réel en allant à l'essentiel. Cela permet de réaliser des plans ou dessins techniques lisibles et compréhensibles de manière rapide et automatisée en éliminant les détails superflu.

Le terme de rendu expressif est souvent utilisé par opposition au rendu dit photoréaliste qui vise une représentation objective du réel. Cependant, il n'y a pas de limite stricte entre ces deux procédés: en effet, on constate qu'il y a toujours une intention donnée dans les œuvres dites photoréalistes et les artistes y introduisent souvent des effets plus artistiques afin de passer une idée, une émotion ou un message. Avec le rendu photoréaliste, tout comme en prise de vue réelle, on peut par exemple jouer sur l'éclairage, le cadrage ou sur la disposition des éléments de la scène pour exprimer une vision spécifique.

On retiendra ainsi que le rendu expressif consiste en un ensemble de techniques de rendu ayant pour mission principale de permettre à l'utilisateur de partager une vision personnelle et subjective du monde en l'amenant à développer sa propre manière de représenter. Il en résulte un visuel original pouvant exprimer une idée, une intention, une signification ou une émotion particulière. Ces méthodes servent donc à aider les artistes à transmettre un message de manière efficace.

3.2.2 Caractéristiques relatives au rendu expressif

Le rendu expressif présente généralement un ensemble de caractéristiques qui permettent de le distinguer des autres méthodes de rendu.

On relève tout d'abord la présence d'une forme d'abstraction ou de simplification. Ces techniques peuvent servir à mettre de l'emphase sur certains éléments selon les désirs

de l'artiste. Cela permet généralement de rendre l'image claire et intelligible pour le spectateur.

Ensuite, on peut noter l'emploi de la stylisation, avec par exemple l'ajout de contour, ou l'emploi de couleurs « plates » et d'ombres dures comme on peut le voir avec la technique du cel-shading, qui imite le rendu d'animation en deux dimensions.

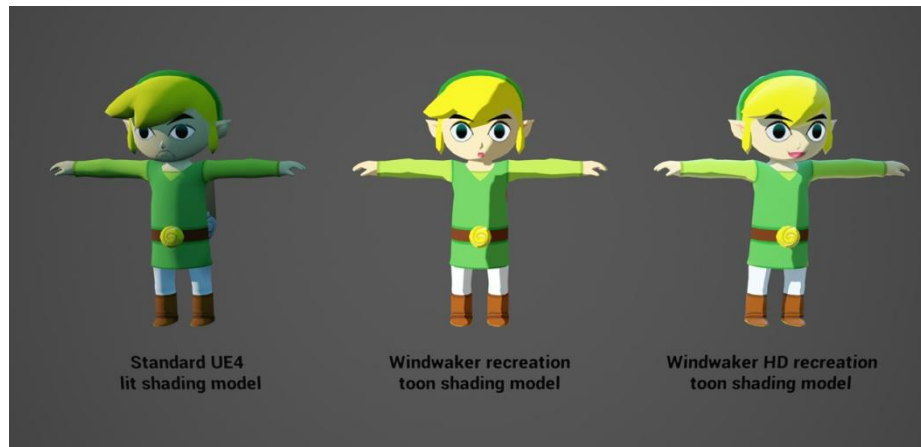


Illustration 39 - Exemple de cel-shading sous le moteur Unreal Engine

Il faut également prendre en compte la notion d'aléatoire et d'erreur. Le rendu expressif permet généralement de s'éloigner de l'aspect lisse et parfait attribué au photoréalisme. Ainsi, il est possible d'avoir des traits ou des contours tremblants ou imparfait qui peuvent ressembler à de l'esquisse. C'est ce qu'on peut voir dans la technique « loose and sketchy »⁴¹ de Cassidy Curtis. Son algorithme redessine la silhouette d'un objet 3D en partant d'une *depth map* qui génère également un champ de vecteurs. L'image de la silhouette indique le niveau de remplissage d'encre nécessaire en se basant sur les contours. Un système de particules est généré par la suite avec, pour chaque particule, une accélération qui varie en fonction du champ de force et de paramètres aléatoires. Elles remplissent progressivement les zones de contour nécessitant de l'encre en créant, selon les paramètres donnés, des lignes plus ou moins lisses et plus ou moins continues. On se retrouve à la fin avec différents styles de dessins au trait.

⁴¹ « Loose and Sketchy Animation ». Consulté le 16 mars 2019.

<http://otherthings.com/uw/loose/sketch.html>.



Illustration 40 - Technique Loose and Sketchy

Le traitement de la couleur et de la lumière est important. On peut par exemple choisir d'employer les couleurs de manière non réaliste en jouant sur les contrastes et la complémentarité comme on peut le voir dans certains rendus qui imitent la peinture impressionniste tels que ceux du jeu *Rock of Ages 2*. Il est également possible de jouer sur la diffusion de la lumière. Certaines personnes comme Hadyn Lander⁴² ont par exemple tenté d'animer les ombres en utilisant des fonctions de *perlin noise* dans le but d'imiter l'étalement de l'aquarelle sur du papier.

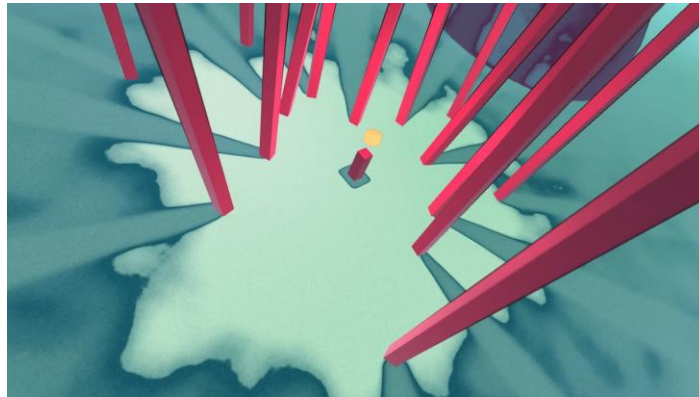


Illustration 41 - Effet aquarelle sur Unity - Hadyn Lander

Enfin, il ne faut pas oublier la question de la gestion des détails. L'artiste est amené à faire des choix sur ce qu'il veut mettre en valeur, en peignant par exemple certains éléments avec plus ou moins de précisions. Il peut s'agir par exemple de détecter traits du visage, de placer des détails en fonction de la profondeur ou des contours des objets. Il est également possible de gérer ces détails à l'aide de calques.

⁴² « Watercolor Shader Prototype ». Gumroad. Consulté le 14 novembre 2018.

<https://gumroad.com/l/skybreaker-watercolor>.

3.2.3 Apports techniques et utilisations

Ce type de rendu peut par exemple être employé dans le cas de limites techniques afin de privilégier le message ou l'émotion à transmettre. Dans l'étude *Regression Progression: Non-Photorealistic Rendering Choices in Game Design*⁴³ de Tom Hurlbutt, il prend l'exemple de l'utilisation du *cel shading* dans le jeu *The Wind Waker* dans un but expressif. Il évoque le fait qu'à cette époque, il aurait été difficile de rendre correctement les émotions du personnage de manière réaliste et que l'emploi de la stylisation a permis de créer un rendu plus attirant pour le spectateur. En effet, lorsque nous sommes face à une image très réaliste, nous avons tendance à remarquer plus facilement le moindre défaut de celle-ci, ce qui peut totalement ruiner l'effet recherché. Il s'agit de la vallée de l'étrange: un concept de Masahiro Mori qui explique que plus un avatar atteint un haut degré de réalisme, plus les erreurs et imperfections de celui-ci vont nous apparaître comme dérangeantes. Ainsi, dans le cas de ce jeu, l'emploi de techniques non réalistes permet de générer un visuel plaisant tout en faisant passer les émotions du personnage afin de permettre une proximité entre celui-ci et le spectateur.



Illustration 42- Utilisation de la simplification et de la stylisation dans le jeu Wind Waker

Le rendu expressif peut également servir à mettre de l'emphase et à guider le spectateur en mettant en valeur certaines informations spécifiques, ou en lui indiquant comment et où regarder. Par exemple, dans l'article *Directing gaze in 3D models with stylized focus*⁴⁴ publiée en 2006, les chercheurs ont tenté d'utiliser le rendu non réaliste pour diriger le regard du spectateur en rendant certains éléments de la scène plus ou moins

⁴³ Hurlbutt, Tom. « "Regression" Progression: Non-Photorealistic Rendering Choices in Game Design », s. d., 25.

⁴⁴ « Directing Gaze in 3D Models with Stylized Focus ». Consulté le 16 mars 2019.

https://gfx.cs.princeton.edu/pubs/Cole_2006_DGI/index.php.

détaillés. L'étude *Abstracted painterly renderings using eye-tracking data*⁴⁵ par Anthony Santella et Doug DeCarlo consistait à traquer le parcours du regard sur des images pour ensuite gérer le niveau de détail, les zones étant les plus regardées ayant le plus haut niveau de détail. Cela permet de palier au fait que les effets de peinture ont souvent tendance à s'appliquer partout de la même manière et ne permettent pas de donner une intention à l'image.

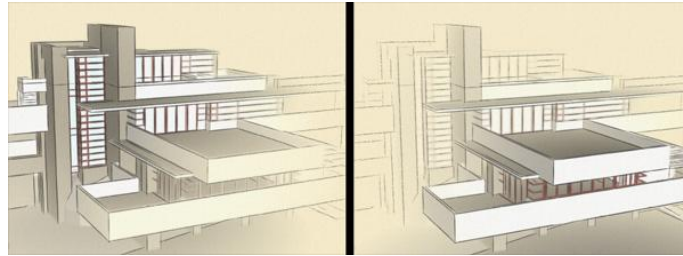


Illustration 43 - Différence de focus dans la méthode Directing Gare in 3D Models with Stylized Focus

3.3 Tour d'horizon de techniques de rendu expressif et étude du lien entre l'outil et l'utilisateur

Nous allons effectuer un état de l'art non exhaustif de certaines techniques de rendu expressif ayant été développées depuis les années 1990 jusqu'à aujourd'hui afin d'avoir une vision plus globale des différents moyens mis en œuvre pour obtenir des effets non réalistes grâce à l'outil numérique. Il s'agira également d'évaluer le degré d'interaction entre le programme et les artistes ainsi que les libertés qu'offrent chacune de ces techniques en terme de développement de style et d'expression artistique.

3.3.1 Filtres et traitement d'image

Les techniques de rendu expressif par filtre reposent sur différents processus de traitement d'image. Les filtres permettent d'appliquer un certain nombre d'effets sur une image mais peuvent également fonctionner sur des vidéos et sont particulièrement adaptés au rendu temps réel car peu coûteux en terme de calcul et facilement réalisables en *post*

⁴⁵ Santella, Anthony, et Doug DeCarlo. « Abstracted Painterly Renderings Using Eye-tracking Data ». In *Proceedings of the 2Nd International Symposium on Non-photorealistic Animation and Rendering*, 75–ff. NPAR '02. New York, NY, USA: ACM, 2002. <https://doi.org/10.1145/508530.508544>.

processing via le GPU, étant donné qu'il s'agit ici de traiter et d'effectuer des opérations sur chaque pixel de l'image.



Illustration 44 - Création d'effet de peinture par convolution

La plupart de ces techniques reposent sur la transformation des pixels par ce qu'on appelle un *kernel* ou matrice de convolution. Ce type d'opération permet par exemple d'obtenir un effet de flou, ou d'effectuer de la détection de contour. Nous en utiliserons notamment certaines au cours des expérimentations dans la deuxième partie du mémoire. Dans ce domaine, on peut citer par exemple la différence des gaussiens ou Dog qui consiste à effectuer une soustraction entre deux images ayant été floutées avec une intensité différente puis à réaliser une convolution. Cette méthode, que l'on doit à Holger Winnemöller, permet de créer des effets crayonnés, ou hachurés, ou encore de souligner les ombres des objets en gérant le niveau d'abstraction. Il existe un grand nombre de techniques utilisant le principe des filtres et celles-ci peuvent se combiner entre elles. Par exemple, dans l'article *Image and Video Abstraction by Coherence-Enhancing Filter*⁴⁶, les chercheurs génèrent des formes d'abstraction à partir de vidéos en temps réel grâce à une succession de lissages adaptatifs et de renforcements de contours et à un tenseur qui sert à guider les opérations de lissage en récupérant des vecteurs orientés en fonction des contours présents dans l'image.

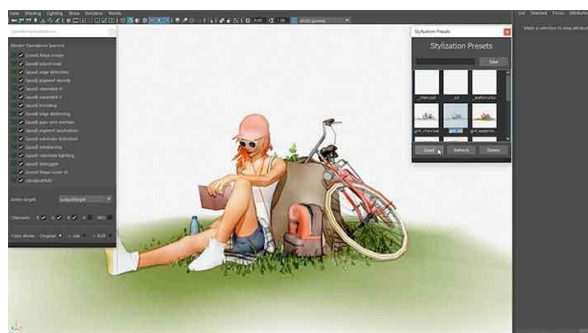


Illustration 45 - MNPR framework dans Maya

⁴⁶ Kyrianiadis, Jan Eric, et Henry Kang. « Image and Video Abstraction by Coherence-Enhancing Filtering ». *Computer Graphics Forum* 30, n° 2 (avril 2011): 593-602. <https://doi.org/10.1111/j.1467-8659.2011.01882.x>.

Dans le domaine de la 3D, il existe également des outils se basant sur les filtres pour créer des effets artistiques et aider les graphistes à exprimer un style plus personnel. Par exemple, le framework MNPR⁴⁷, permet de réaliser des rendus stylisés et expressifs dans le *viewport* du logiciel Maya. L'originalité de leur technique est qu'elle peut être utilisée pour générer un grand nombre de style différents en offrant plusieurs de niveaux de contrôle, quand la plupart des techniques de rendu expressif ont tendance à se focaliser sur un seul type de rendu spécifique qui peut être limitant pour les artistes. Un accent est donc mis sur l'interaction entre l'utilisateur artiste et le programme. Ce framework semble ainsi particulièrement approprié au domaine de la direction artistique. Les artistes ont plus d'ouvertures et peuvent plus facilement exprimer une vision ou une émotion spécifique grâce aux différents effets de style. Son utilisation consiste en différents niveaux de manipulation: il y a d'abord le contrôle global, qui s'applique à toute la scène, pour lequel il est possible d'utiliser un grand nombre d'effets prédéfinis. Il y a ensuite le contrôle sur le matériau qui permet de modifier certains effets de peinture tels que la dilution, ou la transparence en imitant des outils traditionnels tels que la peinture à l'huile ou l'aquarelle. Le *mapped control* s'utilise plus à un niveau local et le *proxy control* consiste à placer des objets 3D non visibles dans la scène qui permettent de créer certains effets à des endroits spécifiques, par exemple en ajoutant de la turbulence. Il utilise notamment des noises pour créer des effets de distorsion et de déformation. Pour générer ces filtres, l'outil s'inspire des méthodes traditionnelles en travaillant sur la densité et l'application du pigment sur différents types de papier, le dessin de contour des objets et le niveau de détail.

⁴⁷ Montesdeoca, Santiago E., Hock Soon Seah, Amir Semmo, Pierre Bénard, Romain Vergne, Joëlle Thollot, et Davide Benvenuti. « MNPR: A Framework for Real-Time Expressive Non-Photorealistic Rendering of 3D Computer Graphics ». In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering - Expressive '18*, 1-11. Victoria, British Columbia, Canada: ACM Press, 2018. <https://doi.org/10.1145/3229147.3229162>.

3.3.2 Rendu à base de marques ou SBR

La méthode *stroke based rendering* ou rendu à base de marques consiste à placer des touches de pinceau d'une certaine manière en partant d'une image ou une vidéo afin de générer une forme d'abstraction ou de stylisation de celle-ci.

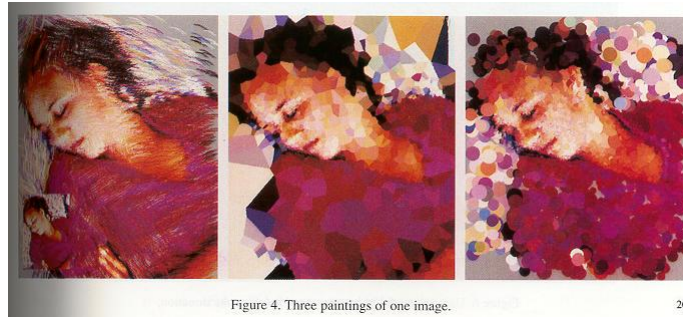


Illustration 46 - Paint by numbers - Paul Haeberli

Paul Haeberli, dans *Paint By Numbers: abstract Image Representations*⁴⁸ en 1990, a travaillé sur une forme d'automatisation de création coups de pinceau avec des paramètres inspirés des techniques traditionnelles. Cette méthode est interactive: l'utilisateur a la possibilité de contrôler la direction des coups de pinceau via la souris en se basant sur une image en entrée. Plus l'utilisateur peint lentement, plus les touches de pinceau sont petites. Il utilise un pinceau simulé qui dépend de plusieurs paramètres : position de départ, couleur, direction, épaisseur et forme. La couleur sera par exemple définie en échantillonnant l'image en entrée.

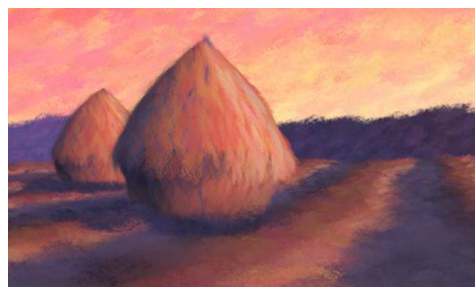


Illustration 47 - Stroke based rendering for 3D animation, Barbar Meier

⁴⁸ Haeberli, Paul. « Paint by Numbers: Abstract Image Representations ». In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, 207–214. SIGGRAPH '90. New York, NY, USA: ACM, 1990. <https://doi.org/10.1145/97879.97902>.

Barbara Meier, en 1996 publie un article sur rendu le expressif pour séquences d'images⁴⁹. Elle constate qu'il existe deux contraintes pour créer une animation non photoréaliste. D'abord, l'effet *shower door* ou rideau de douche. Ce problème apparaît lorsque les coups de pinceau ne suivent pas bien l'objet d'une image à l'autre. Cela donne l'impression que les éléments de l'image se déplacent simplement derrière un voile constitué de formes semi transparentes. Le second problème est le manque de cohérence et la discontinuité. Etant donné que les coups de pinceau doivent donner l'impression d'être posés de manière aléatoire, il faut trouver un moyen pour que cet aléatoire soit cohérent dans le temps dans le but d'éviter l'apparition et la disparition continuelle de touches de peinture pendant l'animation. Afin de résoudre ces problématiques, elle utilise un système de particules qui représente les coups de pinceau. L'orientation et la position sont déterminées en fonction de la géométrie, ainsi que des informations provenant des matériaux appliqués et de la lumière. Il y a donc une cohérence d'une image à l'autre, les coups de pinceau suivant le mouvement des objets 3D. Elle utilise une banque de formes qui serviront à définir la brosse du pinceau.



Illustration 48 - Rendu de peinture utilisant le raffinement par couches

Aaron Hertzmann, en 1998, utilise une technique de raffinement couche par couche: des touches de pinceau de tailles différentes s'empilent selon un ordre spécifique. Il explique ce procédé dans l'article *Painterly rendering with curved brush strokes of multiple sizes*⁵⁰.

⁴⁹ Meier, Barbara J. « Painterly Rendering for Animation ». *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96*. Consulté le 6 mai 2019. https://www.academia.edu/153902/Painterly_rendering_for_animation.

⁵⁰ Hertzmann, Aaron. « Painterly Rendering with Curved Brush Strokes of Multiple Sizes ». In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, 453-60. Not Known: ACM Press, 1998. <https://doi.org/10.1145/280814.280951>.

D'abord, des formes larges sont placées de manière plus grossière pour remplir l'image, puis des touches de plus en plus fines de peinture sont appliquées l'une à la suite de l'autre sur des zones définies. En effet, en peinture, on peut constater que les artistes tendent à peindre certaines zones de manière imprécise pour concentrer les détails sur des éléments plus importants : par exemple, ce qui est plus proche aura souvent des touches de pinceau plus petites et précises que ce qui est dans le fond de l'image. On verra en deuxième partie que cette gestion du détail est importante car elle donne une intention à l'image et permet à l'artiste de choisir ce qu'il veut mettre en valeur dans la scène.



Illustration 49 Réflexion - Yoshimichi Tamura (2012)

Le studio les films du Poisson Rouge a créé un logiciel M.O.E qui permet d'animer des rendus non réalistes en s'inspirant de techniques traditionnelles telles que la peinture, le pastel, le crayon... L'effet est produit à partir d'animations 2D ou 3D ou de prises de vue réelle. Il a été utilisé pour la première fois sur le court-métrage *Réflexion*.

Récemment, un outil nommé OverCoat⁵¹ a été développé par Disney dans le but de permettre à des graphistes de peindre de manière interactive sur des objets en trois dimensions. Cette fois-ci, les coups de pinceau sont placés dans la scène en trois dimensions et ne sont plus appliqués sur une surface plane comme dans la technique interactive de Haeberli par exemple. Bien que l'utilisateur peigne sur une vue en deux dimensions, l'outil utilise un champ scalaire basé sur une géométrie donnée en entrée appelée proxy pour déterminer la direction des touches de peinture dans l'espace en trois dimensions. L'utilisateur travaille avec des calques et peut définir lui même la couleur à appliquer ainsi que la forme.

⁵¹ Schmid, Johannes, Martin Sebastian Senn, Markus Gross, et Robert W Sumner. « OverCoat: An Implicit Canvas for 3D Painting », s. d., 9.

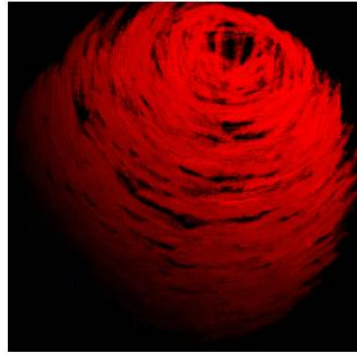


Illustration 50 - Rendu de peinture par ray tracing

Certaines personnes ont même créé des ray tracer à effet de peinture. C'est le cas de Markus Meister, Caroline Dullof et Remco Chang qui ont implémenté un outil de raytracing basé sur la pose de touches de peinture⁵². Lorsqu'un rayon atteint une géométrie de la scène, au lieu de créer un pixel, l'algorithme place un coup de pinceau et l'oriente en fonction de la normale de la géométrie.



Illustration 51 - 11-11 Memories Retold (2018)

Les outils de SBR ont parfois été utilisés dans le domaine du cinéma d'animation et du jeu vidéo comme par exemple dans le jeu *11-11 Memories Retold* portant sur la Première Guerre mondiale. Pour réaliser cet effet, les artistes ont eu la possibilité de personnaliser les coups de pinceau dans les différentes parties de l'image qui est séparée en plusieurs couches.

⁵² « Painterly Renderer Ray Tracer ». Consulté le 28 mars 2019.

<http://www.cs.tufts.edu/~remco/older-stuff/painterly.html>.

3.3.3 Transfert de style par deep learning

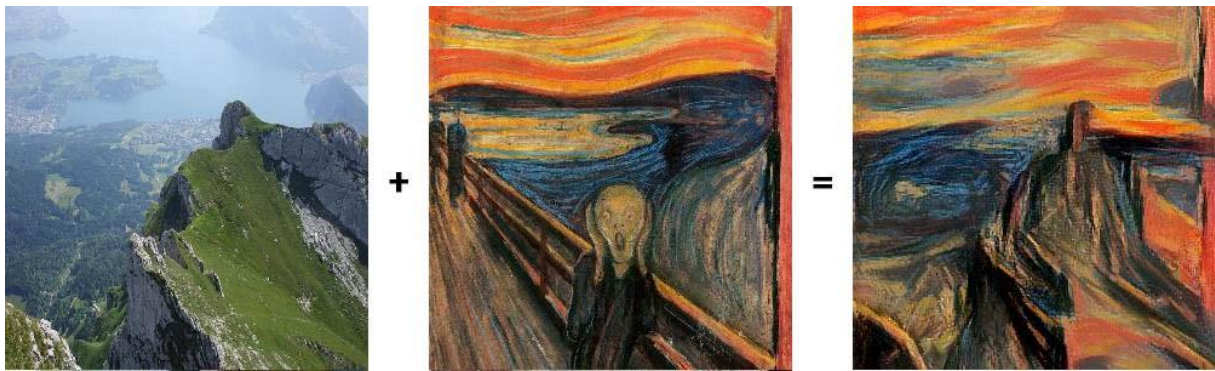


Illustration 52 - Transfert de style

Certaines applications qui ont été développées ces dernières années utilisent l'intelligence artificielle dans le but de transférer un style d'une image l'autre. Cette méthode se base sur les CNNs ou convolutional neural networks qui sont généralement utilisés en la reconnaissance faciale. Pour cela, on utilise le deep learning qui consiste à faire travailler un réseau avec différentes couches de neurones destinées à la reconnaissance d'éléments spécifiques tels que des motifs, couleurs, ou contours... Tübingen et Houston se servent de cette technique pour extraire des informations de texture et ensuite les appliquer sur d'autres images. Aujourd'hui, cette technique peut également être adaptée aux vidéos. Certaines méthodes se servent de l'optical flow est pour suivre le mouvement au sein de l'image et stabiliser l'effet de style. Il existe également des méthodes fonctionnant en temps réel, telles que celle développée à l'université Tsinghua décrite dans l'article *Real-Time neural style transfer for videos*⁵³ qui consiste à entraîner les réseaux de neurone pour que ceux-ci maintiennent une cohérence temporelle d'une image à l'autre.

3.3.4 Méthode par création de shell ou coquille

Pour faire suite à l'outil OverCoat, Disney a développé une technique de rendu à effet de peinture adaptée cette fois-ci au temps réel qui sera décrite dans *Fin Textures for Real-*

⁵³ Huang, Haozhi, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, et Wei Liu. « Real-Time Neural Style Transfer for Videos ». In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7044-52. Honolulu, HI: IEEE, 2017.

<https://doi.org/10.1109/CVPR.2017.745>.

*Time Painterly Aesthetics*⁵⁴ paru en 2015 au Siggraph. Ils utilisent une technique inspirée des « shell textures » en générant, à partir de l'extrusion d'un proxy 3D dans le sens des normales des faces, un nouveau *mesh* appelé « fin mesh ». Son offset est paramétrable et celui-ci est texturé au niveau des contours du modèle. Il est possible d'ajouter plusieurs couches successives.

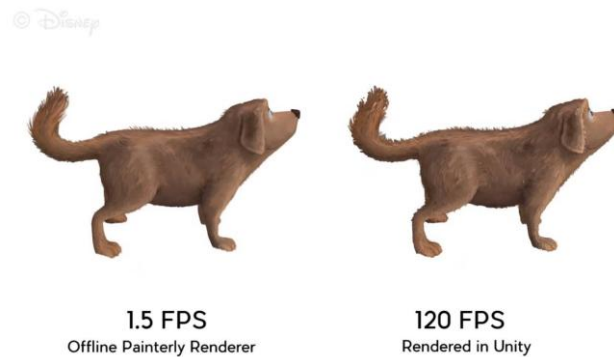


Illustration 53 - Méthode de rendu par shell en temps réel dans Unity

Une autre méthode basée sur la création de *shells* a également été créée pour le moteur de jeu Unreal Engine. Il s'agit de disposer plusieurs couches de polygones autour d'un modèle et qui sont peintures avec des textures de crayon ou de peinture. Ensuite, il faut en sorte de ne laisser visible que la partie de la couche proche du contour de l'objet en diminuant de manière progressive l'épaisseur de ce contour. On a ainsi l'impression que des coups de crayon dépassent du personnage, ce qui lui donne un aspect esquissé. Un effet de bloom est ajouté pour rendre le tour plus lumineux. On peut ensuite renforcer ce contour en extrayant les parties colorées du contour et les parties lumineuses et ensuite en modifiant leur aspect en *post processing*⁵⁵.

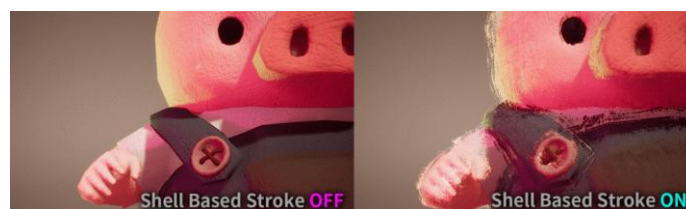


Illustration 54 - Effet de crayon par couches dans Unreal Engine

⁵⁴ Imhof, Nicolas, Antoine Milliez, Flurin Jenal, René Bauer, Markus Gross, et Robert W. Sumner. « Fin Textures for Real-Time Painterly Aesthetics ». In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games - SA '15*, 227-35. Paris, France: ACM Press, 2015. <https://doi.org/10.1145/2822013.2822021>.

⁵⁵ « ゲームの花園 ». Consulté le 14 avril 2019. <https://aqu.hatenablog.com/>.

3.3.5 Rendus expressifs basés sur la lumière

Seif El Nasr, dans son article *Temporal Dynamic Expressive Lighting Engine*⁵⁶, s'inspire de techniques de théâtre et de cinéma ainsi que de théories du design pour les appliquer au jeu vidéo. Elle utilise un système d'éclairage dynamique basé sur des motifs. Son travail porte sur la manière de créer des effets émotionnels dans le jeu vidéo en passant par des techniques non réalistes. Elle étudie notamment le fonctionnement de l'œil par rapport aux couleurs et les réponses émotionnelles induites par celles-ci. Par exemple, lorsqu'il est exposé à un contraste important, l'œil essaie de corriger les couleurs pour obtenir un blanc et compenser les fortes différences de teinte. Par exemple, si on projette du rouge, l'œil va essayer de le compenser avec du cyan. Cela engendre une fatigue chez le spectateur qui augmente son niveau de stress et crée ainsi une tension uniquement en passant par l'utilisation de couleurs. Ainsi, la saturation et le contraste participent à l'élaboration d'une excitation et peuvent être utilisés à des fins émotionnelles, qui auront un impact concernant la réception de l'œuvre.

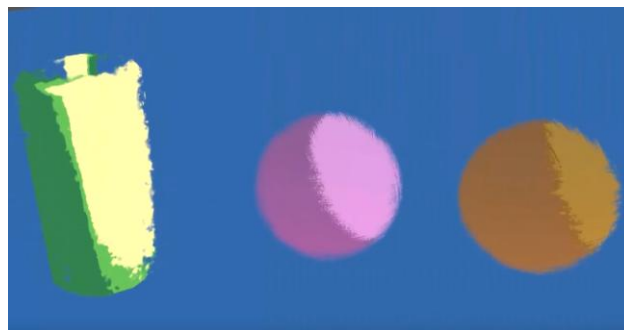


Illustration 55 - Technique utilisée pour Pearl

*Pearl*⁵⁷ est un film en 360° sorti en 2016. C'est le premier film en réalité virtuelle à avoir été nommé à l'académie des Awards. Tuna Bora, qui était chargée des concepts graphiques, a voulu créer un style se rapprochant du « polygon impressionism »⁵⁸. C'est Cassidy Curtis, un chercheur qui avait déjà travaillé sur le rendu non photoréaliste dans les années 1990, qui a participé à la recherche de développement du projet. Il fallait avoir un niveau de détail qui varie en fonction de la distance par rapport à la caméra. Ils ont effectué des recherches pour trouver un moyen de créer un rendu sans contours, constitué

⁵⁶ El-Nasr, Magy Seif, Simon Niedenthal, Igor Knez, Priya Almeida, et Joseph A. Zupko. « Dynamic Lighting for Tension in Games ». *Game Studies* 7 (2007).

⁵⁷ (2017), *Pearl*.

⁵⁸ Gnomon. *The Making of PEARL with Academy Award-Winning Patrick Osborne*. Consulté le 26 mars 2019. <https://www.youtube.com/watch?v=zZSIhwxLqu0>.

uniquement de formes colorées. Ils ont travaillé sur une manière de rendre les bords des formes vibrants et esquissés et de contrôler la forme de la lumière projetée sur les personnages. Les lumières sont en fait des géométries qui entrent en intersection avec le modèle. Pour cela, ils rendent trois passes, l'une est d'une couleur lumineuse, la deuxième est de couleur sombre et la troisième est un masque défini manuellement qui indique sur quelles zones seront placées les lumières. Cela a permis de générer cet effet polygonal.

Cette présentation n'est pas exhaustive et il existe encore un très grand nombre de techniques de rendu expressifs qui ne sont pas classées ici. Cependant, elle nous a permis d'appréhender un peu mieux les enjeux de la création de ce type de rendu en posant certaines problématiques telles que celles de la cohérence temporelle, de l'optimisation, de la gestion du style et du rapport entre l'outil et l'artiste. A présent nous allons pouvoir nous intéresser aux différentes expérimentations que j'ai effectuées lors des périodes de recherche.

II Expérimentations et créations d'outils de rendu expressif aux effets variés, s'inspirant de techniques traditionnelles

En première partie, nous avons étudié les différents besoins des artistes ainsi que la manière avec laquelle ceux-ci utilisent les différents médiums afin de s'exprimer artistiquement. Nous avons ainsi pu voir que le numérique pouvait proposer un grand nombre d'outils prévus à cet effet dans le domaine du rendu non photoréaliste et du rendu expressif. Il s'agira dans cette deuxième partie de présenter les différentes recherches et expérimentations effectuées au cours de l'année, ainsi que les outils développés à cet effet.

Pour réaliser ces expérimentations, j'ai choisi d'utiliser Unity, un moteur de jeu 3D créé d'abord sur Mac au début des années 2000 au Danemark par David Helgason, Joachim Ante et Nicholas Francis. Il permet aujourd'hui de développer pour un nombre varié de plateformes, allant du web (WebGL) aux applications mobiles et ordinateurs de capacités diverses en passant par les consoles de jeu. Il est possible de créer ce qu'on appelle des composants (components) grâce à des scripts en langage CSharp ou javascript qui servent à attribuer des comportements aux différents éléments constituant la scène, que l'on appelle *game objects*. Cette application est rapide à prendre en main et est particulièrement adaptée au prototypage. Depuis quelques années, Unity a commencé à être également de plus en plus utilisé pour le film d'animation, avec par exemple le court-métrage de Disney *Baymax Dreams of Too Many Baymaxes*⁵⁹ sorti en septembre 2018, ou *Sonder*⁶⁰ de Soba Productions, paru aussi en 2018. L'utilisation d'Unity pour le film est liée à plusieurs innovations, dont notamment l'amélioration de la qualité graphique grâce par exemple au HDRP (*High Definition Render Pipeline*) et à la possibilité de créer son propre pipeline de rendu. Cela est aussi dû au fait que Unity gère très bien lien vers les logiciels d'Autodesk comme Maya ou 3Ds max. Enfin, l'apparition de fonctionnalités telles que la *timeline*, un séquenceur qui permet de créer et d'organiser des clips, d'ajouter des pistes audio, ainsi que le cinemachine qui sert à créer des *rigs* de caméra de manière simple et procédurale en simulant des techniques cinématographiques de prise de vue réelle ont également facilité

⁵⁹ (2018), *Baymax Dreams of Too Many Baymaxes*

⁶⁰ (2018), *Sonders*

l'utilisation d'Unity pour le film. Comme expliqué en première partie, le temps réel offre ainsi de multiples possibilités artistiques concernant la réalisation de films et de séries d'animation. J'ai choisi ce moteur spécifiquement car je le trouvais pratique et simple à exploiter pour du développement de *shaders* grâce notamment au langage déclaratif Shaderlab. Il existe un nombre important de ressources et de documentations permettant d'apprendre à programmer pour cette plateforme. Il est ainsi possible d'expérimenter et d'obtenir rapidement des résultats intéressants qui peuvent ensuite être mis à la disposition d'artistes.

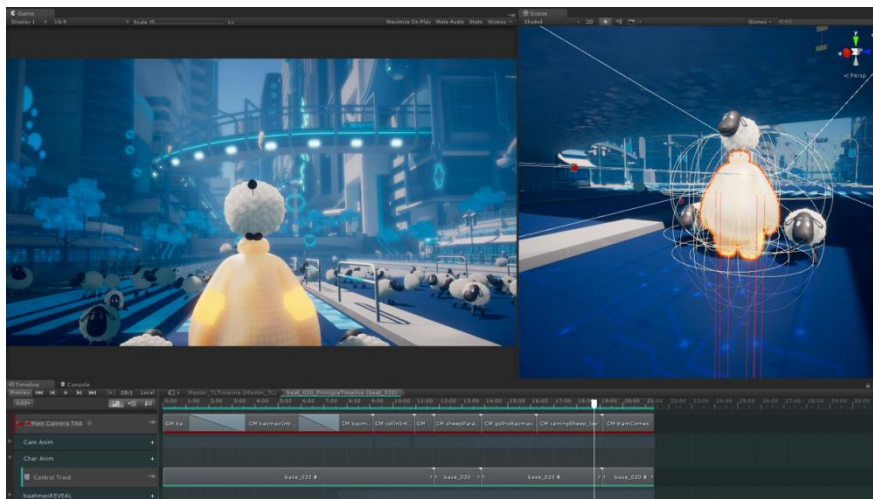


Illustration 56 - Baymaw Dreams dans Unity

1. Permettre l'expression artistique à travers divers matériaux virtuels grâce aux outils de traitement d'image

1.1 Généralités et principes du *post-processing* par traitement d'image en temps réel

1.1.1 Rôle et apports dans le domaine du temps réel

Le *post-processing*, dans le domaine de la 3D temps réel, permet d'apporter des ajouts et des modifications à l'image avant que celle-ci ne soit affichée sur l'écran. On appelle ces modifications des filtres ou effets. Pour cela, il faut récupérer l'image de rendu de la scène conservée dans un tampon pour lui appliquer un fragment *shader*. Le fragment *shader*, comme nous l'avons vu en première partie, est un petit programme exécuté en

parallèle qui peut prendre en entrée différentes valeurs telles que par exemple la position du fragment, ou encore des coordonnées de texture et qui retourne en sortie une couleur qui sera attribuée au pixel à la position courante. Son rôle est donc de renseigner sur la manière par laquelle il faudra rendre ces pixels. Lors de l'écriture du *shader*, on définit un certain nombre d'instructions qui devront être réalisées indépendamment sur chaque pixel de l'image. Ainsi, lorsqu'un processus, ou thread, est exécuté sur un pixel particulier, il est impossible de connaître les résultats des opérations des autres threads s'exécutant simultanément. Il n'est pas non plus possible de mémoriser les opérations effectuées précédemment puisqu'à chaque fois qu'un thread a terminé ses opérations, on lui attribue une nouvelle tâche, et ce indépendamment de ce qu'il faisait auparavant. Ce sont les deux principales difficultés liées à la programmation GPU.

Le *post-processing* est généralement utilisé pour générer certains effets utiles tels que de l'*ambient occlusion* ou de l'*anti aliasing* ou des corrections colorimétriques. Il est également possible d'apporter du réalisme avec par exemple du *motion blur* ou des aberrations chromatiques. Enfin, on peut utiliser cette procédure afin d'appliquer des effets plus artistiques comme l'ajout de bloom, de vignettage... Il faut bien avoir conscience que l'ensemble de ces calculs se fait à partir des informations 2D rendues de la scène. On va alors effectuer un traitement de l'image par différentes opérations.

1.1.2 Précisions et descriptions techniques: comment développe-t-on un shader dans Unity?

Unity nous propose différents templates de *shader*. Les *surface shaders*, par exemple, permettent de réaliser des *shaders* pouvant réagir à la lumière sans avoir à programmer tous les calculs de lumière. Nous n'en aurons pas besoin pour effectuer du *post-processing* puisqu'il s'agira uniquement d'exécuter du traitement d'image, ce qui ne nécessite pas d'opérations de *lighting*. On va plutôt utiliser un simple vertex/fragment *shader*, qui dans Unity correspond à ce qu'on appelle *unlit shader*, pour non éclairé. Les *shaders* sont écrits dans deux langages similaires: Cg et HLSL. Il s'agit de langages de programmation haut niveau destinés spécifiquement à la programmation de *shader*, la syntaxe ressemblant à celle du langage C. HLSL, pour High Level Shader Language est le langage destinée à l'API

DirectX de microsoft et Cg pour C for Graphics est le langage de programmation de shader réalisé par le constructeur de cartes graphiques NVIDIA.

Un shader dans Unity se compose principalement de deux parties. Il y a d'abord les propriétés, ou *properties*, qui servent à déclarer différents types de variables (floats, textures 2D, vecteurs, couleurs ...) qui seront des paramètres modifiables dans l'interface utilisateur que l'on appellera *inspector*. Par exemple, lorsque l'on crée un matériau utilisant un *shader* spécifique, les propriétés attribuées à ce matériau agiront sur celui-ci uniquement. Il sera possible de créer d'autres matériaux utilisant le même *shader* mais avec des propriétés différentes. La deuxième partie est le *subshader* qui contient une déclaration des propriétés afin de les rendre utilisables par le programme ainsi que le programme même du *shader* qui déterminera son comportement et les opérations à effectuer. Il peut y avoir plusieurs *subshaders*, cela permet de déclarer différents *shaders* dont un seul sera exécuté en fonction de ce que peut supporter la machine, et ainsi avoir une application pouvant tourner sur un grand nombre de plateformes différentes.

1.1.3 Comment réaliser un effet de post processing via Unity?

Pour appliquer un effet de *post-processing* dans Unity, une des méthodes les plus simples est de créer un script qui sera assigné à la caméra principale. On utilise ensuite la fonction *Blit* qui permet de copier une texture dans une *render texture* et de lui appliquer un *shader*. Cette fonction doit s'exécuter dans la méthode *OnRenderImage* de la classe *Camera*, qui est appelée à chaque fois que celle-ci a terminé le rendu d'une image. Cela permet de récupérer la texture en entrée, c'est à dire l'image rendue sans opérations de *post-processing*, et de passer un matériau exécutant les transformations d'images que l'on souhaite obtenir ainsi qu'une *render texture* qui contiendra l'image ainsi transformée qui sera affichée à l'écran.

1.2 Développement de la palette de matériaux

J'ai commencé par effectuer un certain nombre d'expérimentations pour générer des effets de matière sur des images en *post-processing* en temps réel afin de découvrir et me familiariser avec la programmation de *shaders* dans Unity. Je me suis servie d'une partie de ces expérimentations pour ensuite créer un outil de rendu expressif simple d'utilisation. Le

but de cet outil est d'offrir un certain nombre de possibilités créatives aux artistes en leur permettant de générer et d'hybrider plusieurs styles visuels qui prendront effet directement sur leurs images. L'enjeu est ainsi de mettre à leur disposition une palette de matériaux dont il sera possible de modifier de manière procédurale les paramètres afin de les appliquer en temps réel et de constituer une image expressive et personnelle à partir de ceux-ci. Pour cela, je me suis inspirée de divers mouvements artistiques étudiés lors du premier chapitre. J'ai ainsi tenté de reprendre certaines caractéristiques de l'expressionnisme telles que l'emploi de distorsions, les imprécisions et la possibilité de délimiter ou non les objets par des contours comme on peut le voir dans les œuvres de Marianne Von Werefkin. Je me suis également basée sur le fauvisme et notamment certains tableaux de nature d'André Derain et Maurice de Vlaminck pour l'aspect vibrant des couleurs, et la simplification du dessin par touches de peinture. J'ai ainsi souhaité accorder une importance aux divers matériaux employés pour que l'utilisateur puisse reproduire et créer de nouveaux effets afin d'exprimer une vision ou une émotion via un outil de rendu temps réel, de la même manière que le faisaient les peintres du début du XXe siècle sur les médiums traditionnels.



Illustration 57 - Tableau de Marianne Von Werefkin

1.2.1 Touches de peinture

Lors de mes expérimentations, j'ai d'abord commencé par chercher à créer un effet pictural qui puisse s'appliquer globalement sur le rendu final en transformant les pixels pour faire comme s'ils avaient été tracés par des coups de pinceau. Pour cela, je me suis tournée vers le filtre de Kuwahara. C'est un algorithme de traitement d'image permettant de flouter une image en préservant les contours. Il est utilisé généralement comme filtre permettant

de réduire le bruit, mais peut également servir à créer un rendu pictural en transformant les pixel afin de donner un aspect « touches de peinture ». Alors qu'un algorithme de flou classique se contente pour chaque pixel de récupérer les valeurs de ceux situés autour et de lui assigner la moyenne de ces pixels, le filtre de Kuwahara fonctionne avec des *kernels* qui sont les 4 régions situées autour du pixel d'une taille pouvant être paramétrée. On calcule d'abord la variance de chaque *kernel*, puis on fait la moyenne des pixels présents dans la *kernel* ayant la plus faible variance et on l'attribue au pixel courant. Les contours sont ainsi conservés puisqu'on retiendra toujours la moyenne des valeurs ayant le moins de différence avec celle du pixel en cours de traitement, ceux dont la variance est plus élevée ne seront pas sélectionnés. Il m'a également fallu adapter cet algorithme pour qu'il fonctionne sur des images en couleur.

$$s^2 = \frac{\sum (X - \bar{X})^2}{N-1}$$

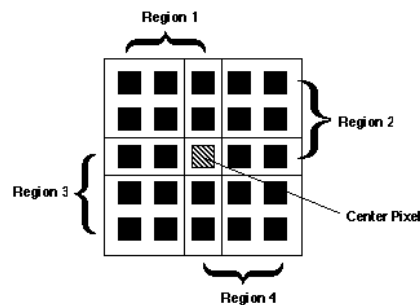


Illustration 58 - Fonctionnement du filtre de Kuwahara

Après avoir implémenté l'effet dans Unity, on se retrouve avec plein de petites zones d'une couleur similaire, ce qui donne l'impression que ces zones ont été dessinées par touches de peinture dont la taille peut être augmentée en fonction du nombre d'itérations. J'ai cependant constaté que les différentes touches sont toutes de forme rectangulaires et sont orientées dans le même sens. Ce résultat n'est pas idéal donne un effet artificiel. En effet, dans la peinture, les artistes ont tendance à donner une intention à leurs œuvres en dirigeant les coups de pinceau d'une certaine manière pour, par exemple, évoquer le mouvement ou le volume. Il aurait été plus intéressant que les touches de couleur soient orientées de manière souligner la forme des objets. Afin d'y apporter une amélioration, je me suis basée sur un article qui parle d'une utilisation modifiée de ce filtre sur Unreal

Engine⁶¹, et je l'ai mise en place dans Unity. Cette technique consiste à utiliser le filtre de Sobel pour détecter les contours dans une image et par exemple obtenir l'orientation correspondant à la direction du contour à un pixel donné. L'algorithme calcule le gradient de l'intensité de chaque pixel et permet d'obtenir une direction vers l'endroit où la différence de niveau d'intensité des pixels est la plus forte. On récupère ainsi un angle qui nous laisse effectuer une rotation de la région de pixels traité afin de suivre le sens des contours. Le fonctionnement du filtre de Sobel sera explicité plus en détail au cours de la deuxième expérimentation. Grâce à cet ajout, on obtient ainsi des coups de pinceau qui vont dans le sens du contour des objets. Cet effet est particulièrement visible sur l'herbe et sur les feuilles. J'ai également tenté de varier quelque peu la taille et la forme des coups de pinceau en modifiant la taille de la zone d'échantillonnage pour chaque pixel. Afin de pouvoir gérer le niveau d'abstraction, j'ai également fait en sorte que l'on puisse diminuer la résolution de l'image de base. On aura ainsi l'impression d'avoir non pas des petites touches de peinture mais de grands aplats qui envahissent la scène et rappellent quelque peu la peinture fauve.

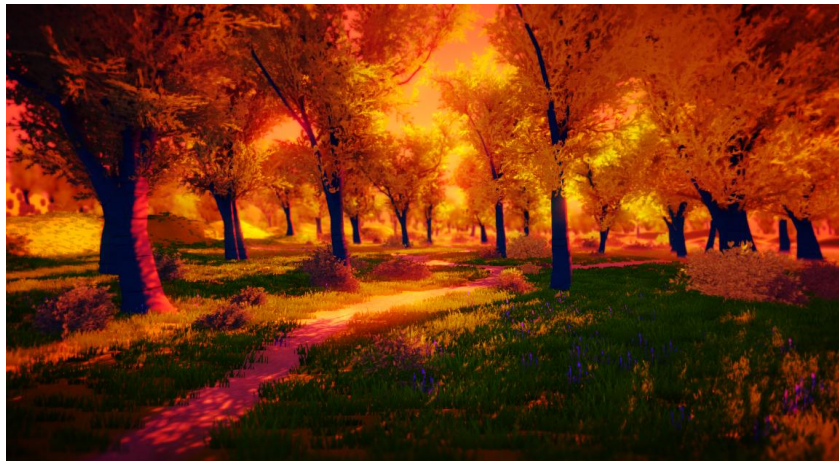


Illustration 59 - Image sans filtre - Asset provenant de l'asset store de Unity

⁶¹ « Unreal Engine 4 Paint Filter Tutorial ». raywenderlich.com. Consulté le 6 décembre 2018.
<https://www.raywenderlich.com/100-unreal-engine-4-paint-filter-tutorial>.



Illustration 60 - Kuwahara avec filtre sans direction



Illustration 61 - Kuwahara avec directions

1.2.2 Aquarelle

J'ai également effectué des expérimentations sur de l'aquarelle dans le but de permettre aux artistes de travailler avec des couleurs plus délavées et d'accorder plus d'importance au support virtuel de l'image, l'aquarelle revêtant un lien avec le papier sur lequel elle est appliquée.

J'ai d'abord essayé de générer ce type de rendu via un logiciel de compositing, After Effects, afin d'avoir une idée des opérations à effectuer pour obtenir cet effet via le *shader* en temps réel. L'idée de base était d'appliquer un traitement différent à la couleur et aux ombres, puis de tout combiner. Ici, j'ai récupéré le rendu d'un film court réalisé au cours de mes études et j'ai sorti une passe de couleur et d'*ambient occlusion*. Sur la passe ayant les aplats de couleur, j'ai mis des effets de flou et de déformations pour obtenir des débordements de couleur. J'ai également effectué des transformations de couleurs pour les rendre plus pâles et diffuses. Pour le calque d'occlusion, j'ai surtout travaillé avec des déformations et des turbulences à un niveau local puis plus global pour donner cet aspect

dilué, principalement sur les contours de l'objet et dans les zones d'ombre. Ensuite, j'ai dupliqué ce calque d'ombres et ai modifié certains paramètres en utilisant l'algorithme de détection de contours d'After Effects pour détecter les bords des objets. Je me suis servie de courbes pour faire en sorte que le contour soit plus foncé sur les bords qu'à l'intérieur des taches de peinture, comme c'est le cas avec l'aquarelle. J'ai enfin ajouté une texture de papier aquarelle pour donner une consistance à l'image et faire comme si la peinture reposait sur un support.

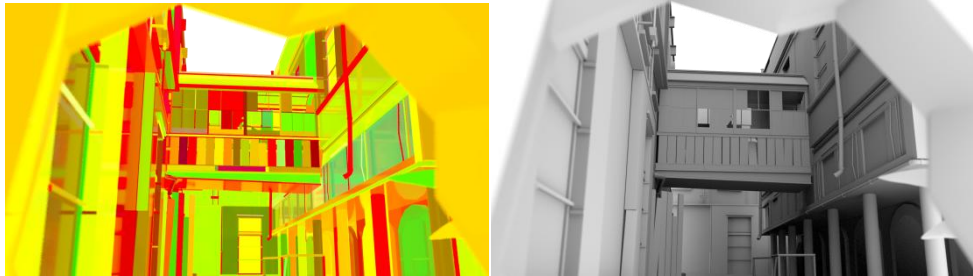


Illustration 62 - Effet aquarelle - Passe de couleurs

Illustration 63 - Effet aquarelle - Passe d'occlusion



Illustration 64 - Effet aquarelle - Rendu un style réaliste provenant d'un ancien projet intensif



Illustration 65 - Effet aquarelle - Rendu après traitement de l'image

Pour réaliser cet effet sur Unity, j'ai du créer un *shader* constitué de plusieurs passes. La première passe effectue un déplacement de pixels paramétrable qui permet de distordre l'image pour mimer l'étalement de l'aquarelle sur une couche de papier. Ce déplacement dépend d'une texture en entrée et de paramètres tels que la quantité de déplacement qui peuvent être modifiés par l'utilisateur. J'ai ensuite une deuxième passe qui effectue deux étapes. Pour traiter les contours, elle floute les couleurs de l'image et récupère la couleur déplacée par la première passe. Elle divise ensuite les pixels de la couleur déplacée par la ceux de la couleur floutée et ajoute du contraste afin de la renforcer et recréer cette agglomération de pigment que l'on voit couramment apparaître sur les bords en utilisant de l'aquarelle. Pour le support, elle prend en entrée un *perlin noise* et réalise une interpolation linéaire entre le résultat et la texture de papier placée en entrée. Diverses corrections colorimétriques sont également appliquées en fonction de paramètres donnés par l'utilisateur comme le contraste ou la luminosité. A la fin, on combine ces couleurs pour obtenir un rendu ressemblant à de la peinture aquarelle.

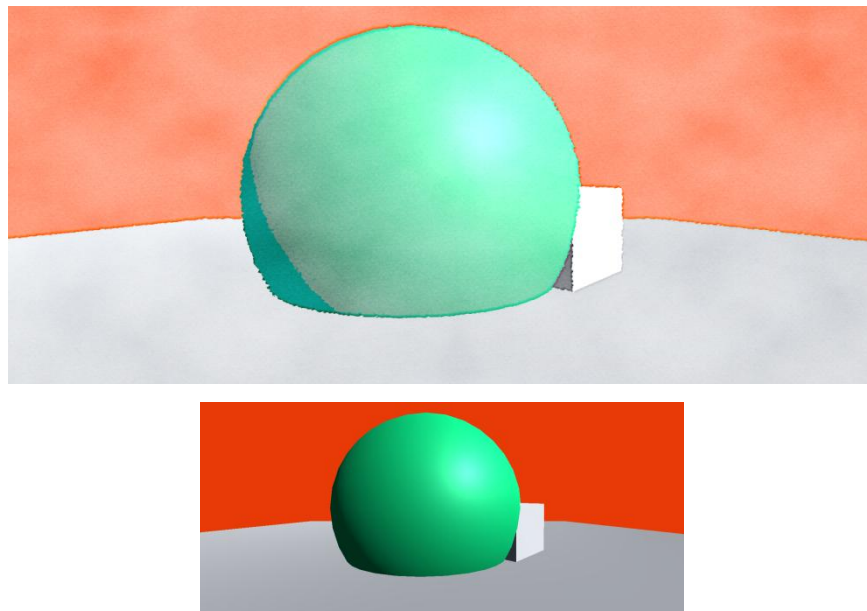


Illustration 66 - Effet aquarelle - Primitives dans Unity

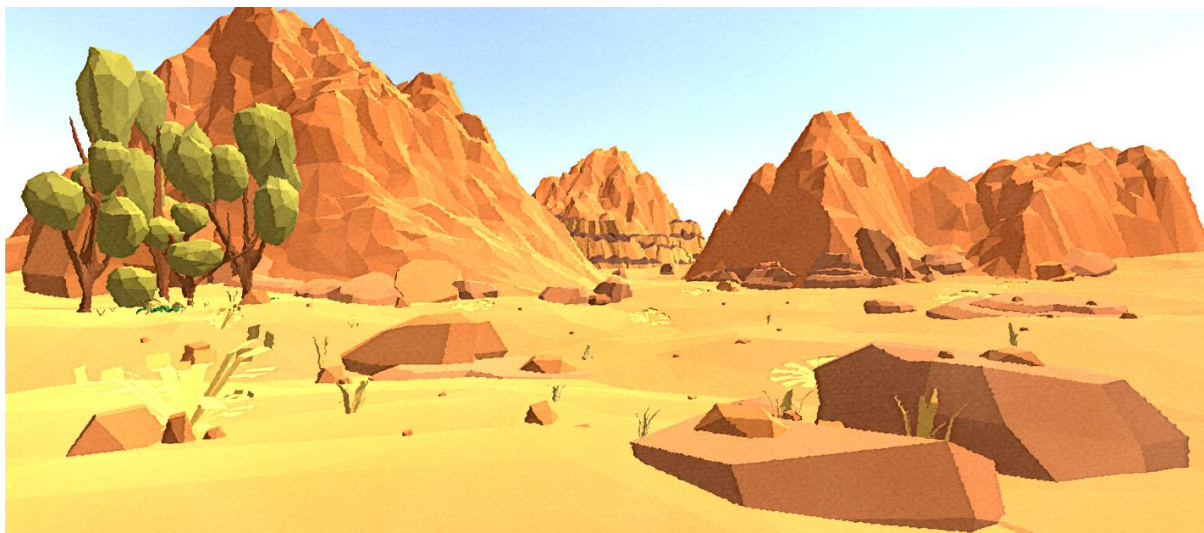


Illustration 67 - Effet aquarelle sur Canyons provenant de l'asset store d'Unity

1.2.3 Ajout de contour, crayonné

Ensuite, j'ai essayé d'implémenter des *shaders* pour pouvoir ajouter des effets de crayonné, d'encre et de contour autour des objets 3D. J'ai testé différentes méthodes en m'inspirant de *shaders* présents sur les plateformes shadertoy, notamment ceux de Daniel Taylor⁶² et du site Shader Laboratory⁶³. Je voulais également qu'il y ait un aspect brouillon ou inachevé dans le rendu final. Le but n'était pas de créer par exemple un contour parfait comme c'est souvent le cas avec certaines techniques telles que le *cel-shading*. Le *shader* résultant consiste à détecter les contours grâce aux gradients de l'image et à placer des traits à ces endroits en variant leur orientation. Les contours et les ombres sont déformées grâce à une texture de noise pour leur donner un aspect tremblant et gérer le degré d'imprécision. Le but ici est de générer de l'erreur grâce à des fonctions aléatoire. Il est possible aussi de démultiplier le nombre de coups de crayon et de les faire plus ou moins onduler avec des fonctions sinus. L'effet est ensuite mélangé à l'image finale avec une

⁶² « culdevu - Shadertoy BETA ». Consulté le 30 novembre 2018.

<https://www.shadertoy.com/user/culdevu>.

⁶³ « Shaders in Unity ». Consulté le 15 novembre 2018. <http://www.shaderslab.com/>.

interpolation linéaire. Il est également possible d'intégrer une texture dans le fond de l'image pour par exemple créer un effet papier.

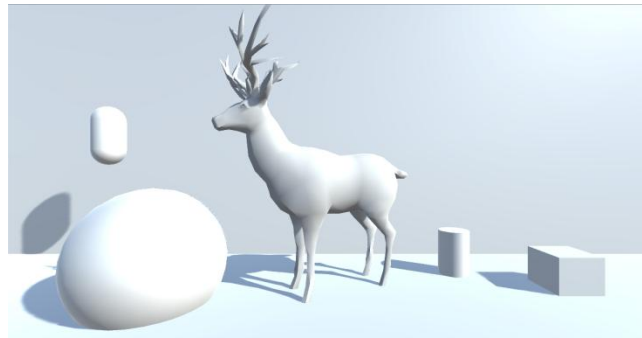
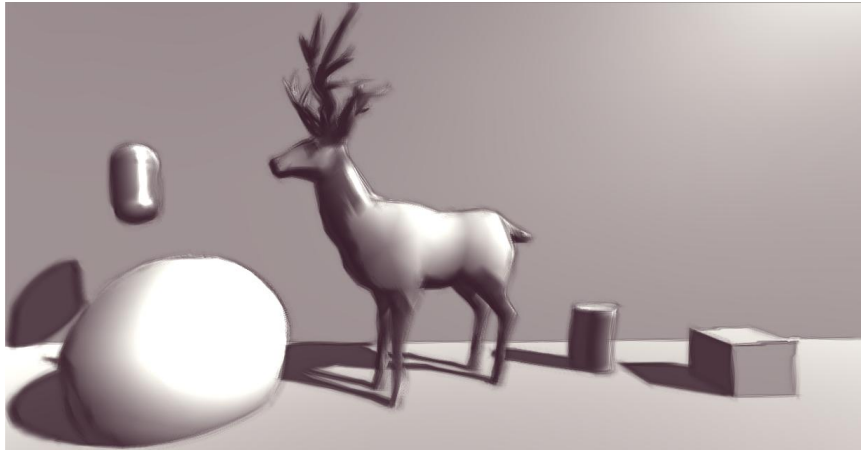


Illustration 68 - Effet sketch dans Unity - Assets provenant de l'Asset Store

1.2.4 Turbulences

En plus de ces diverses expérimentations sur les matériaux et les supports, je me suis également inspirée de la peinture expressionniste pour implémenter des opérations permettant de déformer les pixels de l'image en fonction du temps et d'effectuer ainsi des distorsions. Le but est d'exploiter la dimension temporelle pour engendrer du mouvement et des vibrations dans l'image. L'artiste peut ainsi décider par exemple d'appliquer des turbulences à certains endroits ou à certains moment de la scène pour générer une ambiance ou exprimer une émotion particulière. Ces opérations sont rendues possibles par l'utilisation de flow textures: des textures contenant dans chaque pixel des informations de vecteurs 2D ou 3D qui servent à indiquer des directions de déplacement. Nous les étudierons plus en détail lors de la deuxième expérimentation.

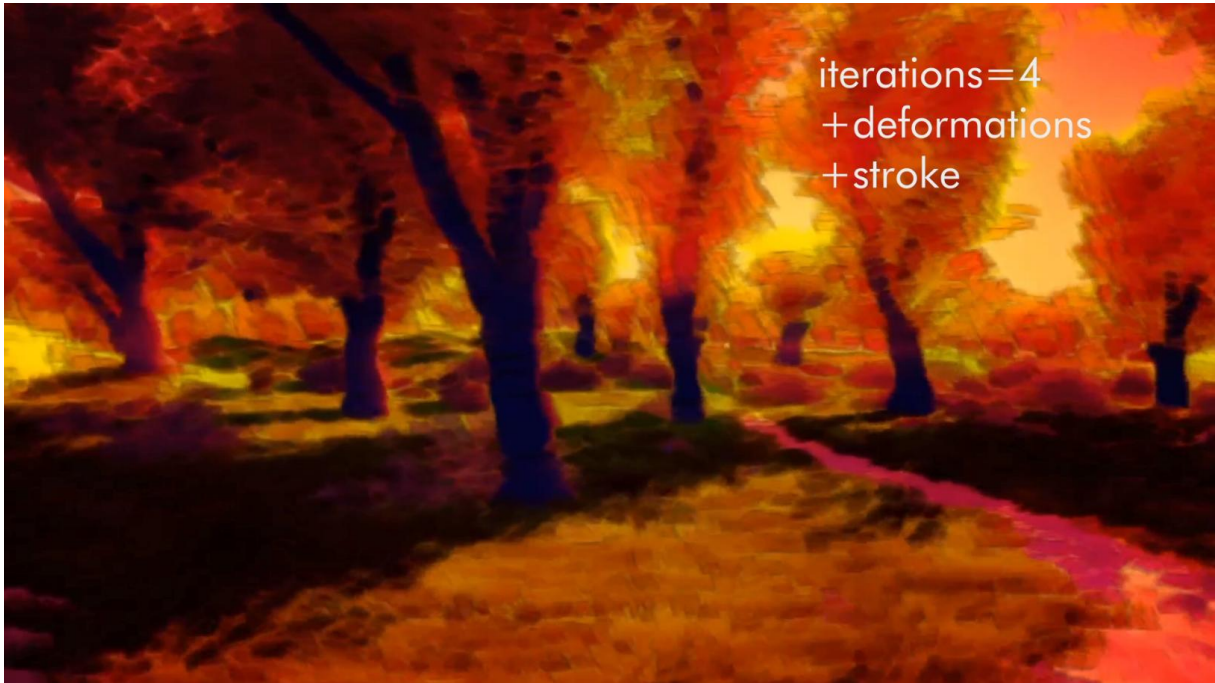


Illustration 69 - Combinaison de plusieurs effets, stylisation et ajout de distorsions

1.3 Retour d'expérience et efficacité

1.3.1 Apports de la méthode: un outil léger qui aide l'artiste à générer une variété de styles visuels expressifs

L'outil final permet facilement de combiner et d'hybrider un certain nombre d'effets. Les artistes peuvent ainsi travailler la couleur de leur scène tout en employant différentes sortes de matériaux. Ils ont la possibilité de gérer des niveaux d'abstraction en induisant une stylisation plus ou moins élevée. Grâce à ce type de rendu, on sent bien la présence d'une forme matière virtuelle. Les jeux sur les couleurs de type fauve apportent une impression éclatante et vive qui peut être exploitée à des fins expressives. On a également pu, grâce au numérique, exploiter le temps pour générer du mouvement dans l'image avec les distorsions et turbulences afin de permettre aux artistes d'apporter une dimension plus organique à leur scène.

1.3.2 Les limites de l'outil: un manque de contrôle au niveau local

Certains points peuvent être améliorés, notamment, le fait que l'effet s'applique sur l'image globale. Il pourrait être plus intéressant d'utiliser les informations de profondeur pour gérer les paramètres en fonction de celle-ci. Par exemple, en augmentant la taille des coups de pinceau avec la distance de l'objet par rapport à la caméra. Il serait aussi

avantageux d'accorder plus d'importance aux ombres et aux lumières, par exemple en générant des dilutions ou distorsions de celles-ci de manière dynamique pour que l'on sente plus la présence et la réaction des pigments sur le support virtuel. On pourrait aussi ajouter plus de rugosité et essayer de donner du relief à l'image, comme on peut le voir dans certains tableaux dans lesquels les couches de matière ressortent et apportent presque une troisième dimension à la peinture.

1.3.3 Appréciations sur la méthode employée et définition des nouveaux enjeux

Ainsi, on a pu voir grâce à ces expérimentations que la technique de rendu en *post-processing* dispose de différents avantages: les images sont rapides à calculer, les opérations que l'on peut appliquer sont nombreuses et variées et ne sont pas très difficiles à implémenter. Cependant, elle ne permet pas à l'artiste d'avoir un contrôle direct sur l'apparence et la forme des coups de crayon ou des touches de peinture. Il ne peut pas choisir précisément d'ajouter plus ou moins d'éléments sur certaines zones de l'image. Nous allons ainsi nous tourner vers d'autres méthodes de rendu expressif qui permettent de gérer d'autres niveaux de contrôle et tenter de les adapter au temps réel.

2. Algorithmes pour générer des peintures de manière procédurale

2.1 Choix de la technique

2.1.1 Principe du rendu basé sur strokes

Le rendu à base de marques ou *stroke based rendering*, regroupe un ensemble de techniques de rendu permettant d'abstraire ou de simplifier une image en posant de manière automatique des formes dont le comportement, l'apparence et la disposition sont définis via un algorithme. Aaron Hertzman, dans son article *A Survey of Stroke Based Rendering*⁶⁴, la définit comme « une approche automatique pour créer des images non

⁶⁴ Hertzmann, A. « A Survey of Stroke-Based Rendering ». *IEEE Computer Graphics and Applications* 23, n° 4 (juillet 2003): 70-81. <https://doi.org/10.1109/MCG.2003.1210867>.

photoréalistes en plaçant des éléments discrets tels que des coups de peinture ou des pointillés ». On l'utilise souvent pour imiter des techniques de dessin ou de peinture traditionnelles ou encore pour rendre une image plus abstraite en la résumant à un ensemble de formes. Nous avons vu en première partie plusieurs techniques utilisant ce procédé, ces techniques étant destinées à différents supports: images fixes, vidéos ou animations 3D. Le but de cette expérimentation est de réaliser un outil à la fois intuitif et modulable, pour permettre aux artistes de créer des rendus en se basant sur l'information de style qu'il exposera en entrée du programme. Etant donné que notre sujet porte sur le film d'animation temps réel, il serait également intéressant d'animer et de pouvoir faire évoluer ces styles au cours du temps.

Globalement, les algorithmes de SBR consistent à placer des entités que l'on appelle « strokes » selon des règles prédéfinies en se basant sur une ou plusieurs images existantes. Ces *strokes* ont plusieurs paramètres, comme par exemple une couleur ou une orientation, qui peuvent varier selon les instructions données en entrée par l'utilisateur ainsi que les informations récupérées sur l'image même. Il y a donc tout un travail à effectuer concernant l'extraction des données d'une image, vidéo ou scène 3D pour ensuite interpréter ces informations et les utiliser d'une manière spécifique, généralement décrite par l'utilisateur. Le but est d'apporter une intention définie en fonction de certains éléments de l'image. Par exemple, il pourrait d'agir de modifier la taille et la répartition de la stroke selon la profondeur ou du niveau de détail de la scène afin d'accorder plus d'importance à certaines zones par rapport à d'autres et donc de choisir ce que l'on veut donner à voir au spectateur. Cette technique, qui sera ici appliquée en deux dimensions, nécessite ce qu'on appelle un tableau ou canevas sur lequel seront disposées les *strokes*. Ainsi, bien que la technique SBR soit automatisée, un algorithme de ce type doit se soumettre à une intention précise posée par l'artiste et son utilisation requiert une sensibilité esthétique si on veut qu'elle soit efficace.

2.1.2 Problématiques et enjeux

Plusieurs questions et difficultés se posent lors de la réalisation d'un tel programme. Il y a tout d'abord la répartition: comment doivent être placées ces *strokes* sur l'image? Il est possible de les répartir de manière uniforme grâce à différentes méthodes comme par

exemple les diagrammes de *voronoi*. Cependant, il peut être plus intéressant de les placer en fonction du niveau de détail exigé pour une zone donnée grâce, par exemple, à l'emploi de calques. Ce placement dépendra alors d'une intention ou d'un but énoncé par l'utilisateur et de la composition de sa scène. Il faut également réfléchir à quelles informations on souhaite tirer de l'image ou de la scène de base pour mener à bien le processus, afin que les *strokes* conservent, modifient ou amplifient certains aspects de l'image donnée en entrée. Par exemple, comment récupérer les informations de profondeur ou d'orientation? Quel aspect mettre en valeur? Enfin, la gestion du mouvement et la cohérence dans le temps apporte aussi son lot de difficultés. Il s'agira d'expérimenter des méthodes pour permettre d'éviter ce qu'on appelle le *flickering*, ou changement brutal d'aspect d'un élément de la scène, ce qui génère un effet de saute. Il y a également le *shower door* qui donne l'impression qu'un objet se déplace derrière un rideau de couleurs. Cela est dû au fait que les *strokes* ne suivent pas ou mal la mouvement des objets. Celui-ci a donc l'air de simplement passer derrière. Toutes ces questions devront tenir compte du fait qu'il s'agit de rendu en temps réel et qu'il faudra optimiser le code pour éviter de surcharger la machine. En plus de ces problématiques, on devra veiller à ce que l'outil soit suffisamment intuitif et simple à manipuler. Effectivement, on peut parfois être tenté d'ajouter un nombre trop important de paramètres et d'options qui rendent l'interface incompréhensible. Le résultat doit également être à la hauteur de ce que souhaitait obtenir l'artiste et ne pas dépasser un certain niveau d'entropie. Ainsi, il s'agira de gérer l'utilisation de l'aléatoire pour avoir un rendu qui ait un aspect plus ou moins esquissé, désordonné sans pour autant que le résultat soit totalement imprédictible car l'artiste doit garder un contrôle sur ce que l'image va exprimer et sur ce à quoi elle ressemblera.

2.1.3 Intérêt pour la création artistique

J'ai choisi d'expérimenter cette technique spécifiquement car elle met l'accent sur un élément qui a longtemps revêtu une certaine importance dans l'histoire de l'art: la touche picturale. Celle-ci dispose en effet de qualités notables en terme d'expression artistique. Elle est généralement employée par les peintres pour traduire une intention, guider le regard, ajouter du dynamisme ou souligner et mettre de l'emphase sur certains éléments tels que des objets ou de la lumière. La touche picturale peut donc remplir différentes fonctions

expressives et est ainsi d'une grande ressource pour les artistes. Elle agit comme une trace de leur passage et permet d'inscrire l'œuvre dans une forme de matérialité. Aujourd'hui, elle a presque complètement disparu de nos médiums numériques. Il peut être intéressant de voir comment il serait possible de la réhabiliter et de l'adapter à ce nouveau type de support. Lors de la réalisation de cet outil, je me suis inspirée de la manière de peindre de certains artistes comme Claude Monet et André Derain pour leur emploi de la couleur et des coups de pinceau. Les peintures figuratives de Kandinsky ont également été une source d'inspiration. Cependant, je n'ai pas voulu me limiter à ces seuls exemples car je souhaitais que l'outil puisse au maximum s'adapter aux besoins expressifs de l'artiste ou du graphiste en générant une quantité de styles innovants et d'aspects visuels très différents les uns des autres. Le but n'était absolument pas d'imiter à la perfection un style graphique ou pictural existant mais de se baser sur une sélection de procédés employés par des artistes, dessinateurs et peintres, pour exprimer leur vision du monde et de les exploiter grâce à l'outil numérique afin d'inventer de nouvelles formes esthétiques.



Illustration 70 - Paysage d'hiver - Vassily Kandinsky (1909)

2.2 Première variante de l'outil

2.2.1 Gagner en performances et en maniabilité avec les compute shaders

Après avoir tenté de très brèves approches d'instanciation de formes en me basant sur une image par l'emploi de méthodes de traitement d'image en *post-processing* telles que

celles étudiées durant la première expérimentation, je me suis rapidement rendue compte que l'utilisation de pixels et vertex *shaders* classiques était difficile à adapter à la méthode SBR. En effet, les fragment *shaders* ne permettent pas de conserver les informations des calculs précédents. Il est certes possible de stocker des informations en sortie dans des textures, si par exemple on souhaite conserver des données telles que position ou l'orientation précédente d'une stroke, mais ce processus d'écriture et de lecture peut alourdir le temps de calcul est difficile à manier. Il serait beaucoup plus pratique, par exemple, d'avoir une structure dans laquelle on définit plusieurs attributs qui constituerait un objet de type stroke et de pouvoir ensuite instancier et mettre à jour à chaque frame les informations de chacune de ces *strokes*. On pourrait faire cela simplement dans Unity avec un script mais le fait de devoir itérer sur un grand nombre d'opérations à la suite risquerait vite de réduire les performances de l'application. Sachant qu'on a besoin de placer un assez grand nombre de *strokes* pour que celles-ci puissent recouvrir toute l'image, ce procédé pourrait donc nous limiter. Une meilleure démarche serait alors d'effectuer les opérations de chacune de ces *strokes* simultanément via le GPU. C'est exactement le genre de choses que permettent de faire les *compute shaders*.

Nous avons vu dans la première partie qu'un *shader* est un programme s'exécutant sur le GPU permettant d'opérer un grand nombre de threads simultanément, les threads ne communiquant pas les uns avec les autres et étant indépendants. Les vertex et fragment *shaders* sont inclus obligatoirement dans le pipeline graphique et ont des entrées et des sorties définies que l'on ne peut personnaliser. Ainsi, ils ont une utilisation spécifique et ne permettent pas au programmeur se s'en servir pour réaliser d'autres tâches et utiliser d'autres types de données. Au contraire, les *compute shaders*, appelés aussi *compute kernels*, sont un type de *shader* qui n'est pas spécifiquement destiné aux calculs de rendu. Ils ne font d'ailleurs pas partie du pipeline graphique et peuvent être utilisés indépendamment de celui-ci. On peut les employer dans des domaines très divers: par exemple générer des textures, faire de la simulation physique, gérer des systèmes de particules ou encore créer des objets de manière procédurale. Leur fonction est générale: c'est au programmeur de décider des tâches qu'il veut faire exécuter ainsi que des entrées et sorties. Afin de profiter de l'accélération offerte par le GPU, il d'abord que le programme à implémenter soit adapté à la programmation en parallèle. Tous les problèmes ne peuvent pas être résolus de manière efficace avec cette méthode. Ici, on souhaite effectuer la même opération sur plusieurs

centaines ou milliers d'éléments afin de calculer leurs attributs (couleur, position, orientation...). Ainsi, pouvoir effectuer ces opérations en parallèle va nous offrir un grain de performances important.

Dans Unity, les *compute shaders* s'écrivent également avec le langage HLSL. A l'intérieur du *compute shader*, faut d'abord définir ce qu'on appelle un *kernel*, c'est à dire une fonction qui va effectuer des opérations au sein du *shader* et qui sera appelée par le CPU. Elle a comme paramètre en entrée l'index du thread courant. On doit ensuite déclarer les différentes variables qui peuvent être partagées entre le CPU et le GPU. Avant d'écrire le *kernel*, il faut aussi renseigner le nombre de threads dans un tableau à trois dimensions que nous allons utiliser pour chaque groupe de thread. Un groupe se compose de 1024 threads au maximum et est instancié par le CPU lorsque celui-ci exécute le *kernel*. Il faudra par exemple adapter le nombre de groupes de threads en fonction du nombre de *strokes* à instancier, en les distribuant par exemple par paquets de 1024 threads. On pourra ensuite appeler le *kernel* grâce à la fonction Dispatch qui prend en paramètres le *kernel* choisi et le nombre de groupes. Les résultats calculés par le *compute shaders* sont récupérés dans un tampon ou buffer. On peut également récupérer des textures mais ce n'est pas ce qui nous intéresse dans ce projet.

2.2.2 Structure principale de la stroke

On va en premier lieu créer une structure qui contiendra les informations nécessaires à chaque stroke pour pouvoir l'instancier. Cette structure est proche de ce qu'on peut retrouver dans un système de particules classique. Comme attributs principaux, on peut compter la position de la stroke sur l'écran, sa couleur, sa taille, son orientation et sa forme (c'est à dire l'aspect de la brosse à appliquer). D'autres valeurs seront ajoutées par la suite.

Il faut déclarer cette structure dans le *compute shader* ainsi que dans le script qui va appeler le *compute shader* et veiller à ce que les noms de variables correspondent. Elle sera ensuite passée au GPU grâce au *compute buffer* dans lequel on stocke les données relatives aux *strokes*. Pour éviter les fuites de mémoire, il faut bien penser à libérer les emplacements utilisés par le buffer grâce à la fonction Release lorsqu'on a fini de l'utiliser.

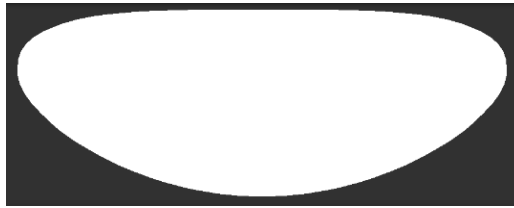


Illustration 71 - SBR - Exemple de forme possible de la stroke après déformation par le vertex shader et avant traitement du pixel shader

Concernant l'apparence finale du *mesh* qui sera instancié, j'ai créé un simple quad auquel j'ai ajouté quelques subdivisions sur un seul axe dans le but d'avoir la possibilité par la suite de gérer la courbure de la stroke. Afin d'afficher les *strokes* à l'écran, on utilisera la fonction `Graphics.DrawMeshIndirects`. Elle consiste à dessiner un *mesh* en utilisant l'instanciation par GPU, c'est à dire que Unity ne créera pas de *game objects* dans la scène. La fonction indirecte permet d'instancier à l'aide d'un buffer, appelé `bufferWithArgs`, qui contient le nombre d'instances du mesh à dessiner. Dans cette fonction, on passe également un matériau qui sera appliqué à l'instance et un volume au sein duquel l'instance pourra être dessinée. Ce matériau utilise un *shader* spécifique qui va récupérer les informations de la structure pour ensuite pouvoir manipuler l'instance via la partie *vertex shader*, en effectuant par exemple des transformations matricielles afin de calculer son orientation finale. La partie *fragment shader* servira à calculer et à afficher la couleur définitive de la stroke. Pour réaliser cette étape, je me suis aidée du guide *Unity Graphics Programming volume 1*⁶⁵ de Masatatsu Nakamura qui aborde les *compute shaders* dans le cadre de la modélisation procédurale.

2.2.3 Transformations et placement des touches de peinture

Un *game object* que j'appellerai `emitter` est placé dans la scène 3D. Il contient le script permettant de faire appel au *compute shader* pour générer les *strokes*. C'est sur ce *game object* que l'utilisateur peut moduler les différents paramètres. Nous allons à présent expliquer la manière avec laquelle les différents attributs des *strokes* sont calculés.

Répartition des *strokes* et travail par calques

⁶⁵ 技術同人誌 「Unity Graphics Programming vol.1」

表紙グラフィックス作成のためのUnityプロジェクト:

IndieVisualLab/UnityGraphicsProgrammingFrontCover. C#. 2017. Reprint, Indie Visual Lab, 2019.

<https://github.com/IndieVisualLab/UnityGraphicsProgrammingFrontCover>.

Les *strokes* sont instanciées aléatoirement dans l'espace 2D de l'écran et se placent par dessus l'image rendue. Elles sont disposées sur trois calques différents qui correspondent à trois niveaux de profondeur: le front layer, le mid layer et le back layer. L'utilisateur a la possibilité de définir le nombre de *strokes* qu'il veut voir apparaître sur les différents calques ainsi que de modifier des paramètres spécifiques aux *strokes* appartenant à chacun des calques. Il peut aussi changer le niveau de profondeur attribué à chaque calque s'il le souhaite. Ainsi, les *strokes* pourront être paramétrées selon deux niveaux: d'abord au niveau global, qui s'applique à toutes les *strokes*, et ensuite au niveau du calque auquel appartient la *stroke*.

Gestion de la couleur

Pour déterminer la couleur d'une *stroke*, on échantillonne l'image rendu en sortie à l'endroit où se trouve la *stroke*, puis on lui assigne cette couleur comme base. L'utilisateur peut ensuite jouer sur différents paramètres afin de transformer cette couleur de base. En effet, je souhaitais lui permettre de travailler la couleur de l'image directement via les *brushes* plutôt qu'avec des effets de *post-processing*. Il a ainsi la possibilité d'effectuer un *color jitter*: la *stroke* prend deux couleurs en entrée données par l'utilisateur et grâce à une fonction de *pseudo random*, le programme calcule une couleur interpolée aléatoirement entre ces deux couleurs. Il peut ensuite choisir de mixer la couleur obtenue avec celle du rendu de base en modifiant l'intensité de la teinte et son opacité. Cela lui permet d'assembler plusieurs couleurs et de générer des effets de contraste. Il a aussi la possibilité de gérer des niveaux d'aléatoire pour ces paramètres afin de ne pas se retrouver avec un rendu trop uniforme. Certains paramètres sont également modulable en fonction du calque attribué à la *stroke*. J'avais également tenté d'ajouter des paramètres s'appliquant à différentes zones en fonction de l'intensité de la lumière, pour par exemple avoir une teinte qui ne s'applique que dans les aires les plus lumineuses et une teinte pour les ombres afin d'aider l'artiste à jouer sur les contrastes grâce à la couleur. Malheureusement, les calculs étaient trop lourds et réduisaient drastiquement le *frame rate* et j'ai donc préféré désactiver cette option.

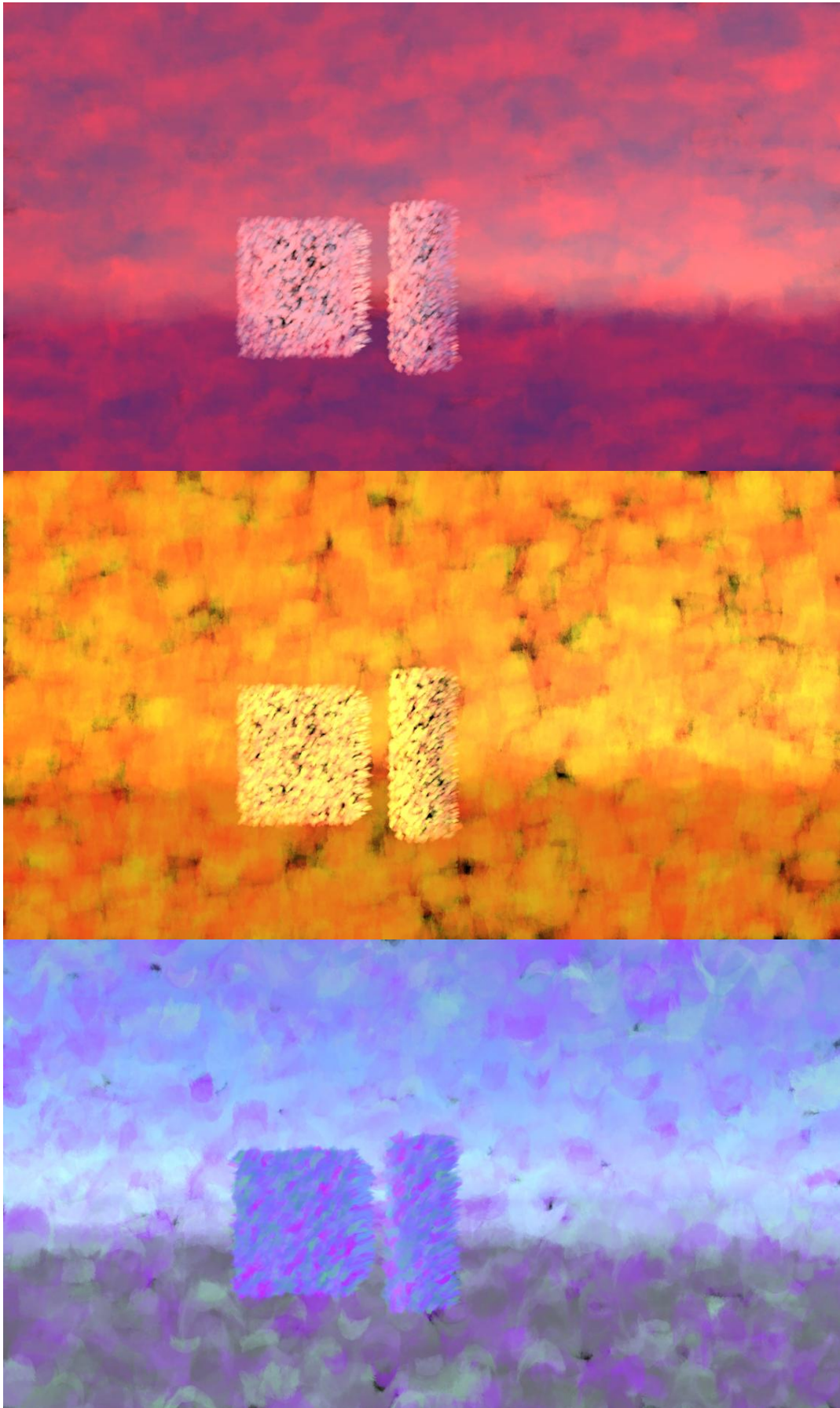


Illustration 72 - SBR Ambiances de couleur

Choix de la forme

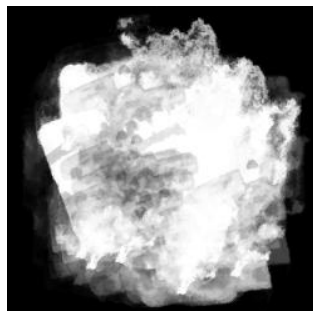


Illustration 73 - SBR - Exemple de masque pouvant être utilisé

La forme de la *stroke* est déterminée par une texture de masque donnée en entrée. L'utilisateur a la possibilité d'entrer autant de masques qu'il le souhaite en fonction des calques. Le script va ensuite assigner aléatoirement à chaque *stroke* l'identifiant d'une texture qui pourra finalement être récupérée dans le *shader* qui permettra d'afficher la *stroke*. Il est donc possible d'utiliser uniquement certains types de brosses à l'arrière plan et de changer le style pour peindre les objets situés au milieu ou à l'avant plan.

Calcul de l'orientation

L'orientation des touches joue beaucoup sur l'aspect final d'un tableau. Comme nous l'avons déjà souligné dans la partie théorique, le sens dans lequel est appliquée une touche de peinture lors de l'exécution va donner une intention dans l'image. Généralement, les artistes ont tendance à s'en servir pour mettre en valeur la forme des objets en suivant ses courbes, comme on peut le voir dans *Nature morte aux pommes* de Cézanne. Certains artistes au contraire, peignent de manière aléatoire, ce qui induit une forme d'énergie et de vigueur désordonnée dans l'œuvre. Enfin, il y a des peintres qui l'emploient pour exprimer une intention spécifique, pour évoquer le mouvement ou pour amplifier la lumière de la scène. On retrouve souvent ce procédé dans les tableaux de Van Gogh. L'outil que je développe permettra de contrôler un niveau d'aléatoire grâce à un paramètre donné par l'utilisateur. Par défaut, les *strokes* devront se placer de manière à épouser la forme des éléments de la scène. Cela a posé certaines difficultés et j'ai effectué plusieurs recherches infructueuses à ce sujet.

J'ai d'abord tenté d'utiliser un filtre de Sobel. Ce procédé, vu rapidement au cours de la première expérimentation, permet de détecter les contours dans une image par

approximation d'une dérivée. Elle sert à calculer la différence entre les régions situées autour du pixel courant. Par exemple, lorsqu'il y a un changement brutal d'intensité, cela indique la présence d'un contour. Il faut détecter séparément les contours horizontaux et verticaux grâce à deux matrices, qu'on appelle masques de convolution.

-1	0	+1
-2	0	+2
-1	0	+1

x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

Illustration 74 - Matrices de convolution pour le filtre de Sobel

Ces matrices permettent de récupérer la quantité de différence d'intensité dans une région de pixels. Pour cela, elles doivent être placées sur chaque pixel de l'image. On va ainsi convoluer la région qui entoure le pixel courant grâce aux deux matrices. L'opération de convolution consiste, pour chaque élément de la matrice, à le multiplier par l'élément correspondant dans la matrice de convolution et à additionner les résultats. On obtient un gradient qui est un vecteur représentant le sens dans lequel la variation est la plus forte et on peut alors calculer la norme de ce vecteur pour obtenir l'intensité de la variation. J'ai échantillonné le résultat à l'emplacement de chaque *stroke* pour les orienter selon cette variation de pixel mais le résultat n'était pas du tout concluant car les variations étaient trop faibles là où il n'y avait pas de contour et ne permettaient pas de retrouver la forme globale de l'objet. J'ai également appliqué des passes pour flouter l'image et essayer ainsi d'approximer un champ vectoriel mais le résultat n'était pas suffisant pour être appliqué à un niveau global.



Illustration 75 - Exemple de calcul détection de contours effectué au cours des recherches

Je me suis alors tournée vers la méthode des *signed distance field* pour orienter les *strokes* perpendiculairement à la direction du point le plus proche situé sur un contour. C'est une technique souvent utilisée en raymarching car elle permet de trouver la distance la plus proche entre une surface ou une courbe et un point. Plusieurs algorithmes permettent d'effectuer ces calculs, la méthode la plus simple, en brute force, consiste à calculer la distance euclidienne entre le pixel courant et tous les pixels de la texture pour trouver le pixel "rempli" le plus proche, le pixel "rempli" étant un pixel présent sur un contour. D'autres méthodes plus rapides existent, comme par exemple la méthode par propagation. Une fois que l'on connaît la distance de chaque pixel par rapport au pixel le plus proche sur un contour, il suffit de calculer la dérivée partielle dans les axes x et y du champ pour obtenir un vecteur. Pour cela, on recherche parmi ses deux voisins dans chaque axe, lequel a la valeur la plus faible, c'est à dire lequel est le plus proche d'un contour et on récupère sa valeur qu'on soustrait à la valeur du pixel courant. Il faut également tenir compte du signe en ajoutant -1 ou +1 dans chaque axe en fonction de la direction puis diviser le résultat précédent par la somme de celui-ci. J'ai tenté d'implémenter cet algorithme dans Unity, mais le temps de calcul des gradients était trop long, ce qui ne rendait pas possible l'utilisation de cette méthode en temps réel. Il faudrait tenter d'adapter l'algorithme pour qu'il puisse s'exécuter plus rapidement en GPU mais faute de temps, j'ai préféré me tourner vers une autre méthode plus adaptée à l'utilisation de scènes 3D en temps réel.

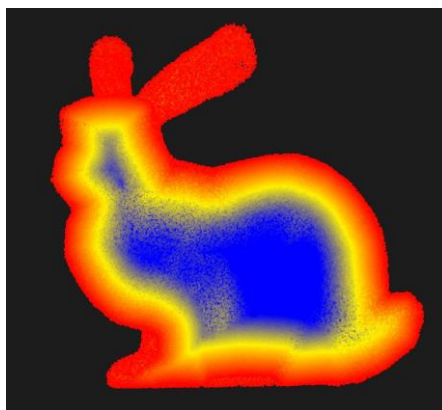


Illustration 76 - Exemple de dégradés pouvant être obtenus par l'utilisation de SDF

Ainsi, j'ai récupéré les normales des géométries de la scène dans l'espace de la caméra avec la variable `COMPUTE_VIEW_NORMAL`. J'ai projeté les vecteurs sur un plan 2D car je veux que les *strokes* soient toujours alignées face à la caméra. Les rotations de la stroke s'effectuent donc uniquement sur l'axe z dans l'espace de la caméra. De plus, j'ai créé

une variable permettant de gérer l'aléatoire et j'ai combiné ces valeurs pour permettre aux artistes de moduler le niveau de fidélité par rapport à la scène de base. L'inconvénient de cette méthode est qu'elle se base uniquement sur la géométrie de la scène et non sur l'image rendue. Elle ne permet donc pas de prendre en compte les textures. Par exemple, si on a une modélisation d'arbre et que les feuilles sont représentées par des plans texturés, les *strokes* ne suivront pas vraiment la courbure des feuilles mais celles du plan. Généralement, cette différence passera inaperçue, mais elle peut poser problème sur des plans rapprochés. Néanmoins, le résultat reste satisfaisant et applicable dans la majorité des cas.

Nous verrons comment apporter d'autres types d'intention telles que les effets de mouvement que l'on peut retrouver dans les tableaux de Vincent Van Gogh lors de la description de la deuxième variante de l'outil.

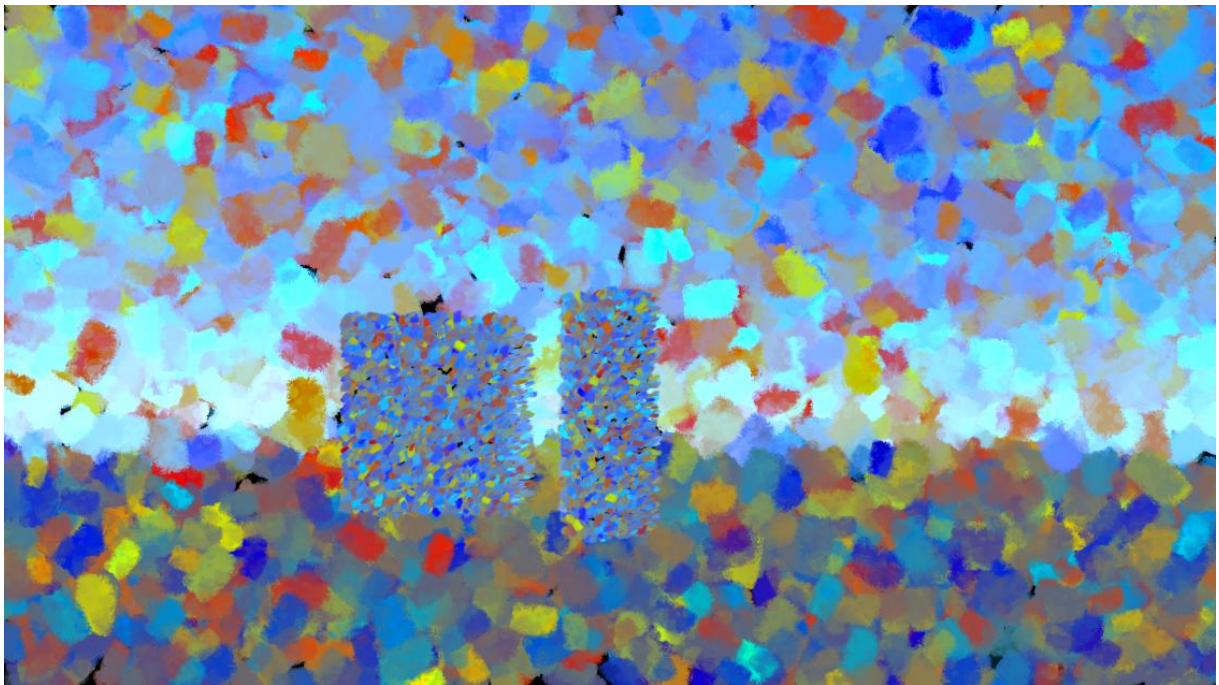


Illustration 77 - SBR Exemple de rendu sur primitives par instanciations semi-aléatoires



Illustration 78 - SBR Exemple de rendu sur primitives par instanciations déterminées par les normales

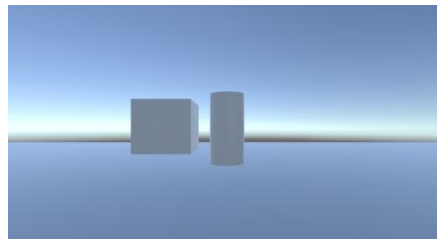


Illustration 79 - SBR Image de base

Détermination de la taille

La taille des *strokes* va dépendre de plusieurs paramètres. D'abord, la taille globale est calculée à partir de la *zdepth*: plus le pixel échantillonné sera éloigné de la caméra, plus la stroke placée en ce point sera grande. Je me suis inspirée du fait que les artistes ont souvent tendance à peindre les objets plus proches avec plus de détails. Cette taille pourra cependant être modifiée par un facteur lié spécifiquement au calque auquel appartient la stroke. D'autres paramètres d'aléatoire s'ajoutent également au résultat final. J'ai en plus donné la possibilité, via la détection de contours effectuée précédemment, de réduire la taille des *strokes* qui se situent au niveau des contours des objets afin de mettre plus en valeur leur forme. Il y a également un paramètre de longueur et d'épaisseur de la brosse qui peut être changé par l'utilisateur, ainsi que de l'aléatoire.

Ajout d'une courbure

En peinture, les coups de pinceau ne sont pas forcément appliqués de manière raide mais peuvent aussi être ondulés ou cambrés. Jusqu'ici, l'apparence droite des touches de peinture donnait parfois un aspect rigide et artificiel puisque, dans la réalité, les fluides ne sont jamais posés d'une manière parfaitement rectiligne. J'ai donc ajouté un attribut à la stroke que j'ai nommé *bend amplitude*. Cet attribut est récupéré dans le *vertex shader* qui va déformer les points en fonction de la valeur de l'amplitude dans l'axe des *y*. J'effectue un calcul en fonction du cosinus des indices des points afin d'avoir une transformation progressive ressemblant donc à une forme de vague dans l'espace local de la stroke. Pour l'instant, ces déformations s'effectuent de manière aléatoire selon une intensité définie par l'utilisateur. Une amélioration possible de ce procédé serait d'accorder plus de contrôle aux artistes en fonction de données spécifiques. Il serait en effet plus efficace de laisser l'utilisateur choisir un niveau et un type de déformation en fonction de certaines zones de la scène, par exemple.

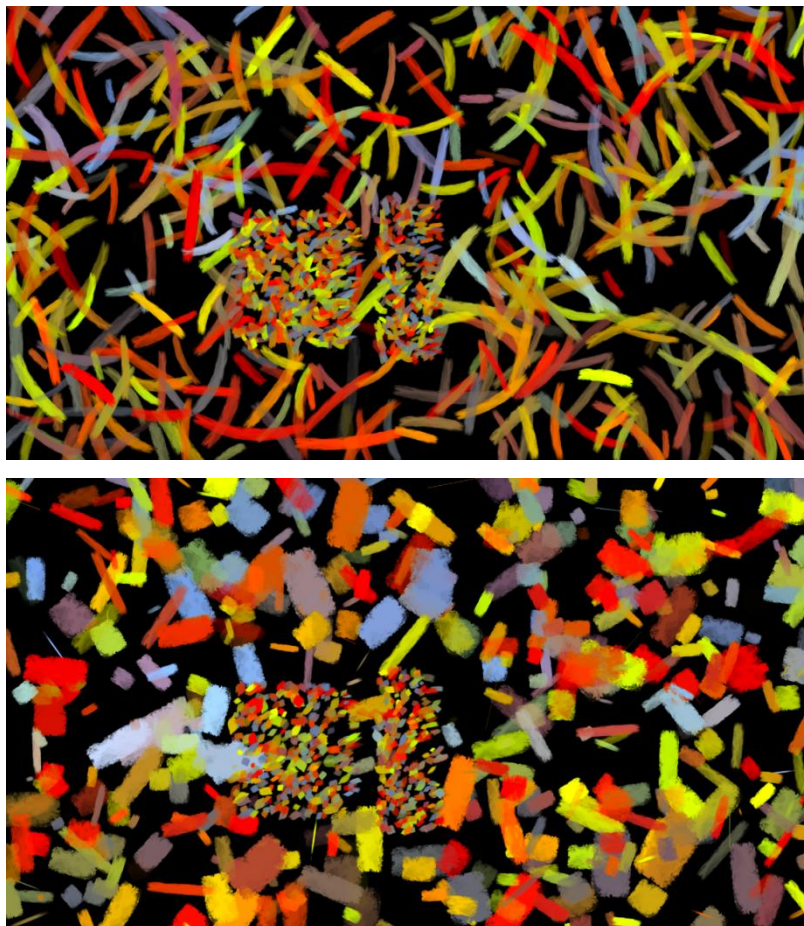


Illustration 80 - SBR Instanciations en variant les paramètres de taille et de courbure des stokes

La problématique du mouvement

Pour gérer le mouvement des *strokes* et éviter l'effet *shower door*, il faut trouver un moyen de traquer le mouvement des objets de la scène afin de faire en sorte que les *strokes* suivent plus ou moins la position d'un point de la surface de la géométrie. Etant donné que nos instances sont placées dans l'espace 2D de l'écran, il peut être ardu de suivre les variations de position des géométries qui se situent dans l'espace monde. Dans Unity, il est possible de récupérer à partir d'une caméra différentes textures dont une texture d'*optical flow* ou flux optique. Il s'agit d'une texture qui contient les informations de mouvement apparent de différents points de la scène par rapport à la caméra. Le flux optique est souvent utilisé en traitement vidéo pour estimer le mouvement dans une image et par exemple effectuer du tracking ou recréer une caméra virtuelle. Ici, grâce aux informations de direction conservées dans les pixels, il est possible de déplacer les *strokes* afin de les faire suivre les points en mouvement. J'ai ainsi ajouté un attribut de vitesse aux *strokes* qui récupère les valeurs de directions à partir de la texture d'*optical flow*. Ensuite, pour chaque *stroke*, s'il y a un mouvement apparent à leur position courante, c'est à dire, si le pixel échantillonné n'a pas une valeur nulle, alors on ajoute à la position actuelle la valeur de la vitesse multipliée par le deltatime et un facteur de vitesse. Un problème peut alors se présenter: les *strokes*, une fois déplacés, s'accumulent dans certaines zones de l'écran. Pour pallier à ce problème, j'ai assigné une durée de vie à chaque *stroke* que je diminue à chaque fois que celle-ci est déplacée. Si sa durée de vie est nulle, alors elle disparaît progressivement et se repositionne à un autre endroit de la scène. Cela permet de limiter l'apparition de trous lors du rendu, si par exemple certaines *strokes* se retrouvent balayées de l'écran.

Une autre difficulté peut parfois apparaître lors de la transformation des *strokes*, par exemple lorsque celle-ci change de taille ou d'orientation. En effet, il peut arriver que le changement soit trop brusque et que cela crée des effets de *flicking*. Il est possible de stabiliser ces mouvements en interpolant progressivement l'orientation ou la taille calculée à la frame précédente avec celle calculée à la frame courante pour éviter d'obtenir des transformations trop subites.

Récupération des informations de la scène

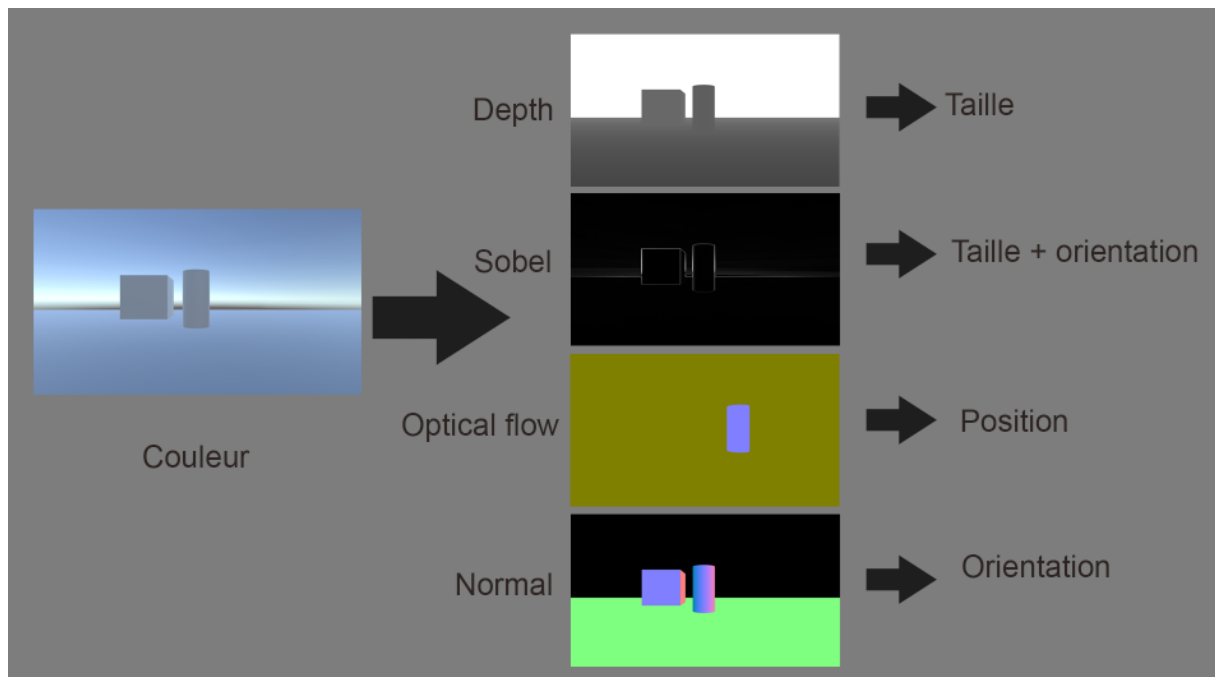


Illustration 81 - SBR Extraction des informations nécessaires à l'instanciation de strokes

Un des principes de la technique SBR est de générer des *strokes* en fonction des informations extraites d'une ou plusieurs images en entrée. Ici, étant donné que nous travaillons dans une scène 3D, il est plus simple de récupérer ces informations directement sans avoir à effectuer d'opérations de traitement d'image comme on peut le voir avec certaines technique qui se basent sur des photographies ou des vidéos. Pour pouvoir calculer tous les attributs décrivant le comportement des *strokes*, il est indispensable de bien obtenir les différentes informations nécessaires dans la scène 3D. J'ai pour cela créé un rig de caméras dont chacune d'entre elles rend une texture spécifique qui sera récupérée dans le script qui gère les instanciations. J'ai ainsi écrit un script que j'ai nommé `GeneratePasses` qui se charge d'attribuer les bons *shaders* aux différentes caméras. Dans Unity, on peut facilement obtenir certaines informations comme la distance de l'objet par rapport à la caméra, ou l'*optical flow* grâce à des fonctions spécifiques. Les *shaders* auront majoritairement pour but de récupérer et de convertir les données obtenues via la caméra en format `rgb` et de les normaliser. On utilise donc cinq textures différentes: une texture de couleur, une texture de profondeur, une texture de normal, une texture de flux optique et une texture de détection de contours. Elles pourront ainsi, selon les besoins, être échantillonnées pour calculer les attributs des *strokes*.

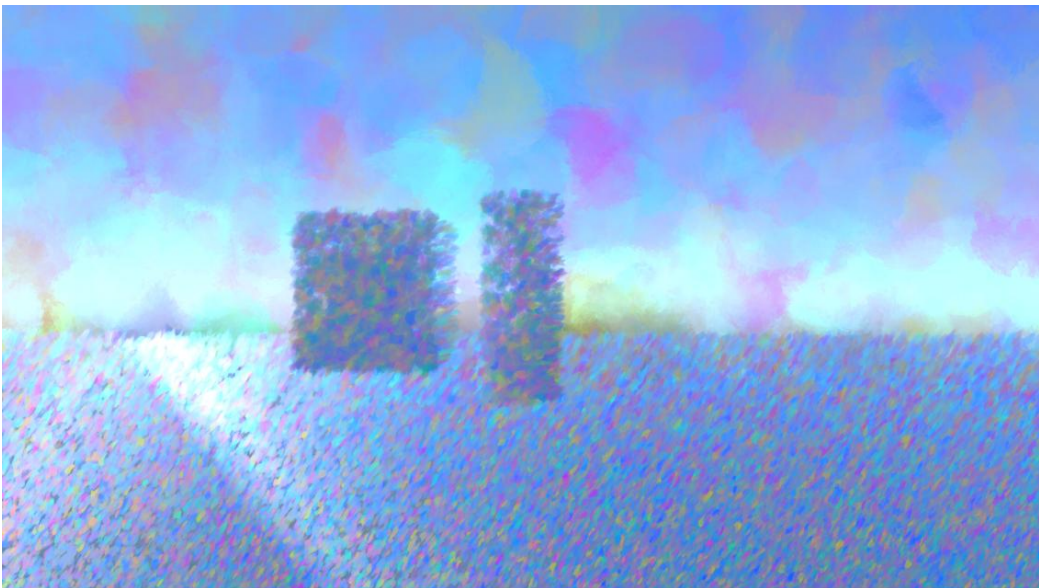
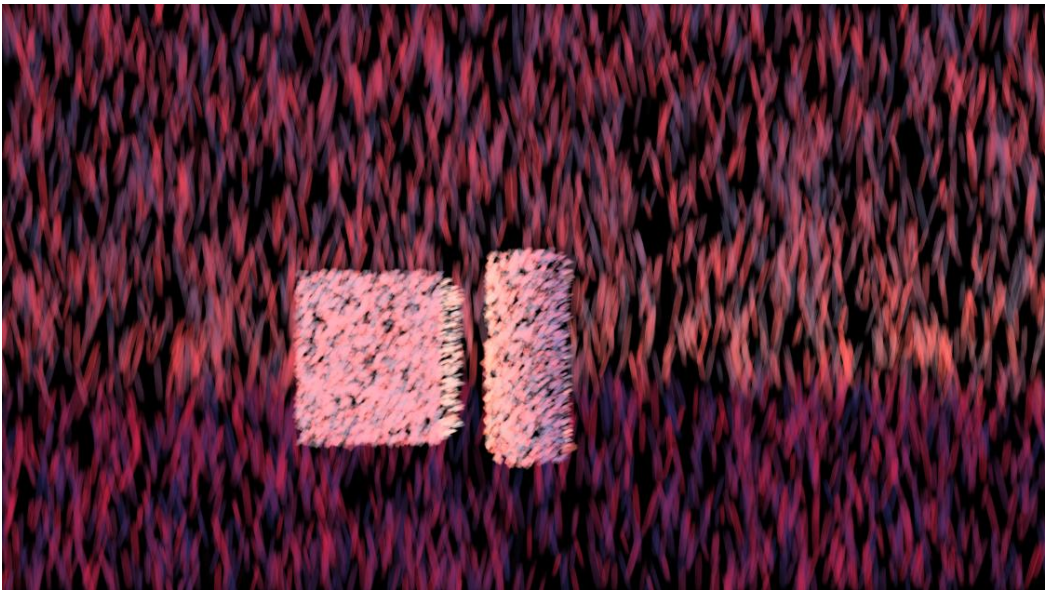
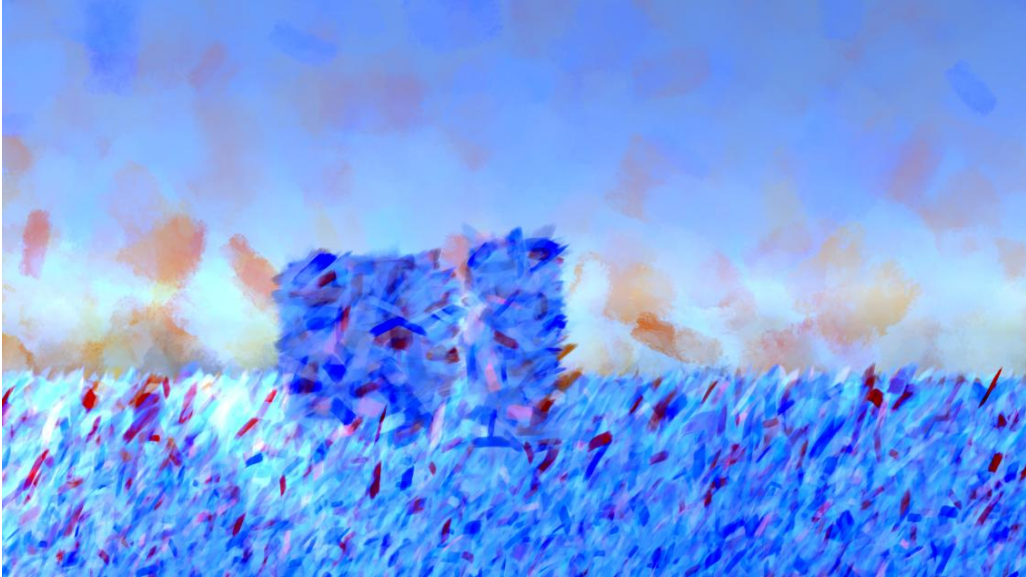


Illustration 82 - SBR Autres exemples d'effets que l'on peut obtenir

2.3 Deuxième variante: Wind Painter

2.3.1 Retrouver la trace de l'artiste par l'évocation du mouvement

Parallèlement au développement de l'outil, j'ai expérimenté d'autres types de procédures afin notamment d'accorder plus de place au mouvement. J'ai donc dérivé mon outil principal pour en créer une variante que j'ai nommée Wind Painter. J'ai eu envie de m'inspirer de certains aspects de la nature un peu à la manière de Van Gogh ou Emily Carr qui suggèrent le mouvement dans leurs paysages grâce à des traînées ondoyantes de peinture.



Illustration 83 - Above the grave pit - Emily Carr (1937)

Je me suis par exemple basée sur certains effets visuels que l'on peut voir sur des représentations de cartes météorologiques qui figurent le trajet du vent et des courants qu'on appelle *wind maps*.

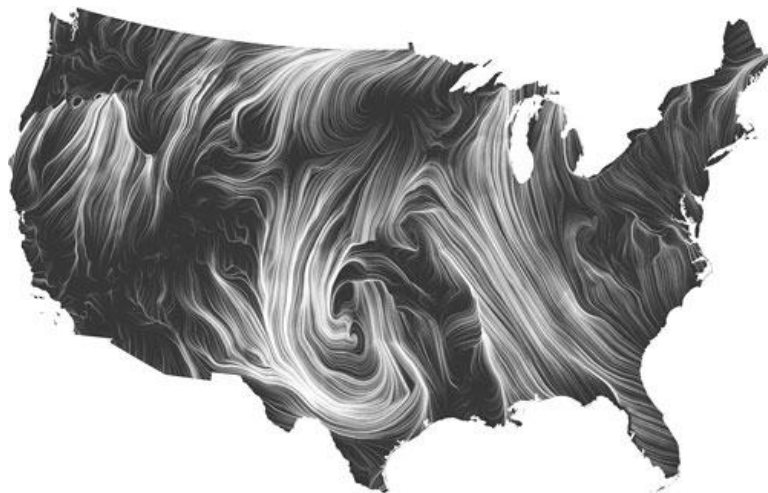


Illustration 84 - Carte des vents

L'emploi de ce type de courbes, associé à un travail de couleur, sont capables de générer une forme d'énergie au sein du tableau. On peut par exemple noter ce genre de procédé dans les peintures de Julia Watkins. J'ai également pris pour référence des œuvres génératives telles que celles de Steve Edwards pour la manière avec laquelle s'étend et se propage la matière picturale virtuelle ainsi que les ambiances de couleur. D'autres artistes contemporaines ont inspiré la création de cet outil, comme Erin Hanson, une peintre post impressionniste qui réalise des paysages colorés dont le déplacement est évoqué par les touches de pinceau. Je me suis aussi basée sur certaines œuvres de Ivana Besevic pour l'aspect diffus, déstructuré et entremêlé que produisent ses touches de couleur. J'apprécie particulièrement l'aspect organique et fugitif qui se dégage de ses images.

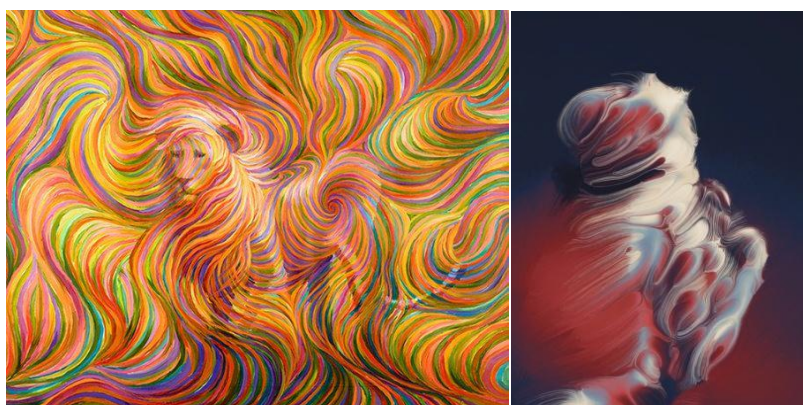


Illustration 85 - Lion Spirit Guide - Julia Watkins
Illustration 86 - Generative art - Steve Edwards



Illustration 87 - Dreamscape - Erin Hanson (2016)
Illustration 88 - Digital painting - Ivana Besevic

Enfin, j'ai eu en plus comme sources d'inspiration certaines installations interactives comme celles du Flow Project de l'artiste Qiu Yi Wu⁶⁶, avec par exemple *Flow Brushes*, une installation qui capture le mouvement des spectateurs pour générer des flots de peinture aquarelle qui se diffusent autour de lui. Ainsi, un des enjeux de cet outil sera d'employer les raccourcis picturaux évoquant le mouvement et la déstructuration pour les amplifier grâce à l'animation et le temps réel afin d'évoquer et d'exprimer la trace de l'artiste.



Illustration 89 - Flow Brushes - Qiu Yi Wu (2015)

2.3.2 Comportement des strokes au cours du temps

Wind Painter est un outil plus expérimental qui emploie néanmoins des principes similaires à ceux du projet présenté précédemment. De la même manière, j'utilise un *compute shader* qui permet de calculer et d'instancier des *strokes* sur un écran en deux dimensions. La différence cette fois-ci est que les *strokes* se comportent comme des particules et se déplacent d'une certaine manière à travers l'image, générant des traînées de peinture dont l'aspect est paramétrable. Ainsi, ce sont les *strokes* qui sont chargées, par leur mouvement, de peindre l'image. Pour réaliser cet outil, je me suis inspirée de fonctionnements décrits dans certains articles tels que *How I built a wind map with WebGL*⁶⁷

⁶⁶ « The Flow Project: Connecting Colors and Movement | The Metropolitan Museum of Art ». Consulté le 12 février 2019. <https://www.metmuseum.org/blogs/digital-underground/2015/the-flow-project>.

⁶⁷ Agafonkin, Vladimir. « How I built a wind map with WebGL ». Points of interest, 27 juillet 2017. <https://blog.mapbox.com/how-i-built-a-wind-map-with-webgl-b63022b5537f>.

de Vladimir Agafonkin, *Generative Impressionism*⁶⁸ de Matt DesLauriers et *Painterly Particular* de Peter Norrby⁶⁹.

Afin de générer le mouvement des *strokes*, plusieurs méthodes sont possibles. J'ai donné la possibilité à l'utilisateur de donner des *flows maps* en entrée ou des textures en niveau de gris. Il a aussi la possibilité de ne pas utiliser de textures et de faire en sorte que les particules suivent la forme des géométries en fonction des normales de celles-ci, ou encore de combiner l'utilisation des normales et de textures personnalisées. Ainsi, à chaque frame, le *compute shader* calcule la nouvelle position à partir des informations échantillonnée sur la ou les textures données en entrée en fonction d'un facteur de vitesse. Ces paramètres vont servir à diriger la stroke de manière à lui donner une intention spécifique. Le traitement de l'information sera différent en fonction du type de texture. Par exemple, s'il s'agit d'une *flow map*, il faudra calculer la vitesse en x à partir de la channel rouge de la texture, et la vitesse en y à partir de la channel verte. Avec des textures en niveau de gris, il s'agira de calculer un angle en multipliant la valeur du pixel par $2 \cdot \pi$ et récupérer la vitesse en x et en y en utilisant le cosinus et le sinus de cet angle.

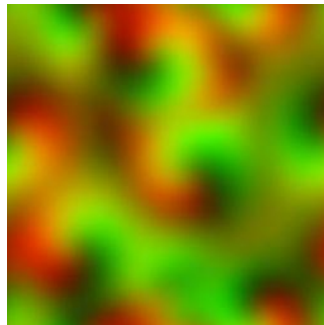


Illustration 90 - Exemple de flow map

Il est possible d'ajouter de l'aléatoire dans l'échelle des particules, de contrôler leur taille de leur définir une forme grâce à un masque. On peut également modifier leur vitesse et leur durée de vie. Tous ces paramètres peuvent être animés et ainsi créer une peinture dynamique dont l'aspect évolue au cours du temps. Pour garder la trace de leur passage, et ainsi leur permettre de remplir l'écran, j'utilise une option dans la caméra d'Unity qui

⁶⁸ DesLauriers, Matt. « Generative Impressionism ». *Matt DesLauriers* (blog), 9 avril 2014. <https://medium.com/@mattdesl/generative-impressionism-afa98ccb97da>.

⁶⁹ « Trapcode - News - Painterly Particular ». Consulté le 6 mai 2019. <http://trapcode.squarespace.com/journal/2014/1/30/painterly-particular.html>.

s'appelle cameraDon'tClear, qui permet de ne pas effacer l'image présente à l'écran et donc de la repeindre continuellement.

Deux modes de remplissage sont possibles: si le mode uniColor est activé, la stroke gardera la même couleur pendant toute sa durée de vie et son mouvement sera apparent sur l'image même si aucun objet de la scène ne bouge. Si ce mode est désactivé, alors la stroke prendra à chaque frame la couleur échantillonnée à sa position courante. L'existence des *strokes* sera visible uniquement lorsque la caméra ou un objet appartenant à la scène 3D se mettra à bouger, laissant une trace derrière lui le temps que les autres *strokes* la fassent disparaître en emplissant l'espace. Tout dépend de ce que l'artiste souhaite évoquer en utilisant l'outil.

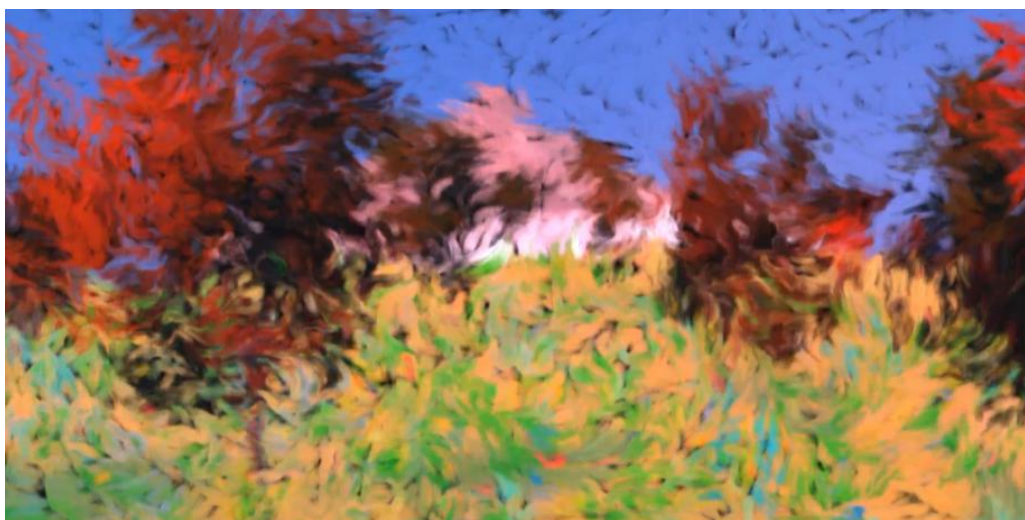


Illustration 91 - Wind Painter - Rendus sur paysages dans Unity

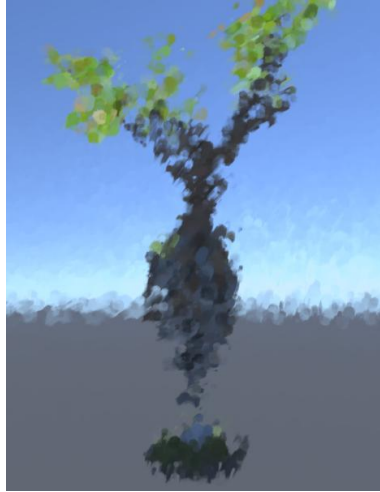


Illustration 92 - Wind Painter - Rendu sur une modélisation d'arbre

2.3.2 Une peinture générative guidée par le son ou la musique

Pour finir avec ces expérimentations, j'ai tenté d'employer le son, et éventuellement la musique comme moyen d'expression visuelle. En effet, la musique a toujours eu un lien plus ou moins direct avec la peinture, et certains artistes synesthètes tels que Kandinsky s'inspirent de la musique pour réaliser leur tableaux qu'ils qualifiaient de symphonie visuelle. J'avais déjà tenté ce type de projet l'année dernière en utilisant les notes de musique en entrée afin de gérer l'ambiance d'un décor ou d'influencer des animations plus ou moins abstraites. J'ai ainsi pu récupérer certains scripts utilisés pour échantillonner des sons capturés en entrée via un microphone ou une piste sonore afin de m'en servir pour moduler les paramètres des *strokes* et modifier leur trajet et leur apparence au cours du temps. Ainsi, certaines notes sont liées à des paramètres spécifiques et influencent le comportement de la peinture virtuelle. Cela permet d'offrir de nouvelles possibilités expressives à des artistes qui souhaiteraient emprunter des moyens originaux pour peindre et créer des animations dans des styles visuels particuliers.

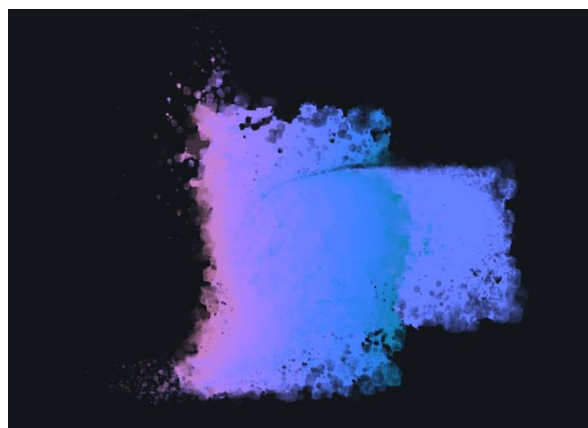


Illustration 93 - Wind Painter - Effets plus abstraits

2.4 Retour d'expérience et efficacité

L'outil que j'ai développé propose un nombre plutôt important d'options et permet de générer une variété de tableaux qui s'animent en temps réel en fonction des éléments disponibles dans la scène 3D. L'artiste occupe la place la plus importante puisque c'est lui qui est chargé d'offrir une description du style qu'il souhaite appliquer sur l'image via les paramètres mis à sa disposition dans une interface utilisateur. On se retrouve ainsi avec ce lien fort entre langage et expression artistique, où l'information laissée par l'artiste permet de définir l'aspect stylistique de l'image. L'emploi du mouvement, en particulier dans la seconde partie, apporte de nouvelles possibilités qui n'ont pas encore été beaucoup exploitées en production de films d'animation à ce jour. Enfin, la possibilité de personnaliser les brosses et de gérer leurs paramètres à différents niveaux d'interaction apporte une certaine liberté aux artistes qui peuvent ainsi générer des effets de matière en y apportant une touche vibrante et organique.

Il pourrait être intéressant de trouver des moyens de mieux optimiser le code afin d'ajouter plus de contrôle au niveau des zones d'ombres et de lumière, et pourquoi pas donner la possibilité d'orienter les *strokes* en fonction des informations de luminosité. Pour l'instant, j'ai préféré limiter l'outil à l'utilisation de trois calques pour ne pas complexifier l'interface utilisateur, mais il serait théoriquement possible d'en ajouter de nouveaux et même de laisser le choix du nombre de calques à l'utilisateur. Il faudrait aussi revoir le placement des *strokes* lorsque ceux-ci ont été bougés par l'*optical flow* pour qu'ils se positionnent aléatoirement en privilégiant les zones dans lesquels il la densité de *strokes* est la plus faible afin d'éviter de se retrouver avec des vides dans l'image. Cet outil pose cependant des problèmes de tracking avec la caméra et lorsqu'il y a des mouvements brusques. Il faudrait ainsi revoir la technique de capture d'information de mouvement afin d'obtenir quelque chose de plus cohérent dans le temps.

3. Peindre une géométrie animée de manière procédurale

Malgré le contrôle que permettent les outils développés au cours de l'étape précédente, il demeure certains problèmes concernant la cohérence du mouvement. Ces problèmes se révélaient particulièrement lors des mouvements de caméra ou lorsqu'un

objet se déplace d'une manière trop brusque. J'ai ainsi décidé de tenter une autre approche en me basant directement sur l'espace 3D et les informations liées aux modèles de la scène et en privilégiant ainsi la cohérence temporelle. Les effets travaillés précédemment s'appliquent essentiellement en deux dimensions à partir d'informations provenant du rendu de la scène sous forme d'image. Ici, nous allons tenter plusieurs méthodes qui consisteront à réaliser les effets expressifs directement dans l'espace en trois dimensions en se basant sur les géométries présentes dans la scène.

3.1 Expérimentations en précalculé

3.1.1 Méthode manuelle par placement de plans

J'ai commencé par expérimenter manuellement la création d'effets de matière et de peinture en plaçant des plans texturés autour d'une modélisation 3D. Je me suis inspirée de l'artiste Miki Bencz qui réalise des tableaux en 3D en peignant à la main ses modélisations. Il place en plus des géométries par dessus la surface de son modèle qui sont rendues avec des textures transparentes et qui permettent de casser l'aspect continu de la géométrie en générant des couches de peinture. J'ai donc créé un *mesh* que j'ai texturé avec des brosses d'aquarelle sur un logiciel de dessin. Ensuite, j'ai modélisé une série de géométries planes que j'ai placées et déformées avec des outils de *sculpting* de manière à ce qu'elles épousent la forme du *mesh*. Je leur ai également attribué différentes textures de brosses. Enfin, j'ai assigné un *shader unlit*, c'est à dire de réagissant pas à la lumière, à ma modélisation et j'ai effacé en partie les bords de la texture pour donner un aspect brute et inachevé au modèle. Le résultat final ressemble à une forme de croquis ou d'esquisse à l'aquarelle. J'ai également complété ce rendu en ajoutant un léger effet de *post-processing* créé lors de la première expérimentation pour donner une apparence moins nette à l'objet. Le point fort de cette méthode est qu'elle est basée sur la géométrie des *meshes* et qu'elle est donc cohérente dans le temps. Cependant, elle est difficilement adaptable artistiquement parlant car la lumière n'est pas dynamique et est choisie par la manière avec laquelle l'artiste a peint la texture. L'intention première ne peut donc être modifiée et l'objet final ne peut pas évoluer par la suite en temps réel si l'on souhaite changer l'ambiance ou les couleurs de l'environnement. Un autre problème est que cette technique est longue et assez fastidieuse et donc difficilement exploitable en production de film d'animation.



Illustration 94 - Création d'effet d'encre de manière manuelle par placement de plans dans Unity

3.1.2 Méthode par motion blur

Parallèlement à cela, j'ai exploré certaines techniques existantes en rendu pré calculé servant à obtenir un rendu expressif en utilisant la géométrie 3D. J'ai ainsi tenté de voir ce qu'on pouvait obtenir à l'aide de la méthode *Painting with polygons*⁷⁰. Elle a été inventée par Isaac Botkin, un article à ce sujet a été publié au SIGGRAPH en 2009. Cette technique peut être réalisée dans la majorité des logiciels de 3D. Elle sert à créer un effet de peinture à la main sur des objets qui est cohérente dans le temps sans avoir à recourir à des algorithmes complexes. Pour cela, on utilise des fonctions de déplacement, ou *displacement* par rapport aux normales des points pour déformer la surface d'un objet 3D. Cette déformation s'effectue un certain nombre de fois entre chaque frame de manière répétitive et cyclique afin de garder une cohérence d'une frame à l'autre. L'effet va ensuite se révéler grâce à l'utilisation du *motion blur*, ou flou cinétique. Il s'agit d'un type de flou qui se base sur le mouvement apparent de l'objet. Il est visible lorsqu'on prend en photo un objet trop rapide ou lorsque l'on a employé un temps d'exposition trop long. En activant le *motion blur*, on voit apparaître les différentes déformations sous forme de couches translucides autour de l'objet. Cela permet d'imiter l'apparence aléatoire et approximative des coups de pinceau. J'ai tenté de reproduire cet effet sur Houdini avec un point VOP, un type de node qui consiste en un graphe permettant de manipuler les attributs des points d'une géométrie en entrée. J'utilise une fonction générant un noise pour déplacer les points et les normales à chaque inter frame. Les images de test ont ensuite été rendues avec le moteur Mantra. J'ai

⁷⁰ « Painting with Polygons ». Consulté le 6 mai 2019. <http://www.isaacbotkin.com/siggraph/>.

essayé de créer un *shader* avec des ombres tranchées pour renforcer le contraste entre les parties peintes du *mesh*. Il est aussi possible de rendre certains types de matière à l'aide du *bump mapping*. Cela requiert donc un travail au niveau du matériau appliqué à l'objet ainsi que du *lighting* de la scène. Les essais réalisés furent intéressants mais difficilement reproductibles dans un moteur temps réel car cela demande de calculer plusieurs samples ou échantillons de positions des points de la géométrie à chaque frame ce qui alourdit le temps de calcul. J'ai ainsi préféré expérimenter d'autres méthodes plus adaptées qui privilégient l'optimisation.

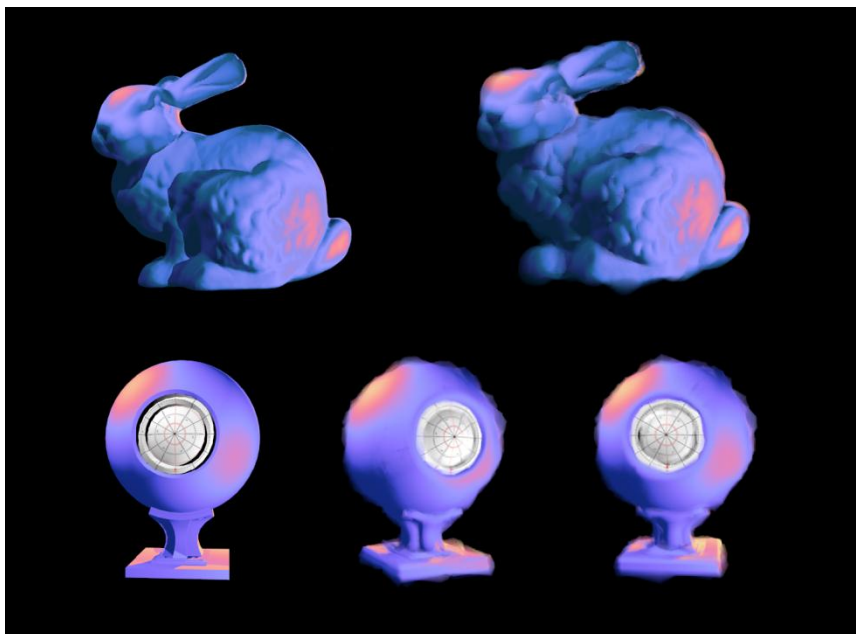


Illustration 95 - Méthode Painting With Polygons testée sur houdini rendue avec Mantra

3.1.3 Méthode par utilisation de techniques de *fur*

J'ai donc effectué d'autres essais, cette fois-ci sur Maya, afin de tenter recréer des styles de dessin ou de peinture par rapport au maillage de l'objet. Pour cela, j'ai utilisé un plug-in de génération de fourrure. Les instances de fur récupèrent le *shading* de l'objet et peuvent être contrôlées par divers paramètres comme la longueur, la densité, l'épaisseur à sa base ou son extrémité, le taux d'enroulement ou l'opacité ainsi que l'aléatoire. Pour que le rendu soit efficace, les touches de peinture sont orientées de manière à toujours être face à la caméra. Cela permet de générer un certain nombre d'effets différents.

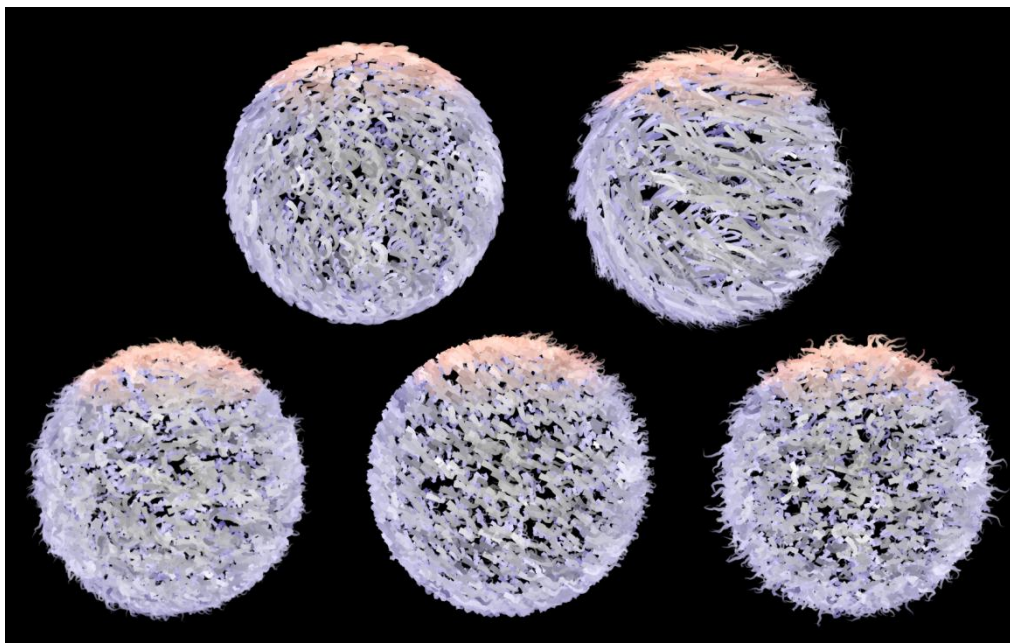


Illustration 96 - Test de fur dans Maya

D'autres graphistes avaient déjà tenté de créer un effet 2D ou pictural à l'aide des outils de fur, comme Neil Stubbings⁷¹ qui se sert du fur et du noise pour créer ses effets dans cinéma 4D comme on peut le voir sur le feuillage de ses arbres.

3.2 Instanciation sur mesh 3D

J'ai donc décidé de réaliser un dernier outil en m'inspirant principalement de la méthode de fur ainsi que de la technique de peinture à partir de plans texturés. Ainsi, je conserve un système de strokes, un peu comme celui vu en deuxième partie, mais cette fois-ci je les place directement sur la géométrie 3D de la scène. J'ai ensuite exposé différents paramètres dans l'interface utilisateur pour que les artistes aient la possibilité de générer eux même les *strokes* en temps réel sur les objets qu'ils souhaitent.

3.2.1 Récupérer les positions des points sur un mesh après déformation du skinning

Le principe consiste donc à instancier un certain nombre de *strokes* sur un modèle 3D. Il s'agit d'abord de récupérer certaines informations telles que les positions et normales des vertices dans une liste et les indices des points permettant de reconstituer les triangles

⁷¹ « Neil Stubbings Shows How He Makes Painterly 3D in C4D ». *Lesterbanks* (blog), 27 septembre 2018. <https://lesterbanks.com/2018/09/neil-stubbings-painterly-3d-c4d/>.

des faces afin de pouvoir par la suite utiliser ces données pour placer et orienter les instances. Comme on est dans le domaine du film d'animation, il faut que les strokes puissent suivre le mouvement d'un objet ayant été animé par un rig. On doit donc pouvoir obtenir les attributs de chaque point après qu'il ait été transformé par l'étape de skinning. Pour cela, j'ai créé un script qui récupère le composant `SkinnedMeshRenderer` du game object avec lequel je peux obtenir le *mesh* en faisant appel à une propriété de la classe ainsi que les joints avec lesquels est effectué le skin de ce *mesh*. On récupère également un tableau de matrices grâce à `mesh.bindposes` qui permet de transformer chaque joint vers sa pose de repos, c'est à dire la pose du joint au moment où *mesh* est passé par l'étape de skin, généralement la T pose ou la pose en V. C'est utile pour obtenir la transformation de chaque joint par rapport à cette pose initiale. Ensuite, pour chaque vertex du *mesh*, l'algorithme calcule une déformation en fonction de tous les joints ayant une influence sur ce vertex selon un poids attribué par le skin et additionne les résultats obtenus afin retrouver la position finale du point à la frame courante. On effectue aussi des opérations de matrices pour que les positions obtenues soient bien dans le repère monde de la scène.

3.2.2 Répartir les instances de manière efficace, scattering

Une fois que l'on est parvenu à obtenir les positions et les normales des points ainsi que les informations des triangles, on peut commencer à essayer de placer des instances. Cette étape s'effectue avec un *compute shader* afin de gagner en performance. J'avais déjà tenté de réaliser cette procédure via le CPU en début d'année, mais les temps de calculs étaient beaucoup trop élevés et rendaient la scène ingérable. Afin de placer les instances, j'ai d'abord tenté tout simplement de les positionner sur les différents points du *mesh*. Un problème se pose alors: les instances ne sont pas du tout réparties uniformément et sont concentrées là où il y a le plus de subdivisions. De plus, elles se positionnent sous la forme d'une grille car elles suivent le maillage du *mesh*. Cela donne un aspect artificiel et peu esthétique.

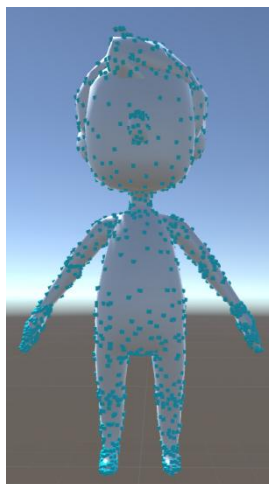


Illustration 97 - Placement de points en bleus sur un skeletalmesh téléchargé dans l'asset store d'Unity

On voudrait pouvoir apporter un peu plus d'aléatoire en faisant en sorte que l'instance soit placée sur la surface de la géométrie sans pour autant qu'elle ne se positionne exactement sur l'emplacement de chaque vertex. J'ai donc décidé d'instancier les *strokes* sur les faces plutôt que les points afin de pouvoir répartir au mieux les formes et gérer la densité. Les instances se répartissent avec un random uniforme sur la surface du triangle suivant l'algorithme *triangle point picking*⁷². Pour cela, j'utilise une matrice qui permet d'obtenir les coordonnées dans le repère d'un parallélogramme formé par le triangle. Ensuite, je génère des positions aléatoires entre 0 et 1 sur les axes x et y dans cet espace 2D grâce à une fonction de pseudorandom, puis je vérifie si le point est présent sur la surface du triangle. Si ce n'est pas le cas, je le rejette et en calcule un nouveau.

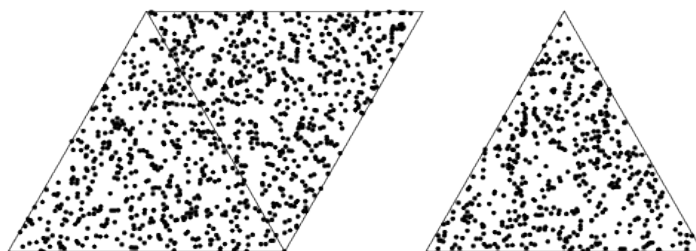


Illustration 98 - Triangle point picking

Malheureusement, avec cet algorithme, on se retrouve avec autant de formes instanciées sur chaque face, peu importe sa taille. J'ai donc fait en sorte que le nombre de *strokes* dépende de la taille du triangle en multipliant le nombre d'instances par face en fonction d'un facteur lié à l'aire du triangle. Les positions relatives à l'espace du triangle sont

⁷² Weisstein, Eric W. « Triangle Point Picking ». Text. Consulté le 6 mai 2019.

<http://mathworld.wolfram.com/TrianglePointPicking.html>.

stockées dans un tableau et permettent à chaque frame de calculer la position finale de l'instance en fonction de la position des sommets du triangle dans l'espace monde. Nous sommes obligés d'effectuer cette opération car nous ne voulons pas que le calcul d'aléatoire de la position de l'instance sur le triangle soit effectué à chaque frame. Au contraire, il doit être réalisé une seule fois à la première frame en initialisation afin que les positions soient cohérentes d'une frame à l'autre.

3.2.3 Orientation des *strokes*

Après avoir placé les formes, je les ai orientées en fonction de la normale de la face, mais le rendu n'était absolument pas esthétique et il y avait une forme d'interpénétration. De plus, on sait que les matériaux en peinture s'appliquent généralement sur une surface plane. J'ai donc choisi d'orienter les coups de pinceau face à la caméra tout en conservant les informations de normales pour orienter la stroke sur les axes horizontaux et verticaux pour la faire plus ou moins s'aligner au contour de l'objet selon les préférences de l'utilisateur. Il pourra aussi ajouter de l'aléatoire pour contrôler la direction de ces strokes. J'ai créé également une distance entre la stroke et la surface de la géométrie par rapport aux normales des faces pour former une sorte de "coquille" d'instances qui entoure la géométrie.

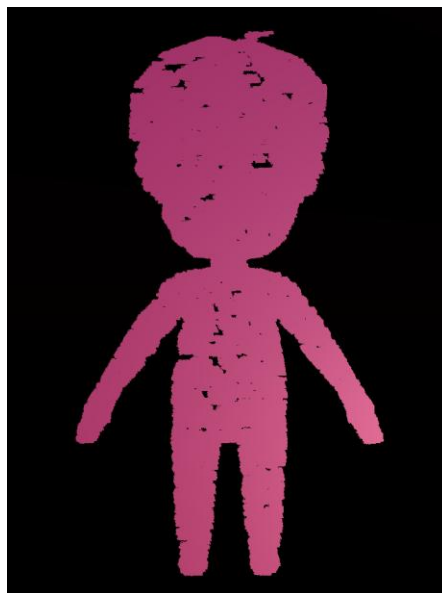


Illustration 99 - Répartition des strokes et alignement face à la caméra (les strokes ne réagissent pas encore à la lumière)

3.2.4 Apparence des *strokes*

Après avoir placé les *strokes*, il faut également penser à leur aspect et leur couleur. Comme en deuxième partie, des masques de transparence sont appliqués au quad pour créer une texture de brosse. Concernant la couleur, il fallait échantillonner la couleur de la texture du *mesh* à l'endroit où se situe l'instance sur la surface. Pour l'instant, cette étape s'effectue via le CPU par la méthode du raycasting: on envoie un rayon depuis l'instance vers la surface du *mesh* pour trouver le point le plus proche placé derrière l'instance sur ce *mesh*. Ensuite, on récupère les positions de ce point dans l'espace des UVs et on peut ainsi échantillonner la texture pour récupérer la couleur du *mesh* à cet endroit précis et l'attribuer à la *stroke*. Cette opération s'effectue une seule fois au lancement de l'application car les *strokes* ne se déplaceront pas sur la surface de l'objet et la couleur de la texture échantillonnée sera donc constante dans le temps. Malgré tout cette technique n'est pas forcément très optimisée et il faudra par la suite trouver un moyen d'obtenir précisément les couleurs des textures d'une manière plus performante. Il serait peut être également envisageable de créer un *compute shader* qui effectuera le raycasting via le GPU. La couleur finale est ensuite calculée à partir d'une opération de color dodge qui prend la couleur de la texture, ainsi qu'une teinte et une opacité donnée en entrée par l'utilisateur. Il est aussi possible de gérer l'aléatoire au niveau de l'opacité de la *stroke*. Le shader attribué aux *strokes* étant un *surface shader*, il réagit également à la lumière et il est possible par exemple d'ajouter une texture de normale ou de *bump* à celles-ci pour créer des effets particuliers et de gérer la réaction de la surface à la lumière.

Concernant leur taille, l'artiste a la possibilité de réduire la taille des *strokes* qui sont sur le contour des objets afin de rendre la forme des géométries plus compréhensible. Cela se réalise en calculant l'angle entre la normale de la face courante et la direction de la caméra. Plus la normale est face caméra, plus la *stroke* sera grande. Il y a aussi des paramètres globaux qui permettent de gérer la taille de toutes les *strokes* et d'apporter de l'aléatoire. Je me suis également servi de ces calculs pour optimiser le programme en effectuant du culling: si la normale ne fait pas face à la caméra, alors les *strokes* appartenant à cette face ne seront pas instanciés.



Illustration 100 - Exemple de rendu sur un mesh d'aigle animé dans Unity

3.2.5 Gérer la lumière

Enfin, j'ai laissé à l'utilisateur la possibilité de séparer les zones d'ombre et de lumière en calques pour qu'il puisse gérer des paramètres spécifiques à ces zones et décider par exemple de travailler la lumière d'une manière spécifique, avec différentes formes, couleurs ou tailles de *strokes*. Pour déterminer l'appartenance d'une stroke à un calque, j'ai calculé l'angle d'une lumière par rapport à la normale du triangle. Cette opération s'effectue différemment selon le type de la lumière, selon qu'il s'agisse par exemple d'une point light ou d'une *directionnal light*. J'ai créé une variable normalisée qui contient le niveau de lumière pour chaque stroke et c'est de ce niveau de lumière que dépend l'appartenance de la stroke à un calque précis.

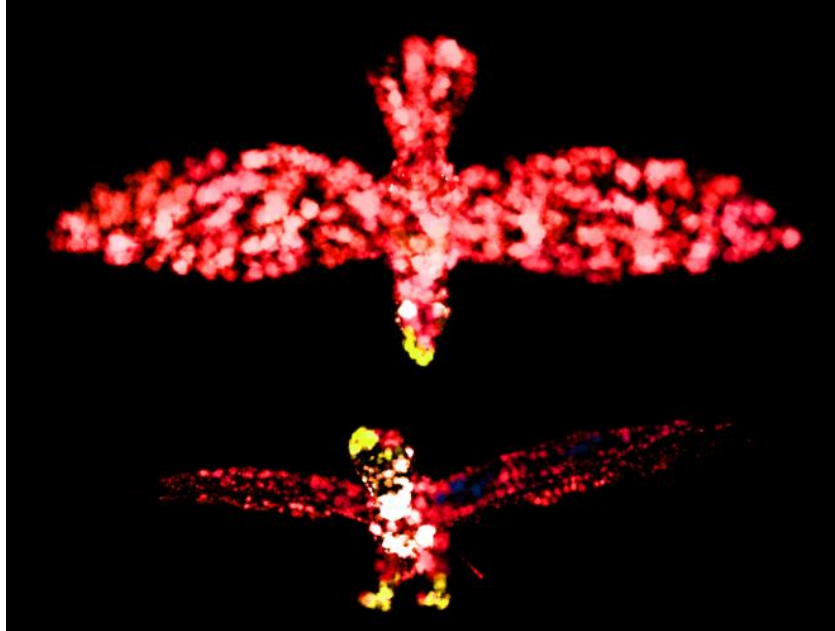


Illustration 101 - Instanciation de strokes avec émission sur un modèle d'aigle téléchargé sur l'asset store d'unity

3. Retour d'expérience

L'outil n'est pas encore parfaitement optimisé, mais il permet tout de même de peindre des *meshs* animés de manière automatique et procédurale, ce qui peut aider à faire gagner beaucoup de temps aux artistes. Le fait de baser le placement des *strokes* sur une géométrie 3D nous laisse obtenir une bonne cohérence temporelle adaptée à l'animation. De plus, la gestion de la lumière se révèle utile car elle permet aux artistes de gérer eux même les contrastes et les ambiances et de travailler l'effet de la lumière sur ces coups de pinceau virtuels. Il peut donc personnaliser l'effet et ajouter par exemple ses propres masques pour donner un style propre à son rendu. L'outil n'est cependant pas complet. Il serait par exemple intéressant de travailler plus sur l'orientation des *strokes* pour leur donner une intention par exemple en les orientant vers les zones les plus lumineuses pour mettre en valeur le *lighting* de la scène. On pourrait également tenter d'utiliser des *strokes* à plusieurs points d'ancrage qui s'étaleraient sur le *mesh* en plusieurs points pour permettre aux utilisateurs de varier la longueur des touches de pinceau en créant par exemple des vagues ou des ondulations. Par la suite, il serait possible d'essayer de combiner cette méthode avec celles vues précédemment dans le but, par exemple, d'apporter plus de mouvement en déplaçant les *strokes* sur la surface des géométries pour créer de nouveaux

effets visuels animés. Enfin, certains bugs de *clipping* et de couleur sont encore à corriger si on veut que l'outil soit réellement utilisable dans le cadre d'un projet non expérimental.

III Réalisation artistique : travail sur la matière céleste par la couleur et la lumière comme sources d'expression et d'immersion

1. Un film immersif

1.1 Présentation du projet

La Tête dans les étoiles est un film d'animation en 180° d'une durée de deux minutes environ en rendu temps réel réalisé sur Unity destiné à être visualisé sur téléphones et tablettes. Le film met en scène un astronaute qui se décline sous différentes formes selon l'imagination de deux personnages allongés sur le sol qui le regardent et le décrivent. Le spectateur est étendu sur l'herbe entre ces deux personnes et a la possibilité d'explorer le ciel au dessus de lui afin de suivre leurs histoires. La vidéo est cyclique, c'est à dire qu'elle se déroule sous la forme d'une boucle qui revient à son point de départ. Ce projet a été réalisé dans le cadre universitaire sur une période de trois semaines et a été achevé en coopération avec Gaëtan Henry. Je me suis principalement chargée de la partie *shading, lighting* et environnement tandis que Gaëtan a plutôt travaillé sur les personnages, l'animation, ainsi que les effets spéciaux, dont notamment les particules.

Le film peut être visualisé via un casque de réalité virtuelle comme l'oculus rift mais le but principal est de pouvoir le visionner sur téléphone. Cela est possible si celui-ci dispose d'un accéléromètre et d'un gyroscope, ce qui est le cas de la majorité des téléphones construits de nos jours. Nous avons fait ce choix car nous voulions qu'un maximum de personnes puissent y avoir accès de n'importe où et le matériel de réalité virtuelle est souvent onéreux, peu accessible au plus grand nombre et nécessite souvent de se rendre dans des espaces dédiés. L'utilisation du téléphone permet ainsi à la majorité des gens d'expérimenter l'œuvre à l'endroit où ils le souhaitent. Ils pourraient ainsi, par exemple, s'immerger dans la vidéo le soir après s'être étendus sur leur lit, ce qui inclut l'œuvre dans l'espace intime du spectateur. Il est également conseillé d'avoir à sa disposition un casque stéréo pour profiter de la spatialisation du son.

1.2 Références artistiques

Pour réaliser ce projet, nous avons été principalement inspirés par la série de vidéos *J and B Rare Stories*⁷³ en 360° postées sur la plateforme youtube. On voulait exploiter les thèmes de l'espace et du flottement afin de générer un mélange de styles avec une association d'éléments dessinés en contour et d'effets spéciaux de construction et déconstruction. On s'est basé sur le travail de différents artistes et illustrations pour définir le type de rendu souhaité, dont notamment James R Eads. Au niveau visuel, nous souhaitons jouer sur l'application d'éléments colorés vibrants et lumineux sur fonds sombres pour créer des jeux de couleur et de contraste. L'emploi de contours luisants et d'effets de nuages et de particules permettent de simplifier le trait et de guider la narration en apportant une dynamique et en mettant en valeur l'animation. Environ un mois avant le commencement du projet, nous avons à deux élaboré scénario dans le but de mettre en place le dialogue des deux personnages et ainsi de prévoir les différentes péripéties que devra affronter l'astronaute.



Illustration 102 - Illustration animée de James R Eads

1.3 Objectifs techniques et esthétiques

L'enjeu de ce projet était pour moi d'expérimenter le développement de *shaders* dans Unity à des fins stylisées et expressives. Une partie de mon travail consistait en une forme

⁷³ JB France. *J&B Rare Stories : Episode 1*. Consulté le 29 décembre 2018.

<https://www.youtube.com/watch?v=KmQ7VLQPLjw>.

d'exercice de style portant sur le rendu de ciel étoilé : j'ai généré différents types de ciels allant du rendu plutôt réaliste au stylisé et à l'abstrait afin de produire une diversité d'ambiances exprimant les différentes personnalités du personnage principal qui se succèdent au cours de l'histoire. Cela m'a donc amenée à travailler la matière céleste dans le but de créer des effets visuels qui serviront à accompagner la narration. J'ai également travaillé sur des *shaders* de personnage et certains effets spéciaux en temps réel pour habiller l'environnement et guider le regard du spectateur afin de l'amener à être immergé dans l'histoire. L'emploi de ces effets expressifs et non réalistes revêt un rôle dans la réception finale de l'œuvre car ils permettent d'une part simplifier certains aspects pour donner à voir l'essentiel en amplifiant les éléments les plus pertinents. En effet, but ici n'était pas de tout montrer grâce à un rendu photoréaliste mais au contraire de suggérer et d'illustrer les histoires que se racontent les deux personnages dans l'herbe et les sentiments qui s'en dégagent en laissant une part d'interprétation au public. D'autre part, ils servent à transporter le spectateur dans un espace immersif vivant et coloré pour l'amener ainsi à ressentir diverses palettes d'émotions par le biais de l'exaltation des sens. Le résultat consiste ainsi en une expérience narrative, visuelle et sonore.

2. Le rendu de ciel étoilé comme exercice de style

2.1 Historique et traitement du ciel de nuit en art traditionnel

Un des principaux défis de ce projet fut la création de l'environnement immersif qui consiste principalement en une voûte céleste étoilée surplombant une prairie. Le ciel de nuit a été interprété de diverses manières au fil des époques et des mouvements artistiques. C'est ce que j'ai pu constater lorsque je me suis rendue à l'exposition *Au-delà des étoiles, le paysage mystique de Monet à Kandinsky* au musée d'Orsay en 2017. Certains artistes voient dans le ciel une quête spirituelle vers l'infini ou la perfection comme Wenzel Hablik. D'autres comme les expressionnistes, le présentent comme étant une métaphore des angoisses et des tourments humains tel qu'on peut l'observer dans *Nuit Etoilée* d'Edward Munch et également dans celle de Van Gogh. Il peut aussi servir à représenter une forme d'intimité, de paysage intérieur, les peintres symbolistes y font souvent appel car elle apporte un aspect onirique et irréel. Certains artistes perçoivent le cosmos comme étant une zone à explorer et le peignent en exprimant un désir de connaissance et de recherche de l'inconnu, ils

questionnent notre place au sein de l'univers. La nuit est souvent utilisée par le surréalistes comme dans *l'Homme de sable* de Dali car elle permet d'appréhender le monde différemment.



Illustration 103 - Nuit Etoilée - Edvard Munch (1922, 1924)

Illustration 104 - L'Homme de sable - Salvador Dali

Illustration 105 - Nuit Etoilée - Wenzel Hablik

Dans l'obscurité, nos perceptions sont différentes et nos limites en terme de vision nocturne, notamment dans le domaine de la vision des couleurs, nous poussent à employer l'imagination pour compléter l'information manquante. Ainsi, l'inspiration du ciel de nuit a permis aux artistes d'expérimenter de nouveaux codes en se détachant du réalisme par rapport au modèle. Ils ont pu imaginer de nouvelles formes artistiques par exemple en employant des tons plus intenses et en réinventant la couleur qu'ils ont du mal à percevoir dans ces conditions, ce type de procédé est particulièrement visible dans les peintures de Leonid Afremov. Le ciel peut ainsi prendre des formes abstraites comme dans les tableaux de Georgia O Keefe ou de Arthur Garfield Dove ou stylisées comme dans *Milky Way* de Peter Doig. Cela permet aussi de mettre en avant la lumière par contraste avec l'obscurité du vide et ainsi d'amplifier certains effets lumineux.



Illustration 106 - Milky Way - Peter Doig (1989)

2.2 Création du ciel procédural

Avant de commencer à travailler sur l'environnement, j'ai d'abord dû m'occuper de la mise en place du nouveau pipeline de rendu HDRP (*High Definition Render Pipeline*). Il s'agit d'un pipeline inclus dans le SRP (*Scriptable Render Pipeline*) toujours en cours de développement qui permet d'obtenir de meilleurs rendus graphiques, avec un système de rendu PBR, la possibilité d'utiliser d'un système de graphe pour la création de *shaders* et une architecture de lumière plus optimisée. Il m'a fallu un peu de temps pour comprendre comment faire fonctionner mes *shaders* et commencer à créer un ciel procédural dans ce nouveau système de pipeline que je ne connaissais pas encore. Je me suis aidée des exemples présents dans le projet initial pour développer mes scripts et *shaders* permettant de gérer la *skybox* de la scène.

J'ai d'abord créé une base de ciel étoilé procédural qui peut être modifiée en fonction des besoins au cours du projet et qui devra pouvoir se transformer au grès des métamorphoses de l'astronaute grâce à des paramètres sur lesquels on peut poser des clés d'animation. Pour cela j'ai repris l'algorithme créé par Pablo Roman Andrioli qui permet de générer procéduralement un ciel rempli d'étoiles. Je me suis aidée notamment de la version présente sur l'asset store et je l'ai implémentée pour le pipeline HDRP. J'ai effectué des ajouts, dont par exemple l'emploi de fonctions de bruit dans la volumétrie pour générer des nébuleuses et ajouter de la matière au ciel. Il m'était ensuite plus facile de créer des déclinaisons et d'animer les paramètres afin de générer différentes ambiances en fonction des diverses personnalités de l'astronaute. Je pouvais par exemple choisir d'avoir un ciel contenant des éléments précis et détaillés ou au contraire de réduire le niveau de détail

pour les plans au rendu plus stylisé. Ce ciel procédural fut une des bases principales du projet.



Illustration 107 - Ciel étoilé de base, généré procéduralement grâce à un shader

2.3 Gestion de variantes de ciel afin de produire des effets visuels accompagnant la narration et exprimant les états mentaux de l'astronaute

Je me suis ensuite chargée de réaliser les différentes déclinaisons de ciel en ajoutant des effets non réalistes d'abstraction et de stylisation et en gérant la lumière produite par celui-ci. La difficulté était de réussir à mettre en valeur la narration et les états mentaux de l'astronaute afin que ceux-ci correspondent d'une certaine manière à ce qui est imaginé par les deux personnes allongées sur le sol. Ainsi, les effets générés par le *shader* doivent pouvoir apporter un caractère précis à l'image.

Dans la première scène, on voulait générer une impression de mystère: c'est le commencement de la séquence et le spectateur doit d'abord avoir la possibilité de d'explorer les alentours en déplaçant la tête ou le téléphone autour de lui. Nous souhaitons ainsi provoquer une sensation de calme et d'éveil et le submerger par l'idée de vide sidéral infini en le plaçant devant une voûte céleste bleue sombre. La quasi-absence d'étoiles dans ce ciel l'invite à prendre en compte les éléments se situant autour et à côté de lui. Il peut ainsi constater la présence de deux autres personnes partiellement cachées derrière des brins d'herbe et peut percevoir des arbres et des montagnes aux alentours. C'est par la suite

que le personnage de l'astronaute va se révéler progressivement. J'expliquerai les procédures utilisées pour rendre ce personnage en seconde partie. Des étoiles commencent à scintiller petit à petit. Le rôle de cette scène d'introduction est de mettre le spectateur en état de quiétude pour qu'il puisse facilement se plonger dans l'histoire. Ainsi, la réalisation d'une ambiance globale bleutée revêt une utilité car le bleu est une couleur qui évoque la profondeur et qui permet d'apaiser. Certaines études comme *Effects of Color on Emotions*⁷⁴ de Patricia Valsez et Albert Mehrabian montrent en effet que le bleu a un effet relaxant sur notre corps. Le regard du spectateur est ensuite guidé par les gestes des personnages allongés à côté de lui.

Ensuite, lors de la deuxième scène, le ciel sert à évoquer la personnalité audacieuse du jeune astronaute aventurier. Pour cela, j'ai décidé de le rendre plus abstrait en imitant un effet kaléidoscope afin de suggérer la présence de portes menant vers plusieurs lieux de l'univers à explorer. Les changements de couleur et l'animation dynamique avaient pour but de faire ressentir l'énergie du personnage. Le *shader* est constitué de trois grilles: deux grilles contiennent différentes animations et ont une échelle, une vitesse et une orientation différentes. La troisième grille sert de masque et affiche dans chaque case une des deux grilles. Elle est elle aussi légèrement animée. Il fallait ensuite réaliser les images présentes dans chaque case de chaque grille. J'ai cette fois-ci utilisé le shadergraph d'Unity afin de créer un *shader* simple de galaxie animé en fonction du temps. J'ai utilisé la fonction *twirl* pour déformer les uvs afin de générer un effet de tournoiement. Ensuite, j'ai ajouté un certain nombre de paramètres et j'ai récupéré le code du shadergraph afin de l'intégrer à mon *shader* de kaléidoscope. Il est ainsi possible de décliner les galaxies en fonction de ces paramètres pour gérer la couleur, la forme, la luminance, l'animation, afin d'avoir plus de variantes et moins de répétitions au sein des grilles. Enfin, j'ai créé un masque pour laisser le ciel étoilé visible à certains endroits et donner un aspect translucide aux formes géométriques. Ces effets visuels accompagnent le mouvement du personnage principal. L'excitation générée par toutes ces couleurs chatoyantes qui tournoient au fond du ciel expriment ainsi le caractère pétulant et fougueux et de l'astronaute et traduisent visuellement son état hyperactif.

⁷⁴ Valdez, Patrick, et Amirhossein Mehrabian. « Effects of color on emotions. » *Journal of experimental psychology. General* 123, n° 4 (1994): 394-409.

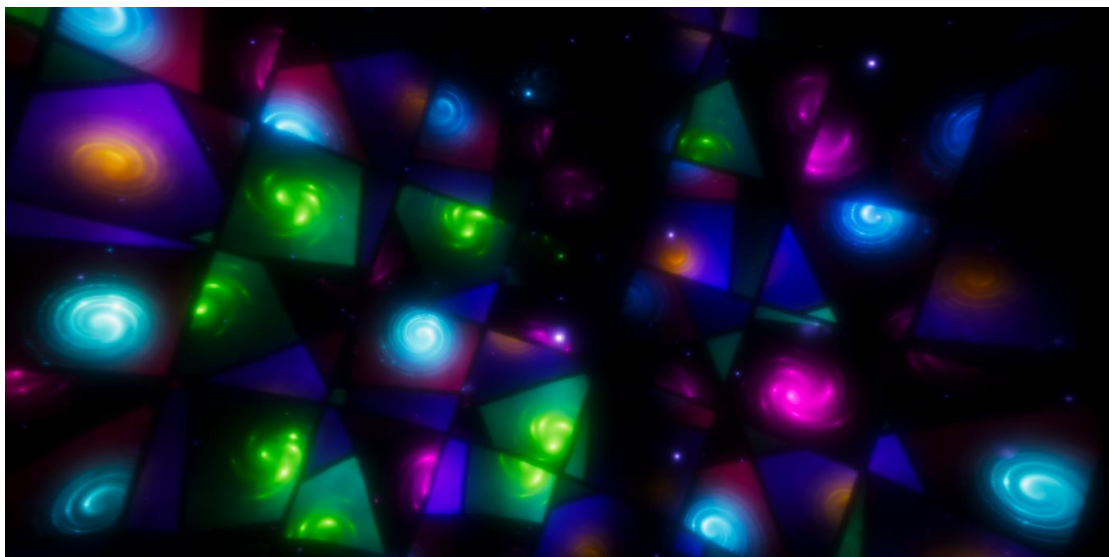


Illustration 108 - Effet de kaléidoscope

Pour les scènes suivantes, j'ai généré un espace plus gazeux et éthéré. Après avoir été aventurier, l'astronaute finit par se métamorphoser en un bébé habillé en scaphandre. Afin d'évoquer la naissance et la découverte du monde extérieur, j'ai disposé des nébuleuses dans le fond du ciel qui adoucissent le décor et imitent les pouponnières d'étoiles, lieux au sein desquels elles naissent et se forment. Par la suite, le personnage effectue une nouvelle métamorphose qui lui donne l'apparence d'un extraterrestre étrange à quatre bras. Pour cela, j'ai essayé de créer une ambiance de couleur dans les tons plus chauds, jaunes orangés, qui engendrent une lumière un peu voilée pour souligner l'étrangeté du personnage. J'ai ajouté des nébuleuses à l'arrière et à l'avant plan grâce à un shader générant une volumétrie grâce à du noise en décalant leur animation dans le temps afin de donner l'impression que le personnage se situe dans une zone complètement différente de l'espace.

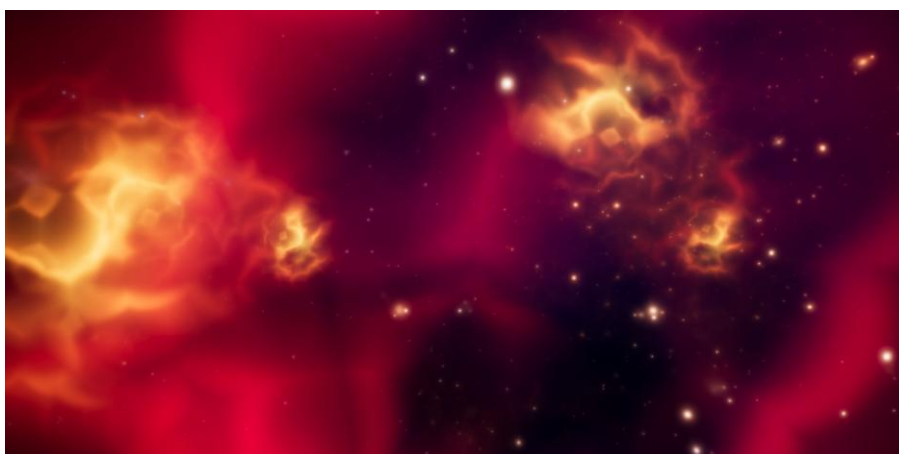


Illustration 109 - Nébuleuses et ambiance orangée

Enfin, pour la dernière partie du film, j'ai développé plusieurs *shaders* dans le but de produire un ciel abstrait, similaire à ce qu'on peut voir dans certaines animations en deux dimensions simplifiées. Ces *shaders* peuvent s'animer autour du personnage qui effectue sa transformation finale. Je me suis inspirée de divers *shaders* présents sur le site Shadertoy et j'ai créé des effets abstraits de vagues, d'ondulation de courbes lumineuses, d'étoiles, et de halos de lumière tournoyant dans le ciel. Le fond est ainsi une combinaison de plusieurs effets générées par les *shaders* et les particules. L'ensemble de ces *shaders* sont appliqués à des sphères qui englobent la scène 3D et ont un niveau de transparence différent pour laisser plus ou moins apparaître le ciel de fond. Il en résulte un climat énergique et spectaculaire, parodiant les séries d'animation de type *Magical Girl*. Ces effets apportent beaucoup à la narration car ils créent un sentiment de suspens et d'attente chez le spectateur et préparent et amplifient la chute de l'histoire. A la fin, les effets se dissipent progressivement et les effets spéciaux réalisés par Gaëtan permettent de faire revenir le ciel à son état initial et de recommencer ainsi une nouvelle boucle.

L'ensemble des images sont également soumises à des effets de *post-processing*. Certains permettent d'apporter de l'anti aliasing et de lisser les éléments. On a en plus ajouté un vignettage, quelques corrections colorimétriques. Il nous a fallu légèrement corriger les niveaux d'exposition qui pouvaient être trop forts au niveau des étoiles les plus brillantes. Enfin, étant donné que la lumière était très importante pour notre projet, on a choisi de l'amplifier grâce à un effet de bloom. Le bloom, ou flou lumineux se base sur un phénomène réel qui apparaît dans la caméra ou notre œil lorsqu'il y a une transition trop forte entre zone lumineuse et zone sombre, ce qui induit un effet de flou autour de cette zone. Cet effet est souvent utilisé dans le domaine du jeu vidéo et peut être réalisé en rendant les zones lumineuses dans une texture HDR afin d'obtenir la plage dynamique de l'image, c'est à dire les valeurs de luminosité les plus hautes sans être limité à des valeurs entre 0 et 1 comme en RGB, puis en appliquant des effets de flou et de *downsampling* à ces zones lumineuses et en les en recombinaison. A la fin, on obtient un effet de halo qui enrichit la qualité esthétique de l'image et permet aux divers objets célestes de ressortir, notamment au niveau des zones ayant de l'émission. Cela rend l'ambiance beaucoup plus expressive et rayonnante. Le bloom permet ici de donner un aspect parfois luisant, parfois

flamboyant aux éléments en leur créant une aura lumineuse. Le résultat de cet effet varie en fonction du *shader* employé et de la quantité d'émission.



Illustration 110 - Ciel de type « magical girl » pour la scène finale

3. Mettre en valeur les personnages grâce au shading et les FX

3.1 Identifier l'astronaute par la couleur

Concernant les *shaders* de personnages, nous avons souhaité attribuer à l'astronaute une couleur reconnaissable qui serait présente d'un personnage à l'autre afin qu'on puisse l'identifier entre chaque transformation. Nous avons choisi le orange car c'est une couleur vive qui se marie bien avec le bleu sombre que l'on retrouve principalement dans le ciel et qui fait ressortir certains éléments du personnage par complémentarité. Cela permet de générer une profondeur via la couleur: les tons rouges étant plus saillants que le bleu du ciel par exemple, les éléments rendus avec ce coloris ressortiront plus dans l'œil du spectateur et donneront la sensation d'être sur un plan plus proche de lui. C'est un procédé souvent utilisé dans la peinture pour indiquer qu'un élément se situe en avant plan. J'ai ainsi créé un *shader* animé en générant une flow map et une texture de lave. J'ai également joué sur l'émission pour faire écho à l'aspect flamboyant du soleil généré en particules par Gaëtan. Ainsi cette couleur se retrouve sur chaque version de l'astronaute : le manteau du premier personnage, les contours de l'aventurier, la couche du bébé, la ceinture de l'alien... Ce

shader permet en plus de faire ressortir certains éléments du personnage pour ainsi guider le regard. Le spectateur ne peut ainsi pas le perdre de vue.

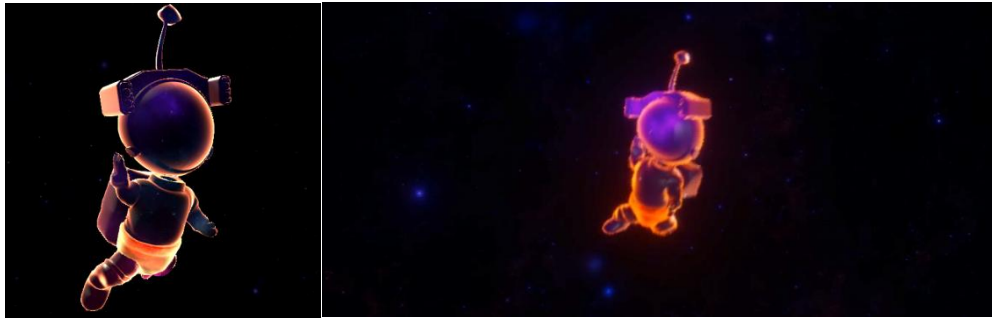


Illustration 111 - Shaders de fresnel et émission appliqués sur l'astronaute bébé

3.2 Constellations

Au début de la séquence, il me fallait faire apparaître un personnage sous forme de constellation. Pour cela, j'ai d'abord tenté de réaliser l'effet avec des particules : j'ai créé un système de particules animées avec un noise et une turbulence. Lorsque deux particules sont suffisamment proches, une ligne se dessine entre les deux pour les relier, ce qui permet de générer des constellations. Cet effet fonctionnait bien si on laissait flotter les particules librement dans l'espace, mais dès que je les faisais émettre du *mesh* de l'astronaute animé, il devenait difficile de reconnaître la forme du personnage. Cela était notamment dû au fait que les segments reliant les particules pouvaient traverser le modèle et le rendaient illisible. J'ai donc créé un *shader* suivant le même procédé : des points sont générés aléatoirement et se déplacent de manière aléatoire sur un plan 2D et des lignes se forment entre eux lorsqu'ils se rapprochent. J'ai ajouté également plusieurs masques pour éviter que tous les points ne se dessinent car cela aurait créé un effet de grille trop régulière sur le personnage. Afin d'avoir une plus grande maniabilité et de pouvoir modifier le rendu à tout moment, j'ai ajouté un certain nombre de paramètres: couleur, brillance, distance minimale, épaisseur des traits... Je les ai animés afin de faire apparaître le personnage progressivement en suivant le timing des voix. Le masque de diffusion m'a permis d'animer l'apparition des points de manière plus aléatoire. J'ai également utilisé ce *shader* pour le ciel afin de générer des formes abstraites et dansantes qui accompagnent le personnage de l'astronaute.



Illustration 112 - Génération de constellations animées

J'ai pu expérimenter d'autres effets intéressants pour les personnages, en créant par exemple un *shader* qui projette une *cubemap* dans le personnage pour générer un effet d'univers intérieur. On a ainsi la sensation qu'un macrocosme constitué de galaxies bouillonne au sein du corps du personnage. Cet effet a notamment été utilisé pour l'astronaute bébé et rappelle la colombe de Magritte ou encore le court-métrage *Day and Night* de Disney. J'ai également joué sur des effets de diffusion et de défragmentation grâce à des noises pour faire apparaître et disparaître des éléments, notamment dans la dernière partie lorsque le personnage retrouve sa combinaison.



Illustration 113 - La grande famille - René Magritte (1963)

3.3 Silhouettes

Le reste des personnages était sensé être réalisé en particules, mais étant donné que sur le nouveau VFX graph de Unity, il n'est pas possible d'émettre des particules à partir d'un *mesh* animé, Gaëtan a du utiliser les particules de l'ancien système Shuriken. Celles-ci étaient moins visibles et il n'était pas possible d'en instancier aussi grand nombre qu'avec le VFX graph car il n'est pas basé sur le GPU. Il fallait donc trouver un moyen via le *shader* de rendre les personnages identifiables sans que le *shader* ne prenne le pas sur les particules.

J'ai ainsi créé un effet de silhouette, grâce en partie au node Fresnel du nouveau shadergraph de Unity. Cet effet a également été appliqué aux deux personnages allongés près de la caméra, car nous souhaitons simplement évoquer leur présence de façon minimaliste afin que l'attention soit portée sur le ciel. Ce *shader* permet ainsi de générer un effet de halo lumineux pour suggérer la forme des éléments présents dans la scène. Il fonctionne à partir du calcul du produit scalaire entre la normale d'une face et la direction de la caméra vers cette face. Obtient ainsi un résultat entre 0 et 1 qui indique par exemple les polygones étant dirigés perpendiculairement à la caméra. Il est ensuite possible d'ajouter des couleurs et de moduler leur diffusion. On peut ainsi choisir d'afficher une couleur sur les bords de la géométrie et une autre sur les faces qui sont le plus au centre et les interpoler afin de créer un dégradé.

3.4 Flat shader

Pour la dernière partie de la séquence, lors de la scène de transformation, il fallait essayer de générer des effets proches de ceux que l'on peut retrouver dans les scènes de transformation de certaines anciennes séries du genre *Magical Girls*. J'ai donc créé un *shader* plat, ne réagissant pas à la lumière, en m'inspirant un peu du *cel-shading* pour le personnage et j'ai réalisé une texture abstraite similaire à celles que l'on peut voir dans les scènes de transformation de séries animées telles que Sailor Moon. J'ai également utilisé la technique de projection de *cubemap* en animant les rotations de celle-ci afin de rendre cet effet scintillant. La texture de *cubemap* tourbillonne à l'intérieur même du personnage sur le point de se métamorphoser, ce qui exprime son état d'excitation et apporte un effet magique à la scène.



Illustration 114 - Flat shader inspiré par Sailor Moon

4. Retour sur les problèmes rencontrés, appréciation globale et améliorations à apporter

Un des principaux problèmes que nous avons eu lors de ce projet fut la difficulté d'avoir un pipeline commun dans Unity. Nous avons décidé, pour éviter les éventuels bugs informatiques, de travailler dans deux projets séparés: d'un côté Gaëtan réalisait le layout dans sa scène et ajoutait les éléments au fur et à mesure. De l'autre côté, j'avais ma propre scène dans laquelle je pouvais expérimenter mes *shaders* et effets sans risquer de corrompre le projet. C'est seulement à la troisième semaine que nous avons tout mis en commun et que certains problèmes se sont révélés.

4.1 Rendu et *shading*

Pour réaliser l'animation, nous avons utilisé des *mesh* téléchargés sur des banques de modèles 3D car nous n'avions pas le temps en trois semaines de tout modéliser à la main. Par conséquent le skin et les uvs des personnages n'étaient pas toujours bons. Gaëtan a consacré une partie de son temps à retoucher certaines topologies et a re déplié certains uvs mais il y avait encore des problèmes sur certains personnages que nous n'avons pas pu résoudre, ce qui fait que certains effets, notamment le *fresnel*, ne pouvaient pas être appliqués sur certains éléments, ce qui était probablement dû à un problème de normales. Il a fallu trouver des alternatives en utilisant d'autres types de *shaders* ou particules afin de rendre malgré tout ces objets.

Un autre soucis lié aux *shaders* est qu'ils sont très vifs et lumineux dans la dernière scène. C'est notamment le cas avec ceux qui ont de l'émission, ce qui a fait que les particules générées par Gaëtan étaient moins visible. Comme nous étions vers la fin du temps imparti, il n'a pas été possible de modifier ce rendu, mais il aurait fallu générer un plus grand nombre de particules et modifier le *shader* de celles-ci pour apporter également de l'émission et ainsi uniformiser les différents éléments.

4.2 Post-processing

Afin de rendre les personnages et les éléments visuels plus brillant et de créer un contraste avec le ciel et l'environnement, nous avons utilisé le bloom du Post process v2

d'Unity. Cependant, certains éléments du fond comme les étoiles étaient trop visibles par rapport à l'avant plan. Une solution serait d'effectuer du bloom sélectif afin d'appliquer différents paramètres à l'avant et à l'arrière plan. Des scripts permettant de réaliser cet effet existent déjà sur l'asset store de Unity mais ne sont pas compatibles avec le nouveau pipeline HDRP. Il aurait donc fallu en créer un par nous même, ce qui aurait été trop long et nous aurait fait perdre beaucoup de temps. Pour l'instant on a simplement baissé le bloom afin d'éviter d'avoir des valeurs trop hautes sur les étoiles, mais le bloom sélectif est une solution à envisager si on souhaite améliorer le rendu. Le deuxième problème lié au *post-processing* était l'apparition de *flickering*, notamment avec certains *shaders* qui parfois renvoient une valeur non valide (NaN) sur un pixel. La solution serait d'effectuer du débog sur certains *shaders* (par exemple vérifier qu'il n'y a pas de division par 0, ce qui en contrepartie peut élever le temps de calcul). L'autre possibilité, moins propre mais plus rapide, serait d'utiliser un NaN killer, un script qui permet de repérer les pixels ayant une valeur NaN et de leur attribuer la valeur d'un pixel voisin afin d'éviter de faire apparaître des taches blanches au rendu.

4.3 Particules

Le fait d'avoir utilisé le pipeline HDRP fut une expérience intéressante mais je ne suis pas certaine que cela ait été une bonne idée pour ce projet. En effet, il y a encore une forte instabilité et la présence d'un certain nombre de bugs étant donné que le pipeline est encore en cours de développement. De plus, cela nous a poussés à devoir apprendre à l'utiliser, ce qui nous a par conséquent fait perdre du temps la première semaine. Les nouveaux outils développés par Unity ne sont pas encore complets, par exemple il n'est pas encore possible d'émettre des particules à partir d'un *skeletal mesh* avec le nouveau VFX graph. Cela m'a déçue car un des enjeux de notre projet était de pouvoir instancier un grand nombre de particules pour créer des effets de traînées de matière. Ainsi, Gaëtan a été obligé d'utiliser l'ancien système de particules Shuriken, ce qui n'était pas prévu au départ.

Pour conclure cette partie, je dirais que la réalisation de ce film immersif m'a permis d'expérimenter un certain nombre d'effets permettant de générer de la matière procédurale expressive. Le travail du *shader* de manière non réaliste offre de nombreuses possibilités

dans le domaine de la création artistique car il permet d'une part d'exprimer et de transmettre des états mentaux en générant des ambiances et en travaillant la lumière et la couleur selon la thématique du ciel étoilé. D'autre part, il peut être utilisé à des fins narratives en animant et en faisant évoluer les paramètres au cours du temps. Il en résulte une œuvre immersive qui plonge le spectateur dans un environnement pittoresque et vivant.

Conclusion

Au fil de ces recherches, nous avons expérimenté diverses manières de générer procéduralement des effets artistiques qui permettent aux artistes de développer un style unique et personnel dans le but d'exprimer une manière propre de voir et de ressentir le monde. L'utilisation du médium numérique fut un moyen de concevoir de nouvelles esthétiques en passant par l'ajout de la dimension temporelle propre au domaine de l'animation ainsi qu'en travaillant sur les possibilités immersives offertes par les technologies actuelles, faisant de l'œuvre numérique une expérience à part entière.

Nous avons vu en première partie que l'expression artistique dépend de l'intuition de l'artiste et consiste en une remise en ordre et une clarification des émotions et impressions internes. Il doit ensuite s'approprier un média et des codes pour, par exemple, donner à voir une émotion particulière et la transmettre à un public par le biais des sens. Cette manière de s'approprier le média s'appelle le style. En étudiant certains mouvements comme l'expressionnisme, le fauvisme, ou encore l'art brut, on a pu constater que l'expressivité d'une œuvre dépendait de divers facteurs. Elle peut ainsi passer par un emploi symbolique ou artistique de la couleur et de la lumière. Les effets de matière et le choix du médium ont également un rôle très important à jouer puisque ce sont eux qui transmettent l'intention ou le message. La répartition du niveau de détail permet également de renforcer cette intention en indiquant au spectateur comment celui-ci doit regarder l'image. Enfin, l'emploi de la stylisation et de la simplification par le biais de distorsions et d'exagérations du réel peuvent amplifier une émotion ou une signification, par exemple par effet d'analogie.

L'enjeu était ici d'exploiter ces notions pour les adapter aux différentes spécificités du médium numérique et aux contraintes du rendu temps-réel. En effet, le temps-réel demande un travail d'optimisation et la programmation sur GPU nécessite une implémentation permettant de faire fonctionner le programme en traitement parallèle, ce qui n'est pas faisable avec n'importe quel algorithme. Les avantages de celui-ci sont qu'ils permettent une interaction directe entre l'artiste et le programme, qui peut immédiatement avoir accès aux résultats produits et génère ainsi une plus grande satisfaction chez l'utilisateur. Après avoir effectué un bref état de l'art des techniques de rendu expressifs existantes et de leur utilisation au sein de plusieurs productions, je me suis consacrée à la recherche et au développement d'outils expressifs inspirés de ces procédés. Ces outils doivent avoir pour fonction de préserver la subjectivité de l'artiste en lui permettant de s'écarter du rendu photoréaliste afin de développer des styles plus personnels et émotionnels.

La technique de rendu par traitement des pixels de l'image est légère et paramétrable et peut permettre de développer et d'hybrider un grand nombre de styles et effets en les combinant entre eux. L'outil ainsi développé laisse par exemple l'utilisateur gérer le niveau global de détail et d'abstraction de l'image pour produire une ambiance particulière. Il peut en plus ajouter ses propres textures pour transformer l'apparence de l'image et également jouer sur des distorsions afin, par exemple, de créer une tension au sein de l'image. Cependant, il ne permet pas un grand contrôle, notamment au niveau local et consiste uniquement en la transformation de valeurs de pixels. Le niveau de personnalisation est encore quelque peu limité bien que les effets générés restent plaisants et facilement utilisables.

J'ai donc par la suite essayé d'autres méthodes, dont la méthode par instanciation de marques ou SBR sur un plan en deux dimensions en m'inspirant de l'art pictural. Cette technique offre beaucoup plus de possibilités et permet un niveau de contrôle élevé. L'artiste a ainsi la possibilité de travailler la couleur directement via les touches de peinture et peut également utiliser ses propres masques afin de définir la forme du pinceau à appliquer. Il a en plus la possibilité de choisir de travailler sur certaines zones de l'image spécifiquement en fonction des informations extraites de la scène 3D telle que les normales des objets, la profondeur ou encore les contours. La limite de l'outil est que le suivi du déplacement est imparfait et que les mouvements de caméra perturbent le placement des

touches qui peuvent s'agglomérer sur certaines zones. L'outil nécessite encore certaines améliorations et optimisations mais demeure une base intéressante pour les personnes souhaitant travailler leurs images de manière plus picturale.

La deuxième variante de l'outil, qui est plus expérimentale, apporte de nouvelles possibilités expressives grâce à l'emploi du mouvement. La peinture traverse l'écran et repeint constamment l'image en temps-réel en générant ainsi des effets de coulure et de dispersion. L'utilisateur peut apporter une intention en variant le style de brosse et en choisissant la manière avec laquelle les flux sont contrôlés. L'apport de la dimension temporelle offre une nouvelle manière de travailler la peinture virtuelle qui peut être combinée avec de l'animation pour créer un effet d'empreinte ou de trace dans l'image et souligner la présence du mouvement.

Enfin, les techniques de peinture automatique de géométries par pose de touches dans l'espace 3D ont permis d'apporter une plus grande cohérence temporelle que celles appliquées sur un plan 2D. Le fait de pouvoir gérer les couches de lumière et d'appliquer cet effet à des personnages animés peut être utile aux artistes car ils ont la possibilité de contrôler l'aspect visuel de chaque géométrie individuellement. Cette méthode pourrait encore être exploitée afin par exemple d'animer les coups de pinceau autour du personnage, ou de gérer différemment le placement et les rotations afin de renforcer les effets de lumière.

La dernière expérimentation, qui consistait en un projet de court métrage d'animation 3D en 180°, fut un moyen de nous aventurer du côté de l'immersion, en mettant en avant certains effets générés par des *shaders*. Le travail de style, d'ambiance et de lumière ainsi que le traitement expressif de la matière céleste ont ainsi pu être utilisés dans le but de mener une narration et de transmettre différentes émotions et sensations au spectateur.

Ainsi, ces diverses techniques permettent d'une certaine manière de préserver la subjectivité de l'artiste en lui offrant plus de libertés et en ne le limitant pas à de simples opérations de paramétrages, comme cela peut parfois se produire en rendu photoréaliste. Etant donné que ces recherches portaient majoritairement sur le temps-réel, il serait intéressant par la suite d'approfondir la conception d'œuvre comme expérience en s'intéressant cette fois-ci aux possibilités offertes par l'interaction. On pourrait ainsi imaginer de nouvelles manières d'interagir avec ces différents effets et amener le spectateur

à expérimenter activement des formes artistiques originales et expressives, selon les désirs de l'artiste.

Bibliographie

Stecker, R., et T. Gracyk. *Aesthetics Today: A Reader*. Aesthetics today. Rowman & Littlefield Publishers, 2010. https://books.google.fr/books?id=UgL_QQAACAAJ.

Dupond, Pascal. *Le vocabulaire de Merleau-Ponty*. Vocabulaire de. Paris: Ellipses, 2001.

Collingwood, R. G. *The Principles of Art*. Read Books Ltd, 2016.

Kobayashi, Chinatsu. « Bosanquet, Collingwood et l'esthétique idéaliste britannique ». *Philosophiques* 36, n° 1 (2009): 149-82. <https://doi.org/10.7202/038014ar>.

Quéré, Louis. *Le modèle esthétique de la communication de John Dewey*. Occasional Paper 19, 2014

Robinson, Jenefer. *Expression and Expressiveness in Art*, Postgraduate Journal of Aesthetics, Vol. 4, No. 2, August 2007

Merleau-Ponty, Maurice. *Phénoménologie de la perception*. Editions Gallimard, 2013.

Ecrits et propos sur l'art (1972), Notes d'un peintre, la Grande Revue, 25 décembre 1908.

Kandinsky, Wassily. *Du spirituel dans l'art et dans la peinture en particulier*. Éditions de Beaune, 1951.

Hubeny, Alexandre. « Le sens de l'histoire dans la philosophie de Merleau-Ponty ». *Annales de l'École pratique des hautes études* 117, n° 113 (2004): 415-20. <https://doi.org/10.3406/ephe.2004.12396>.

Pons, Gilbert, et Maurice Merleau-Ponty. *Le doute de Cézanne : Extrait*. Rochefort: Les petites allées, 2016.

Collingwood, R. G. *The Principles of History: And Other Writings in Philosophy of History*. Édité par W. H. Dray et W. J. van der Dussen. Oxford: Clarendon Press, 2001.

Dewey, John. *L'art comme expérience*. Folio, 2010.

Tolstoy, graf Leo. *What Is Art?* Crowell, 1899.

Deledalle, Gérard. « II. William James (1842-1910). L'expérience pure ». *Le Point philosophique* 3e éd. (1998): 133-45.

Lichtenstein, Jacqueline. *La couleur éloquente : Rhétorique et peinture à l'âge classique*, Nouvelle édition, 2013

Pastoureau, Michel, et Dominique Simonnet. *Le Petit livre des couleurs*. Paris: Points, 2014.

- Alliez, Éric. *L'oeil-cerveau: nouvelles histoires de la peinture moderne*. Vrin, 2007.
- Van Gogh, Vincent. Lettres de à Théo Van Gogh, des 8 et 9 septembre 1888.
- Bonnard, Pierre, Alain Lévêque, et Antoine Terrasse. *Observations sur la peinture*. Strasbourg: L'Atelier Contemporain, 2015.
- Campo, Antoine. *L'Expressionnisme d'hier à aujourd'hui - l'expressionnisme contemporain*.
- Mcluhan, Marshall, et Lewis H. Lapham. *Understanding Media – The Extensions of Man*. Reprint. Cambridge, Mass: MIT Press, 1994.
- Macke, August. *Les masques*. S.l.: Mélanges Editions, 2005.
- Couchot, Edmond, et Norbert Hillaire. *L'art numérique*. Paris: Flammarion, 2009.
- Hurlbutt, Tom. « “Regression” Progression: Non-Photorealistic Rendering Choices in Game Design », s. d., 25.
- Montesdeoca, Santiago E., Hock Soon Seah, Amir Semmo, Pierre Bénard, Romain Vergne, Joëlle Thollot, et Davide Benvenuti. « MNPR: A Framework for Real-Time Expressive Non-Photorealistic Rendering of 3D Computer Graphics ». In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering - Expressive '18*, 1-11. Victoria, British Columbia, Canada: ACM Press, 2018. <https://doi.org/10.1145/3229147.3229162>.
- Haeberli, Paul. « Paint by Numbers: Abstract Image Representations ». In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, 207–214. SIGGRAPH '90. New York, NY, USA: ACM, 1990. <https://doi.org/10.1145/97879.97902>.
- Hertzmann, Aaron. « Painterly Rendering with Curved Brush Strokes of Multiple Sizes ». In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, 453-60. Not Known: ACM Press, 1998. <https://doi.org/10.1145/280814.280951>.
- Huang, Haozhi, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, et Wei Liu. « Real-Time Neural Style Transfer for Videos ». In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7044-52. Honolulu, HI: IEEE, 2017. <https://doi.org/10.1109/CVPR.2017.745>.
- Imhof, Nicolas, Antoine Milliez, Flurin Jenal, René Bauer, Markus Gross, et Robert W. Sumner. « Fin Textures for Real-Time Painterly Aesthetics ». In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games - SA '15*, 227-35. Paris, France: ACM Press, 2015. <https://doi.org/10.1145/2822013.2822021>.

Hertzmann, A. « A Survey of Stroke-Based Rendering ». *IEEE Computer Graphics and Applications* 23, n° 4 (juillet 2003): 70-81. <https://doi.org/10.1109/MCG.2003.1210867>.

技術同人誌 「Unity Graphics Programming vol.1」

表紙グラフィックス作成のためのUnityプロジェクト:

IndieVisualLab/UnityGraphicsProgrammingFrontCover. C#. 2017. Reprint, Indie Visual Lab, 2019. <https://github.com/IndieVisualLab/UnityGraphicsProgrammingFrontCover>.

Valdez, Patrick, et Amirhossein Mehrabian. « Effects of color on emotions. » *Journal of experimental psychology. General* 123, n° 4 (1994): 394-409.

Santella, Anthony, et Doug DeCarlo. « Abstracted Painterly Renderings Using Eye-tracking Data ». In *Proceedings of the 2Nd International Symposium on Non-photorealistic Animation and Rendering*, 75–ff. NPAR '02. New York, NY, USA: ACM, 2002. <https://doi.org/10.1145/508530.508544>.

Webographie

Françoise ARMENGAUD, « **EXPRESSION, philosophie** », *Encyclopædia Universalis* [en ligne], consulté le 6 mars 2019. URL : <http://www.universalis.fr/encyclopedie/expression-philosophie/>. Larousse [en ligne], consulté le 2 mars 2019.

URL : <https://www.larousse.fr/dictionnaires/francais/expression/32326>.

Kemp, Gary. « Collingwood's Aesthetics ». In *The Stanford Encyclopedia of Philosophy*, édité par Edward N. Zalta, Fall 2016. Metaphysics Research Lab, Stanford University, 2016.

<https://plato.stanford.edu/archives/fall2016/entries/collingwood-aesthetics/>.

Guilloux, Robin. « Maurice Merleau-Ponty, Art et Langage ». Le blog de Robin Guilloux.

Consulté le 6 avril 2019. <http://lechatsurmonepaule.over-blog.fr/2018/05/maurice-merleau-ponty-art-et-langage.html>.

« Data Art : une œuvre montre comment s'agite votre cerveau... et c'est beau ! » L'ADN, 20 avril 2018. Consulté le 29 avril 2019. <https://www.ladn.eu/mondes-creatifs/traduire-des-souvenirs-en-oeuvres-dart/>.

Guérin, Michel. « Qu'est-ce qu'un médium artistique ?. Intention et condition ». *Appareil*, n° 17 (29 juin 2016). <https://doi.org/10.4000/appareil.2308>.

AlloCine. *Croc-Blanc BONUS « La fabrication du film »*, réalisé par la Fondation Gan pour le Cinéma. Consulté le 25 avril 2019.

http://www.allocine.fr/video/player_gen_cmedia=19577509&cfilm=248271.html.

« 50 shades of Gris: How Nomada Studio created 'a game that evolves visually, not only mechanically' ». *MCV* (blog). Consulté le 2 janvier 2019. <https://www.mcvuk.com/50-shades-of-gris-how-nomada-studio-created-a-game-that-evolves-visually-not-only-mechanically/>.

« Koro Installation Interactive Battements de Coeur Damien Traversaz ». damien-traversaz.

Consulté le 6 mars 2019. <https://www.damientraversaz.com/koro>.

« Body Paint ». *Memo Akten* (blog). Consulté le 7 avril 2019.

<http://www.memo.tv/portfolio/bodypaint/>.

« L'Histoire du Rendu 3D ». Consulté le 2 novembre 2018. <http://www.3dvvf.com/dossier-383-1-histoire-rendu-3d.html>.

« Loose and Sketchy Animation ». Consulté le 16 mars 2019.

<http://otherthings.com/uw/loose/sketch.html>.

« Watercolor Shader Prototype ». Gumroad. Consulté le 14 novembre 2018.

<https://gumroad.com/l/skybreaker-watercolor>.

« Directing Gaze in 3D Models with Stylized Focus ». Consulté le 16 mars 2019.

https://gfx.cs.princeton.edu/pubs/Cole_2006_DGI/index.php.

Kyprianidis, Jan Eric, et Henry Kang. « Image and Video Abstraction by Coherence-Enhancing Filtering ». *Computer Graphics Forum* 30, n° 2 (avril 2011): 593-602.

<https://doi.org/10.1111/j.1467-8659.2011.01882.x>.

Meier, Barbara J. « Painterly Rendering for Animation ». *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96*. Consulté le 6 mai 2019. https://www.academia.edu/153902/Painterly_rendering_for_animation.

Schmid, Johannes, Martin Sebastian Senn, Markus Gross, et Robert W Sumner. « OverCoat: An Implicit Canvas for 3D Painting », s. d., 9.

« Painterly Renderer Ray Tracer ». Consulté le 28 mars 2019.

<http://www.cs.tufts.edu/~remco/older-stuff/painterly.html>.

« ゲームの花園 ». Consulté le 14 avril 2019. <https://aqu.hatenablog.com/>.

El-Nasr, Magy Seif, Simon Niedenthal, Igor Knez, Priya Almeida, et Joseph A. Zupko.

« Dynamic Lighting for Tension in Games ». *Game Studies* 7 (2007).

(2017), *Pearl*.

Gnomon. *The Making of PEARL with Academy Award-Winning Patrick Osborne*. Consulté le 26 mars 2019. <https://www.youtube.com/watch?v=zZSIhwxLqu0>.

« Unreal Engine 4 Paint Filter Tutorial ». raywenderlich.com. Consulté le 6 décembre 2018.

<https://www.raywenderlich.com/100-unreal-engine-4-paint-filter-tutorial>.

« culdevu - Shadertoy BETA ». Consulté le 30 novembre 2018.

<https://www.shadertoy.com/user/culdevu>.

« Shaders in Unity ». Consulté le 15 novembre 2018. <http://www.shaderslab.com/>.

« The Flow Project: Connecting Colors and Movement | The Metropolitan Museum of Art ».

Consulté le 12 février 2019. <https://www.metmuseum.org/blogs/digital-underground/2015/the-flow-project>.

Agafonkin, Vladimir. « How I built a wind map with WebGL ». Points of interest, 27 juillet 2017. <https://blog.mapbox.com/how-i-built-a-wind-map-with-webgl-b63022b5537f>.

DesLauriers, Matt. « Generative Impressionism ». *Matt DesLauriers* (blog), 9 avril 2014.

<https://medium.com/@mattdesl/generative-impressionism-afa98ccb97da>.

« Trapcode - News - Painterly Particular ». Consulté le 6 mai 2019.

<http://trapcode.squarespace.com/journal/2014/1/30/painterly-particular.html>.

« Painting with Polygons ». Consulté le 6 mai 2019. <http://www.isaacbotkin.com/siggraph/>.

« Neil Stubbings Shows How He Makes Painterly 3D in C4D ». *Lesterbanks* (blog), 27 septembre 2018. <https://lesterbanks.com/2018/09/neil-stubbings-painterly-3d-c4d/>.

Weisstein, Eric W. « Triangle Point Picking ». Text. Consulté le 6 mai 2019. <http://mathworld.wolfram.com/TrianglePointPicking.html>.

JB France. *J&B Rare Stories : Episode 1*. Consulté le 29 décembre 2018. <https://www.youtube.com/watch?v=KmQ7VLQPLjw>.

Filmographie

Laguionie, J.F.,(2011),*Le Tableau*.

Espigares,A.,(2018),*Croc-Blanc*.

(2018), *Baybax Dreams of Too Many Baymaxes*.

(2018), *Sonders*.

(2017), *Pearl*.

Ludographie

Cuevas.A,Roset.A, (2018), *Gris*.

(2015), *Beyond Eyes*.

(2018), *11-11 Memories Retold*.

Table des illustrations

Illustration 1 - Vieil homme dans la tristesse - Vincent Van Gogh (1890).....	18
Illustration 2 - New York movie - Eward Hopper (1939).....	18
Illustration 3 - Jeune homme triste dans un train - Marcel Duchamp (1911).....	18
Illustration 4 - Jaune - rouge - bleu, Vassily Kandinsky (1925).....	19
Illustration 5 - Impression, soleil levant - Claude Monet (1872).....	25
Illustration 6 - Le Cri - Edvard Munch (1893-1917).....	26
Illustration 7 - Le Café de nuit - Vincent Van Gogh (1888)	26
Illustration 8 - La Danse - Henri Matisse (1910)	27
Illustration 9 - Femme endormie - Pierre Bonnard (1928)	27
Illustration 10 - Around the blues - Sam Francis (1957, 1962).....	28
Illustration 11 - Sans titre - Yukio Karkai.....	28
Illustration 12 - Portrait d'un homme - Erich Heckel (1919).....	29
Illustration 13 - Waves breaking against the wind - Joseph Turner (1835).....	30
Illustration 14 - Tableau de Corinne Lamoine.....	30
Illustration 15 - Peinture au doigt - Iris Scott.....	31
Illustration 16 - Walls Bear the Scars of War - Antoni Tàpies	31
Illustration 17 - Schéma représentant l'étape de raytracing	33
Illustration 18 - Oeuf généré procéduralement dans le logiciel de texturing Substance Designer	35
Illustration 19 - Exemple de GUI mal adapté à l'expérience utilisateur	37
Illustration 20 - Le tableau - Jean François Laguionie (2011).....	38
Illustration 21 - Comparaison de style entre le Tableau et la peinture de Modigliani	39
Illustration 22 - Croc-Blanc - Alexandre Espigares - 2018	39
Illustration 23- The Starry night - 360° VR short movie	40
Illustration 24 - Gris - Studio Nomada (2018).....	41
Illustration 25 - Journey - thatgamecompany (2013)	42
Illustration 26 - Botanicula - Amanita Design (2012)	42
Illustration 27 - Beyond Eyes - Tiger and Squid (2015)	42
Illustration 28 - Koro - Damien Traversaz	43
Illustration 29 - Body Paint - Mehmet Akten (2010).....	43
Illustration 30 - First human in orthographic view - William Fetter.....	44
Illustration 31 - Schéma représentant les deux types de rendu: Ray tracing et Rastérisation	45
Illustration 32 - Schéma représentant le pipeline graphique pour OpenGL	46
Illustration 33 - Le GPU possède des milliers de micro processeurs tandis que le CPU en possède moins mais avec une plus grande puissance de calcul	48
Illustration 34 - Transformation de la vue frustum	50
Illustration 35 - Projection de vertices sur un plan 2D	51
Illustration 36 - Etape de rastérisation d'un triangle	52
Illustration 37 - Etape de fragment processing	52
Illustration 38 - Schéma représentant le fonctionnement du double buffering.....	54
Illustration 39 - Exemple de cel-shading sous le moteur Unreal Engine.....	56
Illustration 40 - Technique Loose and Sketchy	57
Illustration 41 - Effet aquarelle sur Unity - Hadyn Lander	57
Illustration 42- Utilisation de la simplification et de la stylisation dans le jeu Wind Waker	58

Illustration 43 - Différence de focus dans la méthode Directing Gare in 3D Models with Stylized Focus	59
Illustration 44 - Création d'effet de peinture par convolution	60
Illustration 45 - MNPR framework dans Maya	60
Illustration 46 - Paint by numbers - Paul Haeberli.....	62
Illustration 47 - Stroke based rendering for 3D animation, Barbar Meier	62
Illustration 48 - Rendu de peinture utilisant le raffinement par couches	63
Illustration 49 Réflexion - Yoshimichi Tamura (2012).....	64
Illustration 50 - Rendu de peinture par ray tracing	65
Illustration 51 - 11-11 Memories Retold (2018)	65
Illustration 52 - Transfert de style	66
Illustration 53 - Méthode de rendu par shell en temps réel dans Unity.....	67
Illustration 54 - Effet de crayon par couches dans Unreal Engine	67
Illustration 55 - Technique utilisée pour Pearl.....	68
Illustration 56 - Baymaw Dreams dans Unity	71
Illustration 57 - Tableau de Marianne Von Werefkin	74
Illustration 58 - Fonctionnement du filtre de Kuwahara	75
Illustration 59 - Image sans filtre - Asset provenant de l'asset store de Unity.....	76
Illustration 60 - Kuwahara avec filtre sans direction	77
Illustration 61 - Kuwahara avec directions	77
Illustration 62 - Effet aquarelle - Passe de couleurs	78
Illustration 63 - Effet aquarelle - Passe d'occlusion	78
Illustration 64 - Effet aquarelle - Rendu dans un style réaliste provenant d'un ancien projet intensif.....	78
Illustration 65 - Effet aquarelle - Rendu après traitement de l'image	78
Illustration 66 - Effet aquarelle - Primitives dans Unity	79
Illustration 67 - Effet aquarelle sur Canyons provenant de l'asset store d'Unity	80
Illustration 68 - Effet sketch dans Unity - Assets provenant de l'Asset Store.....	81
Illustration 69 - Combinaison de plusieurs effets, stylisation et ajout de distorsions.....	82
Illustration 70 - Paysage d'hiver - Vassily Kandinsky (1909)	86
Illustration 71 - SBR - Exemple de forme possible de la stroke après déformation	89
Illustration 72 - SBR Ambiances de couleur.....	91
Illustration 73 - SBR - Exemple de masque pouvant être utilisé	92
Illustration 74 - Matrices de convolution pour le filtre de Sobel	93
Illustration 75 - Exemple de calcul détection de contours effectué au cours des recherches	93
Illustration 76 - Exemple de dégradés pouvant être obtenus par l'utilisation de SDF.....	94
Illustration 77 - SBR Exemple de rendu sur primitives par instanciations semi-aléatoires	95
Illustration 78 - SBR Exemple de rendu sur primitives par instanciations déterminées par les normales	96
Illustration 79 - SBR Image de base	96
Illustration 80 - SBR Instanciations en variant les paramètres de taille et de courbure des stokes.....	97
Illustration 81 - SBR Extraction des informations nécessaires à l'instanciation de strokes	99
Illustration 82 - SBR Autres exemples d'effets que l'on peut obtenir	100
Illustration 83 - Above the grave pit - Emily Carr (1937)	101
Illustration 84 - Carte des vents	101
Illustration 85 - Lion Spirit Guide - Julia Watkins.....	102

Illustration 86 - Generative art - Steve Edwards.....	102
Illustration 87 - Dreamscape - Erin Hanson (2016).....	102
Illustration 88 - Digital painting - Ivana Besevic.....	102
Illustration 89 - Flow Brushes - Qiu Yi Wu (2015).....	103
Illustration 90 - Exemple de flow map.....	104
Illustration 91 - Wind Painter - Rendus sur paysages dans Unity	105
Illustration 92 - Wind Painter - Rendu sur une modélisation d'arbre	106
Illustration 93 - Wind Painter - Effets plus abstraits	106
Illustration 94 - Création d'effet d'encre de manière manuelle par placement de plans dans Unity ..	109
Illustration 95 - Méthode Painting With Polygons testée sur houdini rendue avec Mantra	110
Illustration 96 - Test de fur dans Maya.....	111
Illustration 97 - Placement de points en bleus sur un skeletalmesh téléchargé dans l'asset store d'Unity.....	113
Illustration 98 - Triangle point picking.....	113
Illustration 99 - Répartition des strokes et alignement	114
Illustration 100 - Exemple de rendu sur un mesh d'aigle animé dans Unity.....	116
Illustration 101 - Instanciation de strokes avec émission sur un modèle	117
Illustration 102 - Illustration animée de Jeames R Eads	120
Illustration 103 - Nuit Etoilée - Edward Munch (1922, 1924)	122
Illustration 104 - L'Homme de sable - Salvador Dali.....	122
Illustration 105 - Nuit Etoilée - Wenzel Hablik.....	122
Illustration 106 - Milky Way - Peter Doig (1989)	123
Illustration 107 - Ciel étoilé de base, généré procéduralement grâce à un shader	124
Illustration 108 - Effet de kaléidoscope	126
Illustration 109 - Nébuleuses et ambiance orangée	126
Illustration 110 - Ciel de type « magical girl » pour la scène finale.....	128
Illustration 111 - Shaders de fresnel et émission appliquées sur l'astronaute bébé	129
Illustration 112 - Génération de constellations animées.....	130
Illustration 113 - La grande famille - René Magritte (1963).....	130
Illustration 114 - Flat shader inspiré par Sailor Moon	131