

Université Paris 8

Master Arts

Mention : *Arts Plastiques et Art Contemporain*

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

La Transmutation de la matière dans les effets spéciaux.

Le transfert de particules dans les simulations Realflow.

Quentin ROSSETTO

Mémoire de Master 2

2013-1014

EXTRAIT :

Le sujet principal de ce mémoire est la transmutation de la matière via l'utilisation de systèmes de particules.

Il met en avant les solutions techniques appliquées au cours de multiples tests afin de procéder au transfert de particule d'un système de particule à un autre.

La reproduction exacte d'un phénomène réel en 3D n'est pas le but primordial de ce mémoire. L'intérêt est de prouver techniquement qu'un tel transfert est possible, et d'offrir aux artistes une base technique afin de les aider dans leurs projets personnels.

Ce mémoire, après un état de l'art composé d'œuvres ayant attiré au changement d'état, évoque de manière succincte l'histoire de la relation des effets spéciaux et des systèmes de particules.

Une courte présentation technique vous initiera par la suite au logiciel sélectionné afin de répondre à la problématique.

Enfin, les tests effectués au cours de cette année seront individuellement présentés, et une analyse en profondeur des solutions techniques les plus efficaces vous sera offerte afin de vous permettre de mettre en œuvre vos propres expérimentations.

ABSTRACT:

The main goal of this research paper is the transmutation of matter via particles systems simulation.

Technical solutions will be explored to achieve particles transfer from one system to another. Exact copies of real world phenomenon to 3D is not the goal of this research; rather the fundamental interest is to prove it is technically plausible, and to offer to artists a technical solution atop which they will be able to achieve their own personal projects.

This research paper consists firstly of a selection of art representing matter transmutation. This is followed by a short history of particle simulation in visual effects.

A technical introduction is provided to acquaint the reader with the software selected to answer this paper's goal.

Finally, tests made during this year of research will be presented, followed by an in depth explanation of the most advanced technical solutions utilized to achieve matter transmutation.

De tout mon Cœur, je tiens à remercier l'équipe pédagogique d'ATI, Marie Helene Tramus, Chu-Yin Chen, Anne-laure George-Molland, Cédric Plessiet pour leur cours et leur soutien au cours de mes études à Paris 8.

Merci à mes amis et à tous les satellites du Crou, la 3D ce n'est pas facile, heureusement que les potes sont là pour faire baisser la pression, et monter le son.

Enfin, un grand merci à Gustavo Sanchez, Realflow Product Manager chez Next Limit Technologies, pour sa confiance et l'opportunité professionnelle qu'il m'offre. Quelle meilleure reconnaissance de votre travail que d'aller travailler pour la société produisant le logiciel utilisé pour vos recherches ?

TABLE DES MATIERES

Introduction :	8
PREMIERE PARTIE : Etat de l'art et systèmes de particules dans les effets spéciaux:	9
Le changement d'état dans l'art:	9
Death to the Death DDeath to the Death penalty, réalisé par Pleix et Digital District	9
Creep, Skye Kelly :	10
la persistance de la memoire, salvador dali:	11
Expansion de césar :	11
Nele Azevedo, Melting Men :	12
Spider-Man 3 :	13
TERMINATOR 2 – James Cameron :	14
Historique des système de Particules dans les effets spéciaux :	15
Seconde Partie : La Simulation De Particules Dans Realflow :	16
Présentation de Next Limit :	16
Realflow :	16
XFlow :	16
Maxwell Render :	16
INTRODUCTION A REALFLOW :	17
Types de particules et Paramètres internes:	17
Interaction des particules et de l'environnement, Les daemons :	20
La Géométrie :	23
Le Relationship editor :	23
La programmation dans realflow :	24
L'import, export et la compatibilité	24
TROISIEME PARTIE : Maitriser le logiciel, tests divers et variés :	26
HYBRIDO 2.0:	27
Research paper test 02 : Waterball_01 :	27
Research paper test 03 : Hybrido_01 :	28
AUTRES EXPERIMENTATIONS:	29
Research paper test 04: Bubble_01:	29
Research Paper Test 05 : Skull_01:	30
Research Paper Test: Texture Emission:	31
Research Paper test 08: WetMap:	32
Research Paper Test 10, Filaments:	34
Research Paper Test 12, Scripting:	35
Research Paper Test 13, Scripting:	37
PROJET INTENSIF DE 3 SEMAINES :	39
QUATRIEME PARTIE: Le changement d'état dans la simulation de fluides :	41
Les points clés du changement d'état	41

Du Liquide au Solide, approche nodale et scriptée:	41
Approche Nodale:.....	42
Approche Scriptée:.....	45
Du Solide au Liquide, Approche Nodale et Scriptée :.....	47
Approche nodale :	47
Du Solide au Liquide, Approche Scripté :	49
CONCLUSION :	55
GLOSSAIRE:	56
BIBLIOGRAPHIE:.....	57
Livres et Articles:	57
Vidéo et supports de cours multimédia :	57
Films et court métrages :	57
TABLE DES ILLUSTRATIONS.....	58
ANNEXES:	60
ANNEXE 1 :	60

INTRODUCTION :

Il est amusant de voir à quel point le futur peut réserver des surprises inattendues ?

Il y a moins d'un an, alors que je finis mon année de MASTER 1, rien ne laisse encore présager une orientation professionnelle et technique vers les effets spéciaux et encore moins les simulations de fluides.

A l'époque, je me destine à une carrière dans le rendu, shading, lighting, texturing... Mon stage de fin d'année me fait basculer dans la branche VFX d'une production numérique, bien que je n'ai que peu de connaissances théoriques et aucune pratique. La nécessité faisant loi, je développe peu à peu mon expertise dans ce domaine jusqu'à y prendre gout.

Tout naturellement, dans le but d'améliorer mes compétences dans ce domaine une fois mon stage terminé, je décide de choisir un mémoire ayant attiré aux effets spéciaux.

Mes objectifs au départ étaient très ambitieux, je souhaitais travailler l'esthétique VFX dans un large champ de la destruction à la simulation de liquides, le feu et la fumée, les générations procédurales etc.....

A peine ais je commencé à travailler sur les simulations de liquides qu'il m'est très vite apparu que cette section, sensée n'être qu'une partie de mon mémoire était à elle seule d'une complexité et d'une immensité suffisante pour faire l'objet de plus d'un mémoire de master.

J'ai procédé dans un second temps à un ciblage encore plus précis de mon sujet de recherche afin d'être en mesure d'aborder une problématique pouvant être résolue en une seule année.

En expérimentant ces différentes techniques sans but précis, en cherchant à prendre en main ces nouveaux outils et à me familiariser avec de nouvelles techniques, j'ai choisi la transmutation de la matière par l'utilisation de systèmes de particules.

En effet, Il y a quelque chose de fascinant à regarder fondre une bougie sous la chaleur vacillante de sa flamme, puis voir la cire se solidifier de nouveau, créant de nouvelles lignes et reliefs à sa chandelle.

Voir la matière en transmutation.

C'est ce changement d'état qui donc a finalement reçu toute mon attention, et j'espère que tout au long de ce mémoire, je saurais vous convaincre de l'intérêt de ce sujet.

PREMIERE PARTIE : ETAT DE L'ART ET SYSTEMES DE PARTICULES DANS LES EFFETS SPECIAUX:

LE CHANGEMENT D'ETAT DANS L'ART:

Avant d'aborder les outils et techniques employés durant cette année de recherche, je souhaite mettre en avant un certain nombre de réalisations Artistiques provenant de plusieurs types de supports créatifs.

L'intérêt principal n'est pas, bien qu'indispensable, la technique employée.

L'intérêt de ces œuvres est l'inspiration qu'elles m'apportent ainsi que les idées et principes techniques qu'elles véhiculent.

DEATH TO THE DEATH PENALTY:

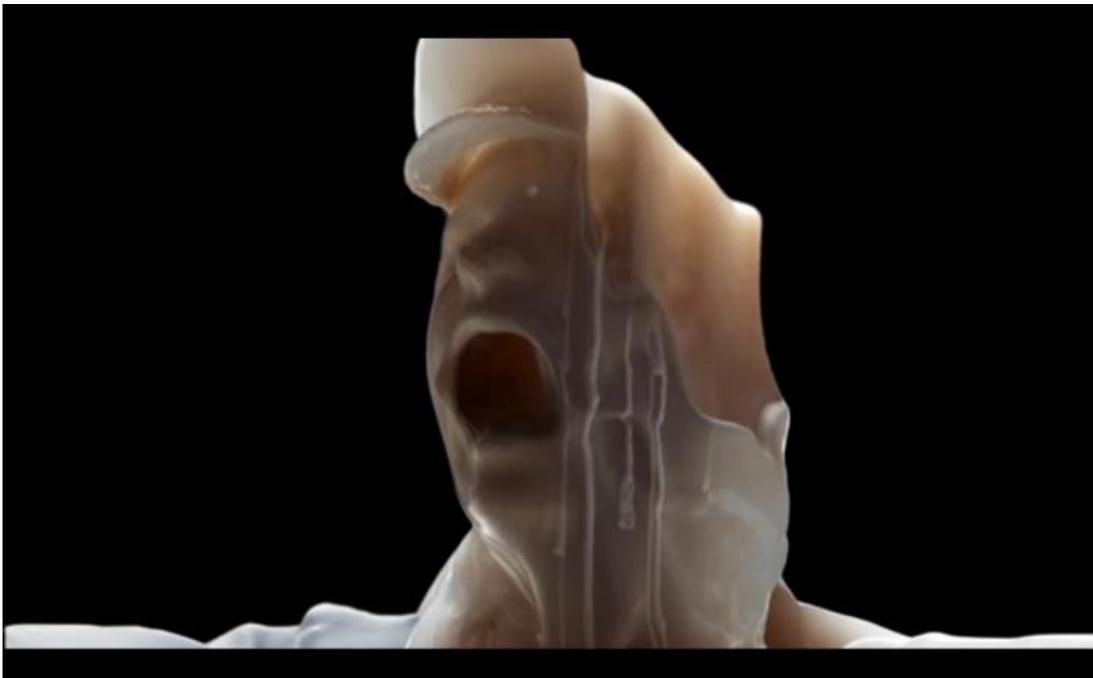


FIGURE1:AMNESTYCOMMERCIAL,PEIX&DIGITALDISTRICT

Cette courte publicité, que je peux volontiers qualifier d'œuvre d'art par la force du message qu'elle véhicule, peut à elle seule exprimer toute les raisons pour laquelle mon sujet de mémoire est celui que j'ai choisi.

Réalisé en 2010 pour Amnesty International, elle met en scène différents tableaux d'oppressions et de négation des droits de l'homme à travers le monde.

Que ce soit en Chine, dans ces pays autoproclamés républiques populaires, démocratiques et tout autre termes dont l'abondance d'adjectifs censés être positifs ne fait que renforcer leur dictature, que ce soit dans ces régions que la religion et la répression étouffent et avilient les Hommes et leur droits fondamentaux, ou les simples libertés, bref, tout ce que nous pensons acquis est bafoués de manière quotidienne.

Elle nous rappelle que la liberté des peuples et que le droit de l'être humain à vivre sans la peur et la souffrance n'est qu'à une bougie, une flamme d'espoir, de la liberté. Toutes ces scènes de torture, d'exécutions sommaires, ces idéologies ne peuvent que fondre face à la raison et les droits de l'homme.

Ce court-métrage a une force remarquable tant par sa mise en scène que sa technique.

Il ne m'a pas été donné de trouver quelques autres exemples de représentation du changement d'état, de transmutation de la matière dans le domaine des effets spéciaux qui soient aussi réussis tant sur le plan esthétique que technique.

La combinaison d'une réalisation au ralenti et en gros plan apporte d'immenses difficultés techniques que les logiciels d'effets spéciaux, même trois ans après la réalisation de cette vidéo, n'offrent toujours pas de solution clé en main face à ce défi technique.

Les plans serrés sur cette transmutation de la matière, de l'état solide à l'état liquide ne permettent pas de dissimuler sous des artifices le changement d'état. Le résultat de la simulation se doit donc d'être d'une qualité et d'un détail très élevé.

Il est envisageable qu'elle n'ait pas été faite en une seule simulation, et que plusieurs ont été combinées, la fonte d'un côté, les gouttes de l'autre. Je ne peux hélas que spéculer, la technique utilisée pour sa réalisation n'ayant, à ma connaissance, jamais été rendu publique.

C'est, en moins d'une minute, le meilleur exemple et la plus grande motivation et inspiration qu'il m'ait été permis de trouver pour m'inciter à m'investir dans cette problématique.



FIGURE 2 CREEP, 2011 PHOTO PAR ALISON FAIRLEY

CREEP, SKYE KELLY :

Creep, de l'anglais littéraire, est un synonyme qui signifie selon l'Oxford Dictionaries "(Of a thing) move very slowly and inexorably.[...] (Of a plastic solid) undergo gradual deformation under stress."

Cela signifie d'un objet qu'il bouge très lentement et inexorablement, et d'un solide en plastique ou autres matériaux proie à la déformation, qu'il est soumis graduellement à une déformation sous le stress », entendez par là une force extérieure.

Skye Kelly a créé une exposition composée de sculptures réalisées à partir de toffee, une confiserie composée de sucre ou mélasse et de beurre. Cette confiserie a tendance à fondre à basse température, ce qui, combiné à l'action de la gravité, crée une œuvre sans cesse changeante, soumise à une transformation de sa forme, une transmutation de sa

matière, d'un état apparemment solide à un état semblant liquide, sous l'influence de la gravité et du temps.

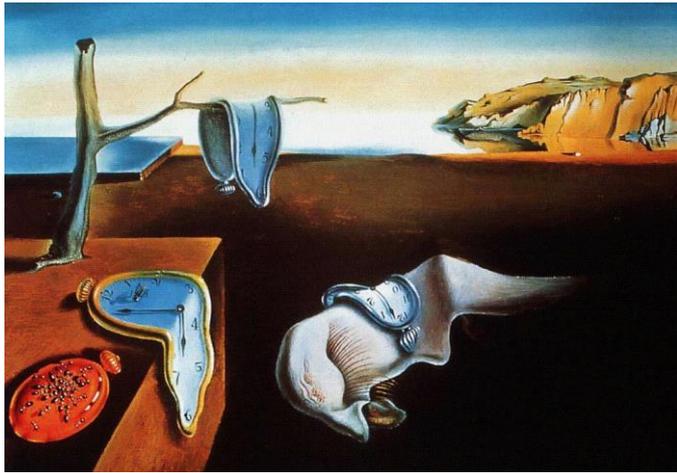


FIGURE 3 LA PERSISTANCE DE LA MEMOIRE, SALVADOR DALI, 1931
PEINTURE A L'HUILE

LA PERSISTANCE DE LA MEMOIRE, SALVADOR DALI:

Sans pour autant parler du sens profond qui a poussé Salvador Dalí à réaliser en 1931 cette œuvre surréaliste, on peut mettre en avant la dualité de la matière, dure et molle, présente dans ce tableau ainsi que l'action même de la transmutation de la matière prenant place sous nos yeux.

C'est un changement graduel, de l'état solide à l'état liquide, qui prend son temps. Il n'y a

pas une transition directe entre un état et un autre, mais un processus de transformation progressif, figé dans l'instant sous le pinceau du maître.

L'un des aspects de ce tableau intéressant d'un point de vue performance technique à relever dans le domaine des simulations pour les effets spéciaux est la tendance des montres à garder leur volume initial. Il nous est possible, malgré le changement d'état visible, de toujours identifier ces objets. Achever un changement d'état similaire dans une simulation nécessiterait de pouvoir faire changer l'état d'un objet non pas de manière binaire, solide ou non solide, mais de façon à avoir une transformation graduelle, un coefficient de fluidité propre à chaque partie de l'élément, et changeant sous la contrainte d'une force externe.

EXPANSION DE CESAR :

La série d'œuvres réalisées en polyuréthane par César à partir de 1967 attise ma curiosité, non pas sur le changement d'état en soit, nous ne sommes pas en train d'assister à une transformation de la matière, mais au résultat obtenu une fois cette transformation



FIGURE 4 EXPANSION AU POT A LAIT, CESAR, 1967

effectuée. Ces œuvres sont la représentation, le

souvenir, la trace laissée par la réaction source de transmutation. Elle m'évoque une coulée de lave s'étant refroidie, ou la cire d'une bougie se figeant sur la table. Elle met en avant la problématique de l'après :

Laisse-t-on à la matière la possibilité de revenir à son état d'origine, post métamorphose, ou bien cette altération est finale, et il n'y a aucune possibilité de revenir à son état premier.

NELE AZEVEDO, MELTING MEN :

En support à WWF, Nele Azevedo, artiste Brésilienne, a créée en 2009 à Berlin une installation temporaire nommée « Melting Men », Hommes fondant, ou les hommes qui fondent.



FIGURE5: MELTING MEN, BERLIN.
© ROSA MERK – WWF, 2009

L'installation était composée de 1000 petites sculptures en glace représentant 1000 silhouettes humaines assise. Il ne fallut que 30 minutes pour que disparaissent les sculptures.

Cette œuvre éphémère, comparé à l'œuvre de Skye Kelly, *Creep*, présente un intérêt supplémentaire, qui se situe dans le fugitif.

L'installation *Creep*, bien qu'évoluant en étant soumise aux mêmes forces, s'étend sur une période de temps assez longue. L'œuvre fait bien plus que fondre, elle se transforme, change son aspect au fil du temps, et pour longtemps.

Melting Men, de son côté, n'est que passager, périssable. Il n'est que dans l'altération, la mutation de sa forme. Il n'est pas remplacé par une autre forme tangible, il est là, puis il ne l'est plus. Il s'efface du monde, il est absorbé, dissous dans l'univers qui l'entoure, jusqu'à ne faire plus qu'un avec lui, et donc n'être plus une entité à part entière, disparaître, s'estomper de la surface du monde, tout simplement.

SPIDER-MAN 3 :



FIGURE 6 EXTRAIT DU FILM SPIDER-MAN 3, COLOMBIA PICTURES & MARVEL STUDIO

Je ne parlerais pas de tous les effets spéciaux de ce film, l'homme araignée se balançant de bâtiments en bâtiments, là n'est pas mon intérêt. Je souhaite vous parler d'un des personnages principaux, d'un des méchants.

Il s'agit de l'homme sable, ou *Sand-man* en anglais, à ne pas confondre avec le marchand de sable, qui est de bien meilleure compagnie, mais n'a absolument aucun intérêt pour ce mémoire.

Pour résumer sommairement ce personnage, l'homme sable, Aka William Baker, se retrouve de manière fortuite soumis à une expérience scientifique qui tourne mal. Il se retrouve alors avec la capacité de changer de forme en transformant une partie ou tout son corps en sable, ou en agglomérant le sable environnant pour augmenter sa masse et son volume.

C'est là le grand défi que représente l'adaptation de ce personnage de comics au cinéma : il s'agit d'animer et simuler une créature n'ayant pas de forme définie, changeant constamment d'aspect, d'état de sa propre matière.

Pour réussir cette prouesse technique, il a fallu que l'équipe en charge, dirigé par Doug Bloom, développent pendant six mois avant la production, des outils sur mesure.

Utilisant plugins, programmation sous python et C++, ils ont mis au point un workflow permettant de passer de rigid body à des simulations de fluides, leur permettant de créer un personnage rigide, animé, et de le faire changer, complètement ou partiellement d'état, passant d'un mesh à une simulation de millions de particules à une frame donnée, sans que le spectateur ne se rende compte de quoi que ce soit.

Bien que le sable ne soit pas une matière que j'envisage de simuler, l'attrait principal de ce personnage est qu'il ne s'agit pas d'un changement total et abrupt, le changement d'état est contrôlé et localisé, et entièrement réalisé sur ordinateur.

TERMINATOR 2 – JAMES CAMERON :



FIGURE 7: EXTRAIT DU FILM TERMINATOR 2 DE JAMES CAMERON (19)

Sans vouloir s'attarder trop longtemps sur ce film, mon travail de recherche serait incomplet sans aborder le second opus de la Saga Terminator, sorti en salle en 1991 et réalisé par James Cameron.

Dans le film, le T-1000 est un nouveau prototype de robot dont un des modèles est renvoyé dans le passé pour, une fois de plus, tenter de détruire la résistance humaine avant même qu'elle ne commence.

Ce robot a pour principal atout de pouvoir prendre la forme et imiter toute chose qu'il touche, changeant d'aspect, de forme et même d'état selon la situation, se rendant liquide pour passer sous une porte, ou transformant ses bras en pied de biche pour ouvrir un ascenseur, etc.

Stan Winston, Dennis Muren, Robert Skotak et Gene Warren ont remporté l'oscar des meilleurs effets spéciaux en 1991 pour leur travail sur ce film.

bien qu'utilisant des rendus générés par ordinateur en complément d'effets pratiques réalisés sur le tournage à l'aide de mannequins, prothèses et autres artifices Sa réalisation n'est pas basée sur des simulations de particules, mais sur la technique privilégiée à l'époque, le morphing, ou morphose. Il s'agit de l'animation de la transformation d'une forme en une autre forme prédéterminée, contrairement à une simulation qui commence à un point déterminé, mais on ne sait pas exactement où on arrive.

C'est cependant la première transmutation de la matière qu'il m'ait été donnée de voir étant jeune. Il mérite sa place dans mon état de l'art comme l'un des films ayant marqué une génération, restant présent dans mon esprit tout au long de cette année de recherche.

HISTORIQUE DES SYSTEMES DE PARTICULES DANS LES EFFETS SPECIAUX :

Pour chaque nouveau média que l'Homme invente, il cherchera toujours un moyen de l'améliorer, d'aller plus loin que ce qu'il est possible de faire avec cette forme d'art.

Pour s'approcher au plus près de la réalité avant l'invention de la photographie, l'Homme a inventé la chambre noire, ou *camera obscura*. Cette invention amena à la création de la photographie, qui permet alors de capturer le réel avec précision comme jamais il n'avait été possible de faire avant.

L'Homme ne s'est pas arrêté là, car pouvoir recopier le réel ne suffit pas, il lui faut pouvoir le transcender, le modifier pour y substituer sa propre vision du monde, et enfin créer sa réalité.

Depuis que l'homme a créé la photographie et le cinéma, il a cherché à montrer plus que ce que l'objectif capturait. L'histoire des effets spéciaux dans le cinéma est presque aussi vieille que l'histoire du cinéma lui-même. George Méliès est considéré comme le père des effets spéciaux dans le cinéma, ayant apporté à ce nouveau média des techniques de photographie ainsi que de nouvelles techniques révolutionnaires et fondamentale de l'industrie.

Bien trop de gens pensent que l'utilisation d'effets spéciaux est une nouvelle lubie, qu'il faudrait revenir à des films « à l'ancienne », n'utilisant pas de techniques d'altération de l'image.

A ces détracteurs, je souhaite rappeler que seulement 20 ans séparent la création du cinéma et la création des premiers effets spéciaux.

Les effets spéciaux numériques, cependant, sont eux assez récents dans notre histoire.

La première utilisation d'un ordinateur pour la création d'animation date de 1967 pour le court métrage de 10 minutes « hummingbird » de Charles Csuri et James Shaffer.

Dans les années 1980, en corrélation avec le développement des ordinateurs et la baisse de leur coût, le numérique s'invite dans le domaine cinématographique de manière intensive pour la première fois avec le film *Tron* de Steven Lisberger en 1982.

Bien que ce film n'est pas obtenu une de nomination pour l'oscar des meilleurs effets spéciaux de 1982 pour avoir « triché » en utilisant des ordinateurs, l'utilisation des ordinateurs pour générer des effets spéciaux est de nos jours devenu la norme pour toute productions, du petit écran aux salles de cinéma.

La première utilisation de systèmes de particules pour les effets spéciaux dans le cinéma peut être attribuée à Bill Reeves (1951-). Travaillant à l'époque pour Lucas films's Industrial Light and Magics, on lui attribue l'introduction du terme « Particle effect » et l'utilisation pour la première fois de cette technique pour la scène genesis du film « Star Trek II : Wrath of Khan » (1982). C'est la première utilisation de systèmes de particules pour des effets spéciaux, l'année même de la première utilisation intensive d'images numériques générées par ordinateur dans le domaine cinématographique.

Les systèmes de particules ont depuis lors été utilisés pour simuler tout éléments que l'on peut juger d'imprévisible : fumées, feux, liquides, cheveux, nuages et banc de poissons...

Les simulations de particules, qu'elles soient statistiques, physiques, voxel, sont un pilier de l'industrie des effets spéciaux, et ne cessent d'évoluer, de se complexifier tant par leur volume que leur propriétés techniques et algorithmes. Elles font partie d'un univers en constante évolution, auxquelles mes recherches, je l'espère, apporteront de nouvelles possibilités et utilisations pour l'avenir.

SECONDE PARTIE : LA SIMULATION DE PARTICULES DANS REALFLOW :

PRESENTATION DE NEXT LIMIT :

De nombreux logiciels permettent d'effectuer des simulations de fluides à petite ou grande échelle. Entre Houdini, Naiad et Realflow, Realflow s'est imposé pour moi comme le choix le plus stratégiquement viable tant pour la recherche effectuée durant cette année que pour un investissement professionnel à long terme.

Next Limit Technologies a été fondée en 1998 par Victor Gonzales et Ignacio Vargas.

En moins de quinze ans, cette société qui avait commencé avec deux personnes, deux chaises et deux ordinateurs est devenue une des références mondiale dans le milieu des effets spéciaux.

« Comble du bonheur », elle n'est pas située de l'autre côté du monde, mais à Madrid en Espagne.

Leur principale objectif est la création et l'optimisation de logiciels de visualisation et simulations techniques, que ce soit pour le grand public au travers des effets spéciaux numériques, ou à destination de scientifiques et d'ingénieurs.

Ils disposent de trois logiciels phares :

REALFLOW :

Realflow est un logiciel de simulation de fluide de petite échelle à grande échelle destiné au milieu des effets spéciaux. Utilisé en partenariat avec les plateformes 3D disponibles sur le marché, Realflow offre des solutions clés en main à grand nombre de défis techniques avec des résultats visuellement plaisant.

XFlow :

XFlow est un logiciel de future génération spécialisé dans la simulation de dynamique de fluides. A la différence d'autres logiciels de simulation de même nature, ce dernier est basé sur un système propriétaire basé sur les particules, mais sans aucune approche de maillage, permettant d'optimiser la simulation de problèmes complexes comme l'aérodynamisme. C'est en quelque sorte une version de Realflow destiné à être principalement utilisé par des ingénieurs.

MAXWELL RENDER :

Moteur de rendu développé par Next Limit, il est principalement utilisé dans l'architecture, la production de longs et courts métrages ainsi que le design de produits et le packaging. Il est utilisable directement depuis Realflow.

INTRODUCTION A REALFLOW :

Avant de commencer toute explication sur les différents tests et recherches effectués au cours de cette année de Master 2, je souhaite vous fournir une compréhension globale des rouages de Realflow. Je vais vous initier à des fonctionnements et des paramètres fondamentaux qui vous permettront de mieux apprécier le fruit de mes recherches.

Cette courte introduction technique du logiciel ne s'attardera que sur la simulation basée sur les particules. Nous ne couvrirons pas la simulation à base de voxels ni les simulations dynamiques d'objets, dont la connaissance n'est pas nécessaire pour l'assimilation des nombreuses explications techniques de ce mémoire.

Les particules standard peuvent être utilisées pour une grande variété de projets, afin de représenter nombres de liquides, que ce soit de l'eau, du miel, pétrole, sang et j'en passe. Il est principalement utilisé pour des simulations de petites à moyennes taille, ayant un niveau de résolution permettant de voir chaque filament, goutte, éclaboussure de votre liquide. Il est utilisé en conjonction de forces externes nommées daemon, mais offre en lui-même de nombreux réglages permettant de créer un fluide ayant les propriétés physiques se rapprochant le plus de ce que l'utilisateur souhaite représenter.

Dans cette partie, je vais passer les principaux paramètres en revue afin de vous donner une connaissance de base des outils à votre disposition.

TYPES DE PARTICULES ET PARAMETRES INTERNES:

Il existe plusieurs types de particules, dont je vais faire une description exhaustive :

Gas : Gazeux, utilisé pour simuler des gaz, de l'air. Les particules de type gaz ont une très forte dilatation et vélocité. Elles vont remplir un espace et s'étendre dans toutes les directions.

Liquid : Liquide, c'est le type de particules de base et celui utilisé tout au long de ce mémoire. Elles permettent de simuler des fluides à viscosité très élevée et sont les mieux adaptées pour des interactions avec des objets.

Dumb : le nom signifiant « stupide », ces particules n'interagissent pas entre elles, elles sont donc très rapides à calculer et sont principalement utilisées pour des effets secondaires, que ce soit de la mousse, de l'écumes et autre.

Elastic : Elastiques, ces particules ne se comportent aucunement comme les autres types de particules, elles se rapprochent des simulations d'objets dynamiques, et sont utilisées pour réaliser des matériaux gélatineux. Elles ne peuvent être utilisées avec d'autres type de particules.

Script : Scriptées avec python ou en C++, ce type de particules vous permet de créer le fluide que vous souhaitez. Aucun script de base ne vous est fourni, vous devez tout réaliser de A à Z.

Les paramètres suivants ont été testés à l'aide des particules de type « Liquid », tout test présent dans ce mémoire utilise, sauf explicitement indiqué, ce genre de particules.

LA DENSITE :

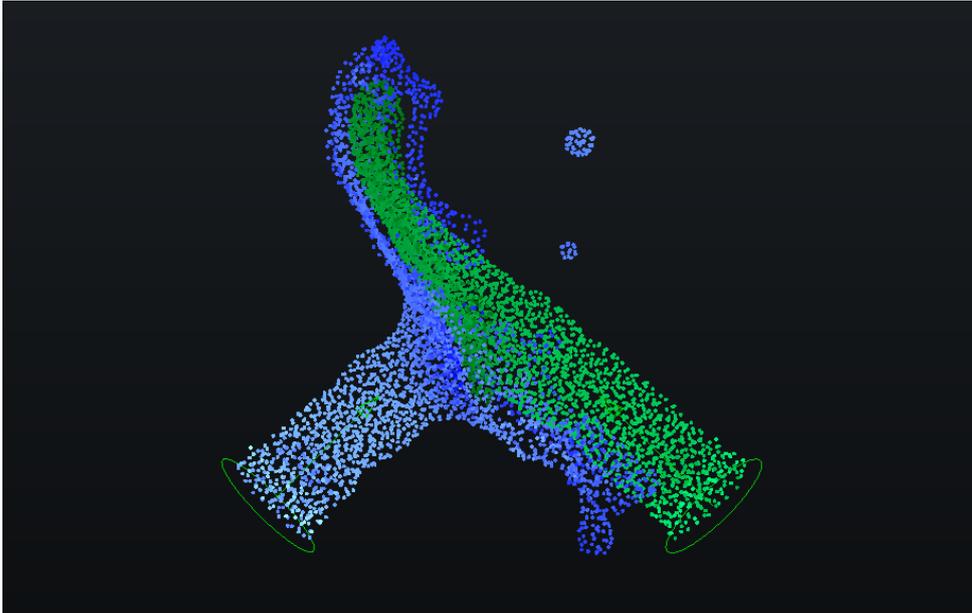


FIGURE 8: REALISATION PERSONNELLE, COMPARAISON DE DENSITE, 10 ET 40

La densité dans Realflow est exprimé en kilogramme par mètre cube. Chaque liquide à une densité bien propre, l'eau est de 999.9720 kg/m^3 , le mercure 13534 kg/m^3 , soit suffisamment pour qu'un boulet de canon flotte à sa surface. C'est un paramètre important afin d'accéder à une simulation et un comportement réaliste, surtout si votre simulation est composée de plus qu'un seul liquide.

LA VISCOSITE :

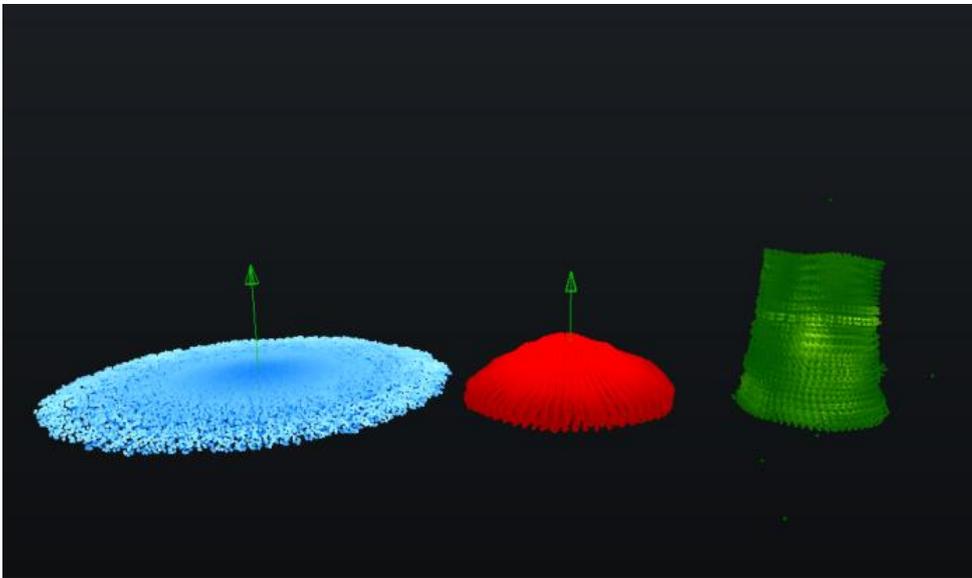


FIGURE 9: REALISATION PERSONNELLE, COMPARAISON DE VISCOSITE, 3, 10 ET 50

Tout fluide tend à garder sa cohésion, ils ont tous un certain montant de viscosité, même l'eau. Les fluides ayant une grande viscosité vont former des filaments, des fils lorsqu'ils sont séparés ou étendus. Augmenter la viscosité permet de simuler des fluides au comportement similaire au miel, au goudron et autre, tandis que les fluides tel que l'alcool ont une très basse viscosité. Ce paramètre devient instable lorsque sa valeur est très élevée, provoquant une explosion de particules à très haute vitesse. Il est possible de contre balancer ce problème en augmentant la résolution du fluide et le nombre de substeps, résultant en une augmentation du temps de calcul.

LA TENSION DE SURFACE :

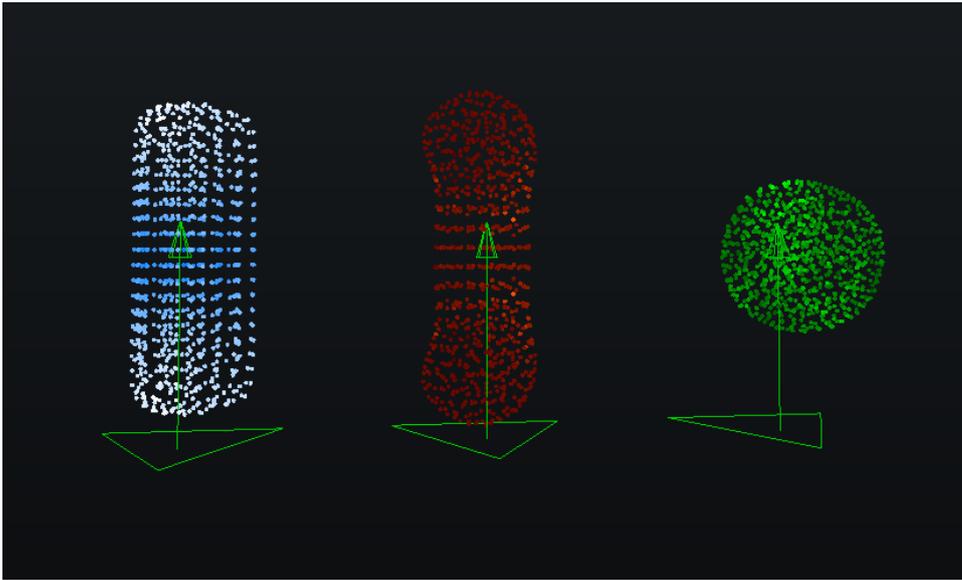


FIGURE 10R RÉALISATION PERSONNELLE, COMPARAISON DE TENSION DE SURFACE, 0, 10, 100

A la surface de tout fluide est visible une sorte de très fine membrane, due aux molécules externes du fluides qui sont attirés les unes au autres par des forces. C'est la tension de surface qui donne à la goutte d'eau sa forme arrondie. Ce paramètre est alors utilisé afin que le liquide simulé ne s'étale pas trop et ne devienne trop fin, il apporte une certaine cohésion au liquide.

INTERACTION DES PARTICULES ET DE L'ENVIRONNEMENT, LES DAEMONS :

GRAVITY DAEMON :

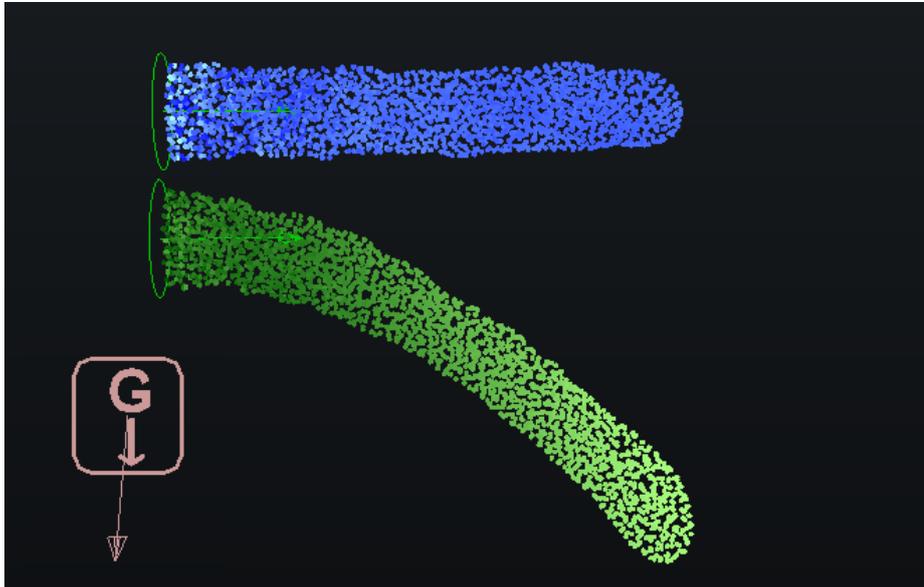


FIGURE 1: REALISATION PERSONNELLE, GRAVITY DAEMON.

Probablement le Daemon le plus utilisé, il permet de simuler la gravité sur votre scène. Ce daemon ne prend pas en compte la masse/densité des objets, quelle que soit leur poids, ils tomberont à la même vitesse (comme dans un environnement sous vide).

Il faut donc adapter ce daemon suivant l'objet ou le système de particules qu'il influence.. Il est possible de compartimenter ce daemon sur une zone délimitée, ainsi que de l'orienter comme bon vous semble. Il applique de base une force de 9.8 suivant un axe donné, à vous de choisir quelle axe vous désirez.

ATTRACTOR DAEMON:

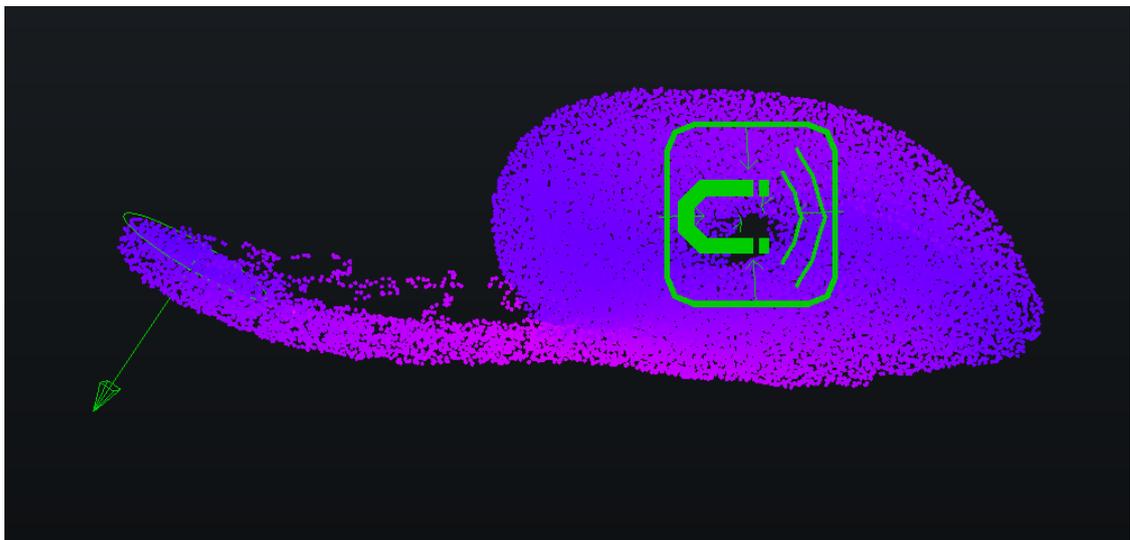


FIGURE 2: REALISATION PERSONNELLE, ATTRACTOR DAEMON.

Egalement l'un des plus utilisés, l'attractor daemon permet de sculpter et diriger un fluide, de le faire danser en attirant et repoussant les particules. Ce daemon permet d'attirer vers son centre des particules, ces dernières étant affectées par la force d'attraction de manière exponentielle à mesure qu'elle se rapproche du centre. Ce Daemon peut être également utilisé avec des valeurs négatives, créant une force qui repousse les particules au lieu de les attirer.

NOISE FIELD DAEMON :

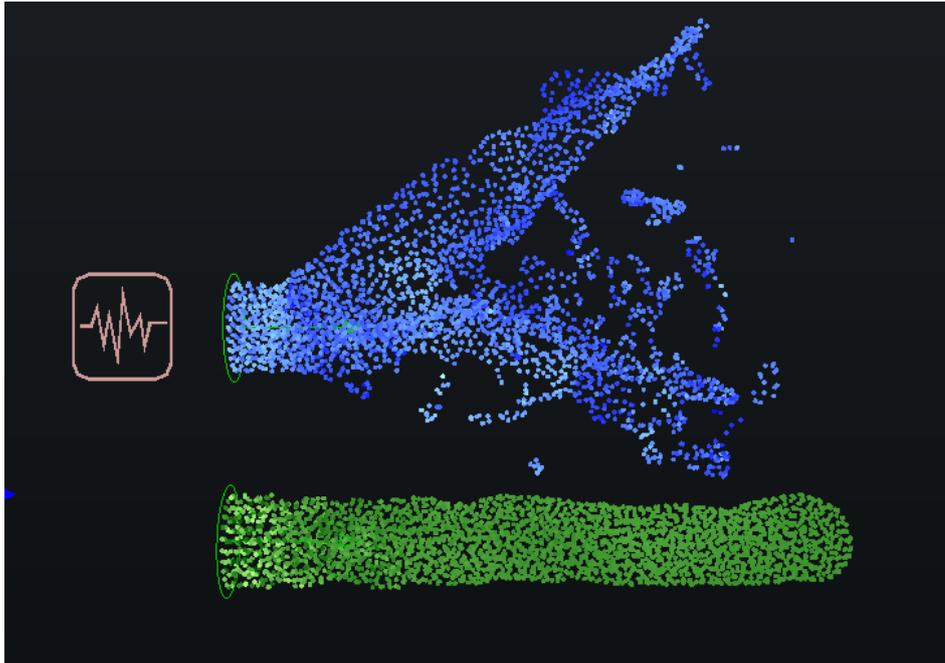


FIGURE 13 RÉALISATION PERSONNELLE, NOISE DAEMON.

Ce daemon est utilisé pour perturber aléatoirement les particules ou objets auquel il est rattaché. Ajouter ce genre de perturbation permet d'avoir un poil plus de réalisme en ayant plus d'aléa dans la simulation. Il peut affecter la scène entière, ou bien être limité dans l'espace. Une utilisation classique de ce daemon est de l'attacher à un émetteur afin de rajouter des turbulences à l'émission, évitant ainsi d'avoir un flux de particules trop régulier, et ajoutant par la même du volume à votre simulation.

MAGIC DAEMON :

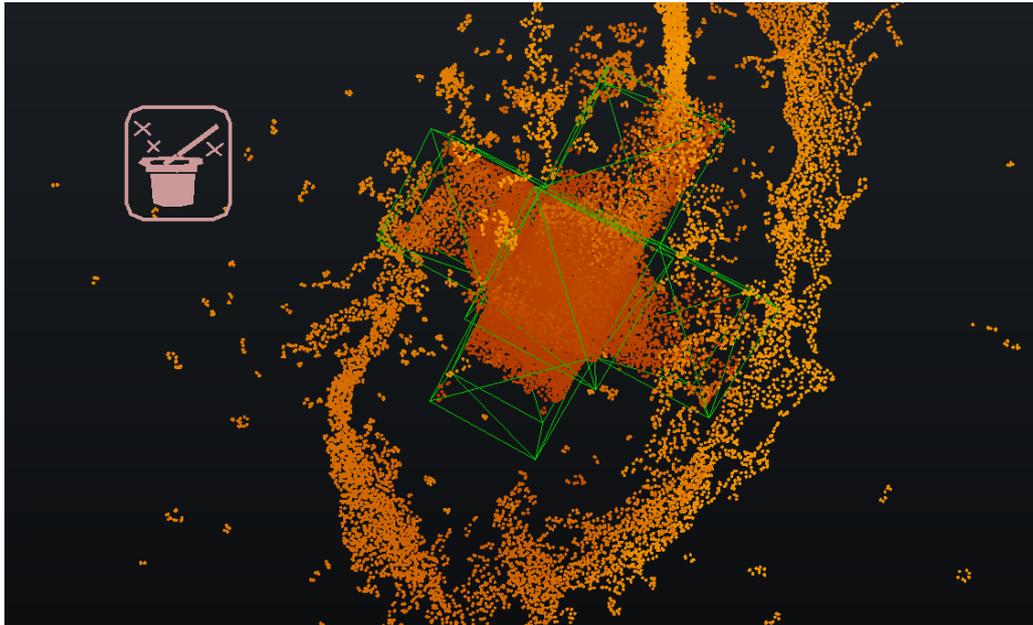


FIGURE 14 REALISATION PERSONNELLE, MAGIC DAEMON.

Un daemon très utilisé dans le milieu de la publicité, le Magic daemon vous permet d'attirer les particules d'un ou de plusieurs émetteurs afin de les faire prendre la forme d'une géométrie choisie. Il permet en quelque sorte d'effectuer une morphose de votre fluide.

Combiné à plusieurs autres daemons ou à un peu de programmation python, le potentiel artistique et technique est très intéressant.

FILTER DAEMON :

Ce daemon a été de grande utilité au cours de mes différents tests, il permet tout simplement de transférer des particules d'un émetteur à un autre, suivant une grande variété de conditions et paramètres.

L'utilisation est très simple : Il vous suffit de choisir un émetteur source et un émetteur Cible, puis choisir une condition et une valeur.

Vous pouvez par exemple choisir de transférer les particules de l'émetteur source à l'émetteur cible si ces dernières ont une vitesse de moins de telle valeur.

L'ayant de nombreuses fois utilisé dans mes tests, Il offre une excellente base pour l'expérimenter sur le transfert de particules. Cependant, l'usage de la programmation reste nécessaire pour les transferts les plus complexes.

LA GEOMETRIE :

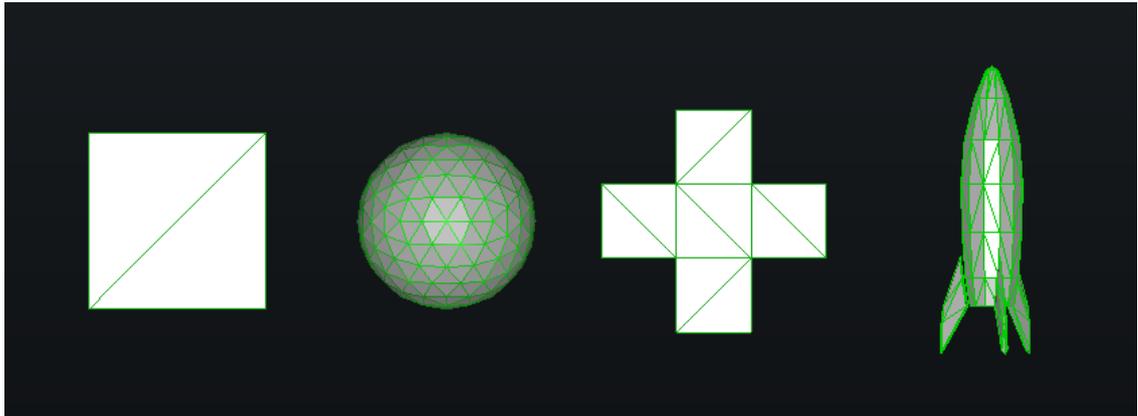


FIGURE 6 REALISATION PERSONNELLE, COMPARATIF GEOMETRIE.

Realfow dispose de plusieurs géométries de base pouvant être utilisées avec les simulations de particules et / ou en tant qu'objets dynamiques. Il est possible d'importer ses propres géométries, à conditions que ces dernières soient triangulées. Un module Realfow peut être intégré à Maya afin d'automatiser l'importation, et permet également d'importer des objets animés.

Les géométries importées dans Realfow peuvent être exportées une fois la simulation finie. Realfow gérant les simulations de rigidBodies et de SoftBodies, il est possible de les récupérer pour les utiliser dans d'autres logiciels de rendu.

LE RELATIONSHIP EDITOR :

Ayant une approche nodale depuis la version 2013 du logiciel, le Relationship editor vous permet de gérer les interactions entre les éléments que compose votre scène. Cela inclut les émetteurs, les daemons et les géométries.

De base, lorsque vous créez de nouveaux éléments, ils sont tous reliés les uns aux autres à travers le « hub », ou pivot central.

Dans cette configuration, tous les éléments de votre scène interagissent entre eux, ils affectent les autres et sont affectés mutuellement.

L'intérêt du Relationship editor est de pouvoir créer des liens spéciaux entre différents éléments. On peut souhaiter que tel fluide soit soumis à la gravité, mais pas telle géométrie. On peut également souhaiter qu'un élément puisse affecter un autre, mais que cette interaction ne se fasse que dans un sens.

Le Relationship editor permet de hiérarchiser sa scène et d'organiser les interactions entre-les éléments la composant afin de parvenir à des simulations plus riches et plus complexes.

LA PROGRAMMATION DANS REALFLOW :

Realfow offre la possibilité de facilement apporter ses propres modifications et scripts à sa scène pour pouvoir l'enrichir et produire des résultats uniques.

Que ce soit en C++ ou sous pythons, de nombreux outils sont disponibles, que ce soit en support d'un effet existant, vous permettant de le paramétrer plus en détail que ce qu'il vous est proposé de base, ou à partir de rien, afin de créer un nouveau bouton, une nouvelle fonction s'exécutant avant, pendant ou après votre simulation, la programmation sous Realfow est intuitive, rapide à rendre en main et dotée de nombreuses aides et documentations.

L'IMPORT, EXPORT ET LA COMPATIBILITE

Realfow dispose de plugins permettant d'aider son intégration à votre workflow.

N'ayant pas essayé de plugins sur un autre logiciel que Autodesk Maya, je me base sur ce dernier pour vous exposer ses principaux atouts.

Realfow ne supporte que les maillages triangulaires, ce qui est à éviter au plus possible en modélisation. Devoir convertir chaque objet manuellement est assez ennuyeux, et pour ceux que vous ayez une animation, avec ou sans déformation de maillage, vous allez très vite vous énerver...

Heureusement, le plugin vous offre la possibilité d'exporter d'un simple clic votre la scène et ses animations. Il est également faisable d'importer des simulations de particules de maya et les maillages en résultant. Le Plugin permet bien évidemment de faire de même dans l'autre sens, afin de réimporter dans votre la scène maya les maillages générés ou bien les particules directement.

Concernant les maillages, ou mesh, il est important lors de leur création d'optimiser au mieux leurs densités par rapport à leur taille à l'écran. Il est très aisé de se retrouver avec un maillage produit à partir d'une simulation de plusieurs millions de polygones.

Une astuce que j'ai trouvée, lorsque vous faites vos rendus avec Vray sous Maya, est de convertir le maillage importé depuis Realfow en proxy Vray. Cela vous permet d'avoir un maillage optimisé pour ce moteur de rendu, plus léger à l'affichage, permettant de récupérer des informations comme une passe de vélocité qu'il n'est pas possible d'avoir en faisant votre rendu directement sur le maillage produit par Realfow.

TROISIEME PARTIE : MAITRISER LE LOGICIEL, TESTS DIVERS ET VARIES :

Realflow est un logiciel complexe, permettant de nombreuses simulations utilisant multiples techniques et procédés. Avant de me spécialiser dans le changement d'état via le transfert de particules, j'ai été amené à expérimenter les divers outils mis à ma disposition afin d'acquérir les bases nécessaires à ma recherche.

L'artistique au service du technique :

Avant de commencer à travailler dans les effets-spéciaux, mon principal intérêt dans la 3D était le rendu. Si je n'avais pas fait de simulations durant mon stage à Ghost Vfx, mon sujet de mémoire aurait certainement concerné les textures, le shading et le lighting.

Je ne souhaitais pas mettre de côté ces compétences durant mon mémoire de m2, et c'est pour cela que chaque test réalisé au sein de Realflow a ensuite été exporté sous Maya, texturé et éclairé à l'aide de Vray, puis composité sous Nuke et/ou After Effect.

Il est important pour moi de montrer le potentiel artistique de mes recherches, qui bien que technique dans le fond et dans la forme, ont été réalisées afin de fournir aux artistes un outil pour leur permettre de transmettre de nouvelles émotions et sensations au spectateur.

C'est la raison pour laquelle, en fin de certaines analyses de mes tests, un court paragraphe développe l'aspect artistique pour mettre en valeur et partager de manière plus agréable le fruit de mes recherches.

Les vidéos de ces tests sont disponibles à l'adresse suivante : www.vimeo.com/rossetto

HYBRIDO 2.0:

RESEARCH PAPER TEST 02 : WATERBALL 01 :

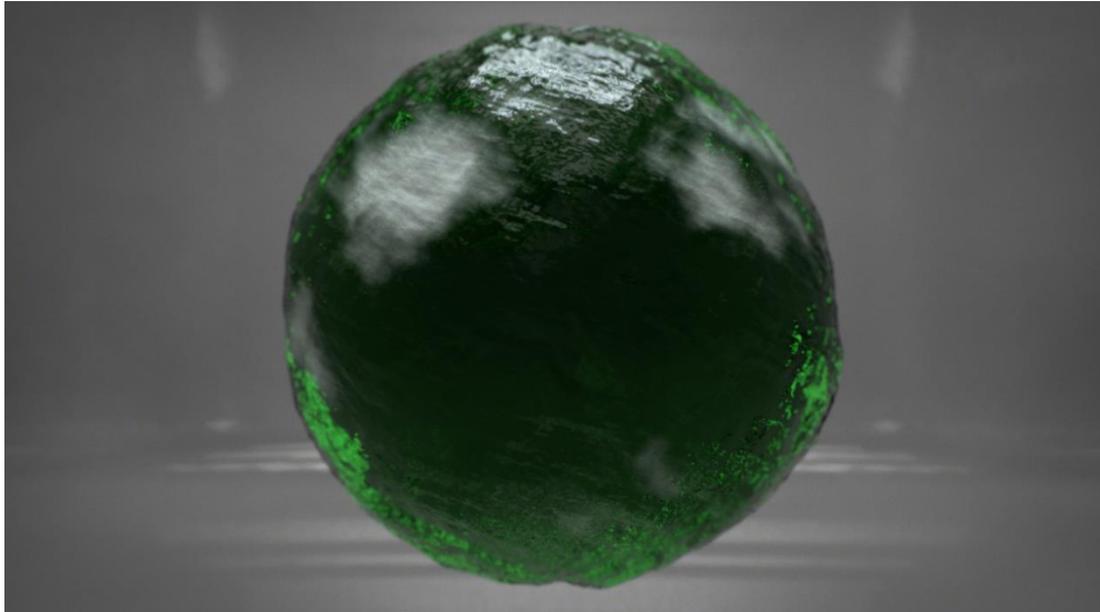


FIGURE 16 REALISATION PERSONNELLE, TEST 02 WATERBALL 01

L'outil Hybrido 2.0 est parfait pour la simulation de grandes étendues d'eau ou le nombre de détails est énorme. Il permet également la création de *displacement map* pour créer par exemple la mer autour d'un bateau, seul l'eau directement autour du navire étant réellement dynamique.

Ayant toujours eu un faible pour les paradoxes, simuler une simple étendue d'eau ne m'intéressait pas vraiment. Ce genre de test peut être trouvé fréquemment sur internet, et je souhaitais, bien qu'en utilisant la même technique, faire quelque chose de différent. Ne pouvant simuler qu'une *displacement map* pour une surface plane dans Realflow, j'ai alors récupéré les fichiers créés, puis passant sous maya, j'ai déplié au mieux les UV d'une sphère afin de pouvoir y appliquer les textures animées.

Cela présente de petits défauts au niveau des pôles de la sphère, les textures étant quelque peu pincées, mais grâce au cadrage de la camera, à la puissance du moteur de rendu Vray et au compositing dans Nuke, le résultat final n'est pas déplaisant. Ce test m'a permis de me remettre à utiliser ces outils, c'est une sorte de mise en bouche avant d'attaquer des tests plus complexes.

Intension Artistique :

Il y a quelque chose de puissant qui se dégage de cette masse d'eau en suspens, défiant l'une des règles les plus fondamentales de notre réalité, la gravité. Seul objet présent dans cette pièce grise et froide, elle est comme une planète, un élément ne répondant qu'à ses propres règles. Sa surface ne bougeant peut, on ne peut qu'imaginer ce qui se passe dans les tréfonds de son être. La lumière se perd dans sa profondeur. Bien que limitée dans l'espace, elle n'a pas de limite en elle-même.

RESEARCH PAPER TEST 03 : HYBRIDO 01 :



FIGURE 17. REALISATION PERSONNELLE, TEST 03, HYBRIDO 01

Ce test avait pour but de se familiariser avec l'outil Hybrido 2.0. L'intérêt principal était d'avoir une connaissance globale de ses capacités afin d'évaluer son utilité pour ce projet de recherche.

Contrairement aux simulations basées sur des particules uniquement, ce simulateur utilise un grillage, tout comme le simulateur Maya Fluid utilisé principalement pour créer des explosions, de feu ou de la fumée (ou les trois en même temps !).

L'approche particules /grille est très rapide, ne nécessitant que peu de steps par frames. Il est très facile de réaliser des simulations de 20, 50 millions ou plus de particules pour un temps de rendu raisonnable. C'est la raison pour laquelle il est principalement utilisé pour des simulations de moyenne à grande ampleur, tel qu'un raz de marée, un bateau dans la tempête, etc...

Ce test n'est qu'un simple conteneur se remplissant d'un liquide. Bien que simple de par sa réalisation et son attrait esthétique, il m'a permis de mettre en avant des problèmes de maillage automatique.

En effet, on peut, lors du visionnage de la vidéo, remarquer de nombreuses taches noires qui apparaissent et disparaissent brusquement. Ce soucis est dû à un maillage pas assez détaillé qui présente des interpénétrations. L'identification de ce problème m'a permis d'être plus attentif sur les tests suivants, et m'a fait gagner en qualité ainsi qu'en productivité.

AUTRES EXPERIMENTATIONS:

RESEARCH PAPER TEST 04: BUBBLE 01:

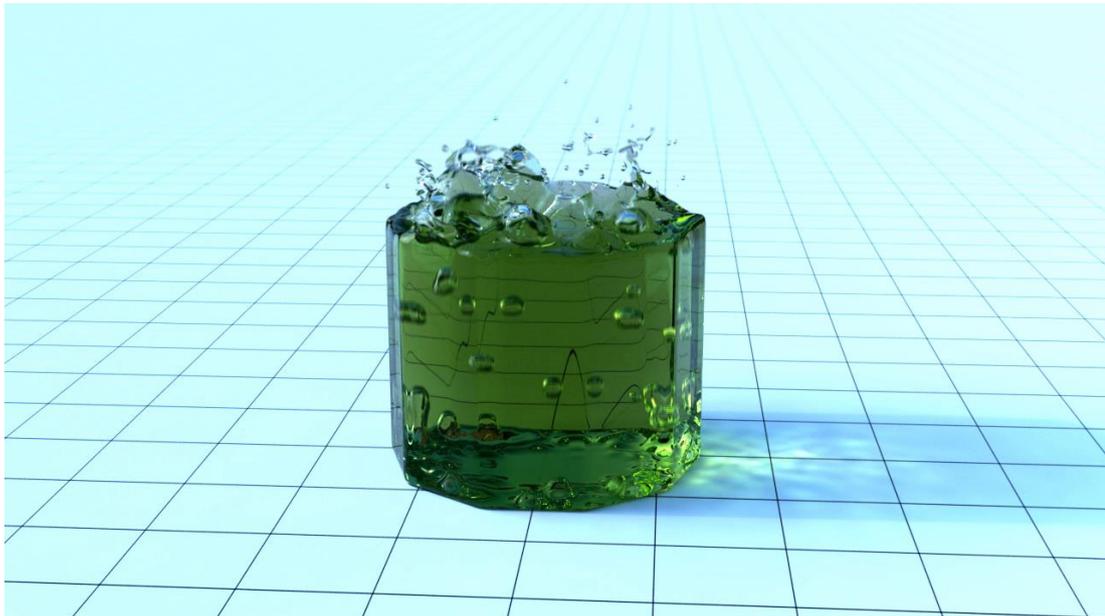


FIGURE 18 REALISATION PERSONNELLE, TEST 04, BUBBLE 01

Durant mon stage chez Ghost Vfx à Copenhague, ma principale mission était la réalisation d'effets spéciaux pour le film Norvégien *Gaten Ragnarok*.

En arrivant dans cette entreprise, j'avais pour but de travailler sur l'éclairage, les textures et le rendu des scènes 3D.

J'ai été quelque peu pris de court lorsque la mission que l'on m'a attribuée a consisté à créer des effets spéciaux, quelques simulations de rigid body, mais principalement du maya fluid . Le but était de générer de la poussière provenant de destruction ainsi que la respiration du serpent mythique présent dans ce film, évoluant dans des endroits sombres, froids et humides.

Une des rares scènes complètement générée par ordinateur consistait en une vue contre plongée sous l'eau. A la surface, on voit la silhouette d'un radeau de fortune. Pour renforcer l'idée du monstre guettant sa proie du tréfond de ce fjord Norvégien, j'ai été chargé de réaliser des bulles afin de représenter sa respiration sous l'eau.

A cette époque, je ne connaissais Realflow que de nom, et n'avait jamais été amené à l'utiliser. Durant ma phase de recherche pour réaliser cet effet, j'ai découvert le travail de Yuval Kolton, aka PixelPro, réalisant de nombreux tests surprenant sous Realflow, dont des bulles dans un verre.

Cependant, faute de pouvoir apprendre à maîtriser ce logiciel dans le délai imparti, je fus contraint d'utiliser les fluides sous maya dont j'avais l'habitude. C'est à ce moment-là que j'ai éprouvé l'envie de découvrir et d'apprendre à utiliser Realflow.

Il était donc important pour moi de réussir à reproduire la vidéo de Yuval Kolton, car c'est elle qui a éveillé le désir de maîtriser cet outil. Si j'arrive à reproduire cette vidéo d'un point de vue technique me suis dis-je, je parviendrai à utiliser ce logiciel dans toute sa complexité.

Une bulle d'air dans un liquide n'a pas une forme ronde parfaite, elle est soumise à une déformation due à la présence d'un liquide offrant une résistance à son passage. Afin de simuler des bulles dans un liquide, il est nécessaire d'avoir deux émetteurs de

particules ayant des densités différentes et soumises à des forces externes n'étant pas les mêmes.

Pour réaliser ce test, le système de particule représentant l'air est émis sous celui représentant le liquide avec une force externe attirant ses particules vers le haut, tandis que les particules représentant le liquide ont une force externe les attirant vers le bas et sont contenues dans un maillage en forme de verre. Les « particules d'air » quant à elles n'entrent pas en collision avec ce maillage. Leur densité étant plus élevée que celle des « particules liquides », elles forcent leur passage vers le haut avant d'être détruites une fois passées au travers de l'autre système de particules.

Concernant la mise en scène, je n'ai pas souhaité mettre le conteneur autour de ma simulation de fluide. Je souhaitais que l'effet produit n'ait pas besoin de réceptacle pour le justifier dans le plan. Il est le résultat de mon travail et n'a pas à être encloisonné, mais mis en avant, présenté pour ce qu'il est.

RESEARCH PAPER TEST 05 : SKULL 01:

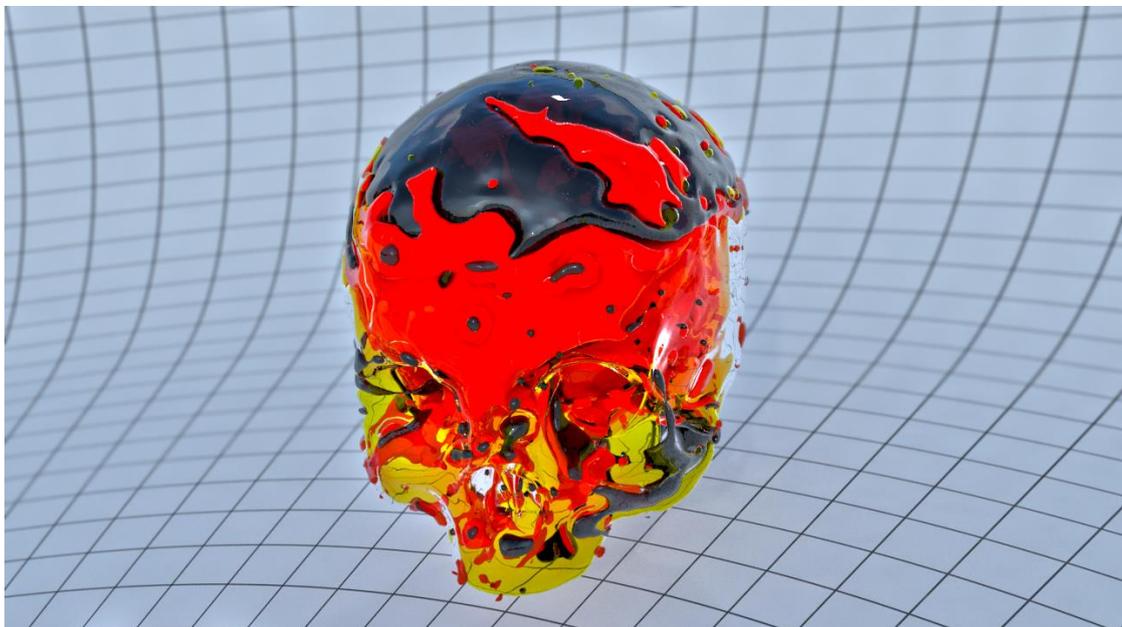


FIGURE 01 REALISATION PERSONNELLE, TEST 05 SKULL 01

Le but principal de ce test est de me familiariser avec les simulations à multiples émetteurs. Avoir plusieurs réseaux de particules au sein d'une seule scène, tous limités dans un espace confiné, m'a permis d'expérimenter avec différentes densités de fluides ainsi que différents daemons.

Après avoir importé dans ma scène Realflow le maillage représentant le crâne, j'ai créé trois émetteurs en forme de sphère, ayant chacun une densité propre, plus ou moins de tension de surfaces et plus ou moins de viscosité.

Ces trois émetteurs confinés dans cet espace restreint ont alors été soumis à différents daemons, les principaux étant un daemon représentant la gravité, et un créant un vortex, forçant les particules à tourbillonner dans ce crâne, et a donc se mélanger.

L'un des principaux problèmes rencontré lors de cette simulation est ce que l'on peut appeler des « Rogue Particles », des particules qui font ce que bon leur semble, créant un effet shrapnel, saturant l'espace de particules partant dans tous les sens

avec une vitesse aberrantes, provoquant de nombreux problèmes dans la simulation. Il est possible de voir dans la vidéo des points noirs apparaissant et disparaissant en dehors du crâne.

Ces particules sauvages, si je puis me permettre l'expression, sont dues à la combinaison de plusieurs facteurs, tels qu'une densité de fluide trop élevée, des daemons leur appliquant une force trop forte et d'un nombre de substeps insuffisants, réduisant la précision et la qualité de la simulation.

Les particules arrivant contre le crâne à une vitesse trop élevée et une force trop importante passent alors au travers du crâne, la simulation n'ayant pas assez de temps pour les prendre en compte.

RESEARCH PAPER TEST: TEXTURE EMISSION:

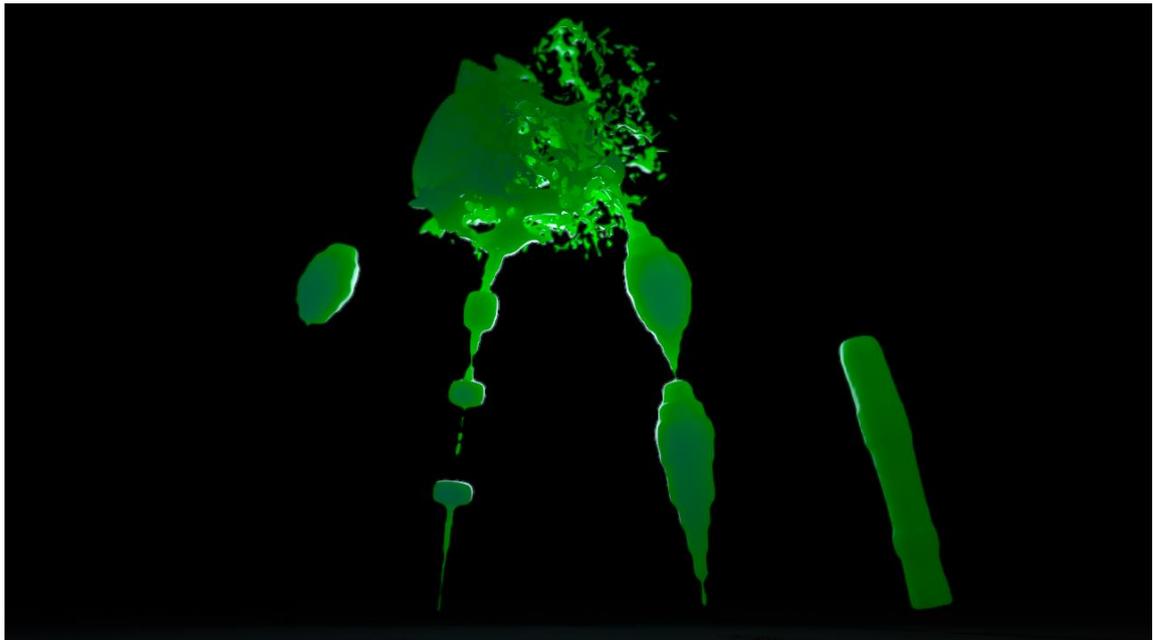


FIGURE 20 REALISATION PERSONNELLE, TEXTURE EMISSION.

En préparation au projet intensif de janvier, j'ai voulu expérimenter les relations possibles entre particules et son. Notre projet étant axé autour de la musique, je voulais pouvoir apporter quelque chose de dynamique, en cohésion avec le rythme sonore.

Il n'existe aucune technique à ce jour permettant d'importer dans Realflow un son et de l'utiliser pour contrôler un paramètre, tel que l'émission de particules.

Ce qui est possible cependant, c'est d'importer des textures animées, et d'émettre des particules à partir de ces dernières. Ces textures ne peuvent qu'être noires et blanches, indiquant à Realflow ou émettre et ou ne pas émettre.

J'ai alors utilisé le logiciel After-Effect avec lequel il est possible de créer des clés d'animations à partir de l'intensité d'un son importé. Ayant à ma disposition les pistes séparées d'un morceau de musique test réalisé par Eddy Camilleri, j'ai pu animer le diamètre d'un cercle blanc sur fond noir. Quand l'intensité de la musique est à 0, la texture est toute noire, et le cercle blanc s'agrandit suivant le rythme et la puissance de l'instrument. Vous pouvez d'ailleurs distinguer dans l'image ci-dessus les formes répétitives que prend le fluide. Chaque jet à sa trace précise, définie par le rythme et la mélodie de l'instrument sur lequel il est basé.

J'ai donc créé une texture animée pour chaque instrument, puis je les ai importées dans Realflo et utilisé 4 émetteurs contrôlés par les 4 textures animées pour arriver à ce résultat.

Le rendu de ce projet a ensuite été fait sous Vray pour Maya, en utilisant un shader translucide.

Travailler sur un son m'a fait réfléchir à la musique, et les aux ambiances dans lesquelles on écoute cette musique, l'éclairage, la décoration... J'ai pensé à la lampe à lave et combien il est aisément facile de regarder sa cire monter et descendre, la lampe du dessous dessinant les nuances de couleur à la surface de chaque bulle. J'ai voulu donc reproduire cette texture, mais également ne pas confiner la cire avec un éclairage trop violent. Je voulais donner au fluide l'énergie que véhicule la musique, le mettre sous les projecteurs.

RESEARCH PAPER TEST 08: WETMAP:

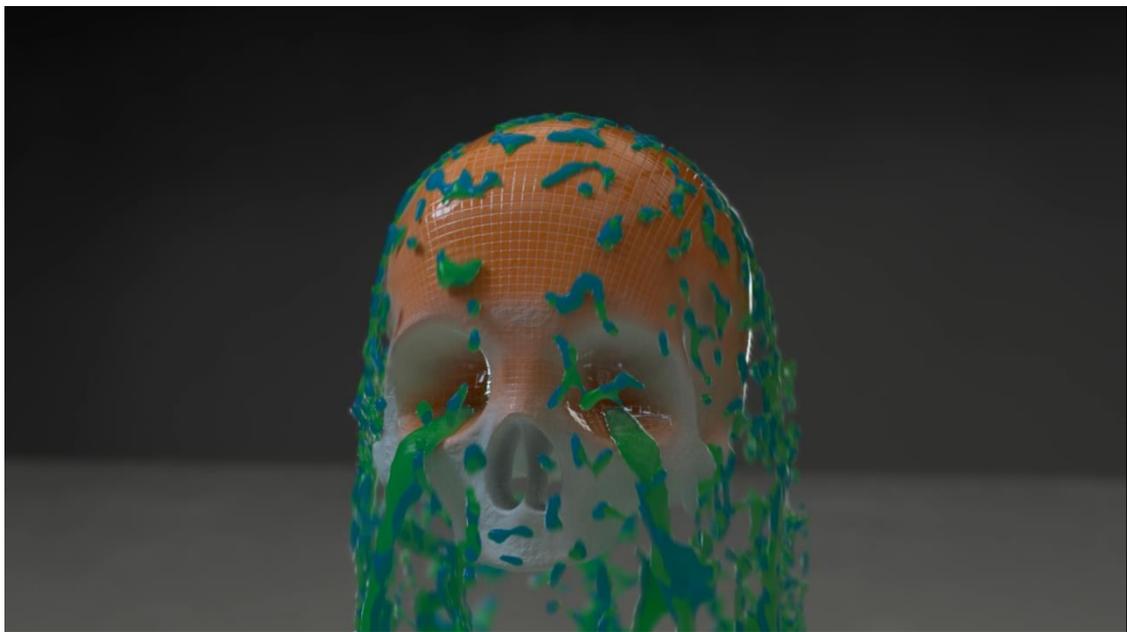


FIGURE 2: REALISATION PERSONNELLE, TEST 08 WETMAP.

Le rendu est un aspect important à mes yeux, d'un test de simulation ou d'un test technique en général.

Il est toujours plus plaisant de voir un progrès technique mis en scène un tant soit peu, ne serait-ce que par respect pour le temps de travail passé sur cette simulation. Si je suis prêt à m'investir intensément pour une simulation de moins de 10 secondes, autant qu'elle soit jolie et potentiellement utilisable pour mon showreel.

C'est la raison pour laquelle j'ai choisi d'expérimenter sur des éléments de Realflo n'ayant pas un attrait technique affectant directement la simulation, mais ayant comme rôle la création d'informations tel que des textures pouvant être utilisées lors de la phase de rendu.

La WetMap est l'un de ces éléments, on peut la traduire en « carte mouillée ». Son utilisation permet de générer dynamiquement des textures permettant de cartographier en quelque sorte les points d'impacts de particules sur un objet donné. Il en ressort une texture allant du noir au blanc, l'intensité changeant selon le nombre de contacts avec des particules sur un endroit donné.

Afin de pouvoir utiliser les Wetmaps, il est primordial d'avoir déplié à l'avance les UV de l'objet, afin que les informations soient transcrites correctement. Cet outil en soit

n'a pas énormément de paramètres, le plus important permettant de choisir à quelle vitesse nous souhaitons avoir la texture redevenir noire, pour simuler l'objet séchant avec le temps.

Ce test présente donc bien l'utilisation, lors de la phase de shading et de rendu de ses textures animées. En les utilisant comme un masque, il m'a été possible à l'aide de Vray pour Maya de changer la texture du crane d'un rendu craie blanche à un vinyle orange.

Malheureusement, bien que prometteur, je n'ai pu réutiliser cette expérience dans mes recherches futures, n'ayant pas trouvé d'utilité aux wetmaps dans le transfert de particules.

J'ai durant un temps caressé l'idée d'utiliser ces textures afin de générer à partir de ces dernières de nouvelles particules, et par là même de créer un effet secondaire pouvant être esthétiquement plaisant. Cependant, ce genre de recherche ne rentre pas directement dans ma problématique, et ne serait qu'un atout au niveau du rendu.

L'intérêt d'un point de vue rendu justement, est bien évidemment viable. Il peut permettre d'apporter vie à l'environnement et à le faire réagir en quelque sorte à votre au fluide de façon à ajouter de nombreux détails et interactions secondaires après le passage de ce dernier, et tout cela pour très peu de temps de simulation supplémentaires.

Je garde ces idées sous le bras Cette recherche pourrait alimenter pour mon prochain mémoire.

RESEARCH PAPER TEST 10, FILAMENTS:

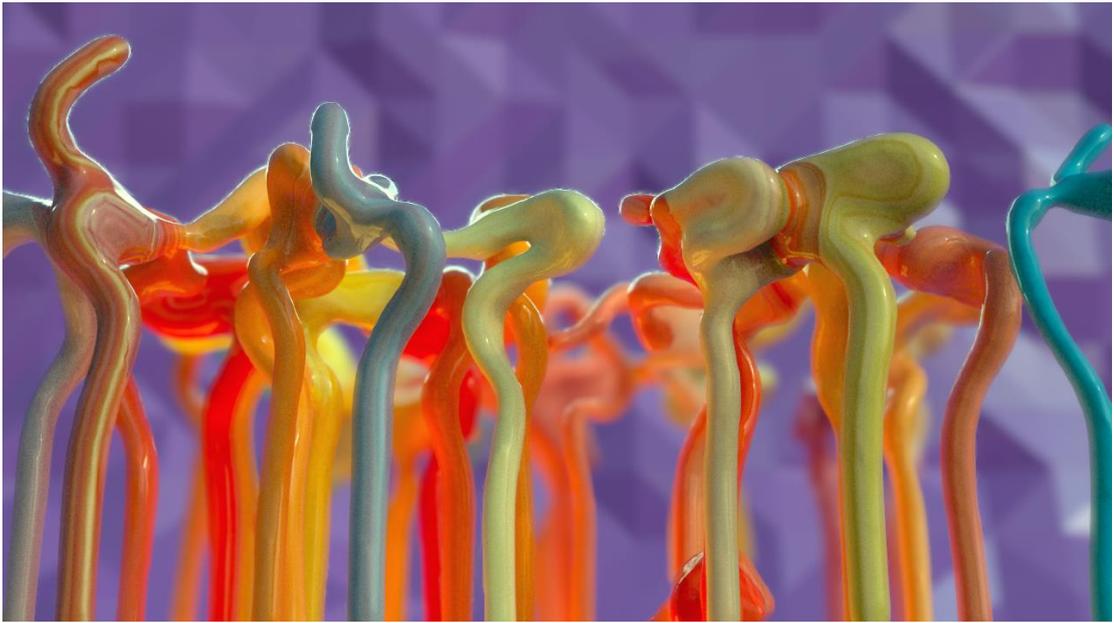


FIGURE 2 REALISATION PERSONNELLE, TEST 10

Ce test est un peu spécial, il a été réalisé juste avant le projet intensif, alors que vivant sur Paris, je n'avais pas accès à tout mon matériel ni ne disposait de beaucoup de temps et d'espace pour travailler sur une problématique avancée.

Ce que je souhaitais réaliser au court de ce test était le genre de filament que l'on peut avoir en pressant un petit tube de crème grasse, ce dernier s'élève et se tourne, il se roule sur lui-même, mais ne gêne jamais vraiment la crème qui sort du tube, il s'élève et forme ses fils qui finissent par retomber sous leur poids, comme une tige ayant grandi trop vite.

Pour réaliser ce test, et afin de ne pas avoir un émetteur pour chaque tige de liquide, ce qui deviendrait compliqué à gérer, j'ai choisi de n'utiliser qu'un seul émetteur utilisant une texture pour définir les points d'émission. Cette texture se compose d'un fond noir sur lequel sont disposés aléatoirement des ronds blancs de divers diamètres. Realflow interprète alors la texture et n'émet de particules qu'au niveau des ronds blancs. L'avantage de n'avoir qu'un seul émetteur pour commencer me permet d'optimiser le temps de simulation, mais également de n'avoir qu'un seul émetteur à régler, et de ne pas avoir à créer plusieurs daemons sur chaque émetteur pour effectuer le même effet.

La scène totale se compose de trois émetteurs : L'émetteur source que je viens de décrire ci-dessous, l'émetteur supérieur et l'émetteur inférieur.

Le fonctionnement de la simulation se déroule ainsi : l'émetteur source émet des particules vers le haut. Au-dessus d'une hauteur donnée, toutes les particules de l'émetteur source sont transférées dans l'émetteur supérieur à l'aide d'un daemon filtre. Les particules de l'émetteur supérieur sont affectées par celles de l'émetteur source, mais pas inversement. Cela veut dire que les particules sources poussent les particules de l'émetteur supérieur sans ralentir ni perdre de force. Ainsi, bien que perdant de la vitesse due à la gravité, les particules supérieures montent plus haut, poussées sans résistance par les particules de l'émetteur source.

Enfin, quand les particules de l'émetteur supérieur retombent en dessous d'une hauteur donné sur l'axe des Y, elles sont attribuées à l'aide d'un autre daemon filtre à un troisième et dernier émetteur, l'émetteur inférieur.

Ce dernier émetteur n'influence ni l'émetteur source ni celui supérieur, mais il est influencé par ces derniers. Ainsi, les particules retombées au sol ne viennent pas gêner les particules sources lors de leur émission, ces dernières pouvant librement s'élancer vers le ciel.

Pour le rendu, n'ayant qu'un seul maillage à la sortie de Realflow, et souhaitant avoir plusieurs nuances de couleur dans le fluide, j'ai choisi d'utiliser une texture en projection, placée au-dessus de la scène. Bien que satisfait du résultat, je reste sur ma faim. J'aurais souhaité pousser le rendu plus loin, avec de la texture translucide, diaphane, et un éclairage volumétrique donnant plus de profondeur à cette réalisation. Le projet intensif arrivant, il m'a fallu alors laisser ce test en son état, et passer à d'autres problématiques plus en accord avec mon sujet de mémoire.

RESEARCH PAPER TEST 12, SCRIPTING:



FIGURE 23 RÉALISATION PERSONNELLE, TEST XHEADMELT

Ce test est le premier réalisé à l'aide d'un script. Realflow n'offrant pas de base une option permettant de faire fondre de manière plausible un objet, j'ai opté pour la programmation python afin d'explorer les options disponibles afin d'offrir l'illusion d'un objet fondant. On peut trouver en ligne des pistes permettant de faire fondre un objet suivant un axe, de haut en bas par exemple, ou permettant de le liquéfier de manière uniforme, de l'extérieur de l'objet vers son centre.

Bien que très intéressantes, les solutions disponibles sur internet n'offrent pas selon moi un contrôle suffisamment poussé pour gérer ce transfert de particules. Elles permettent de le faire seulement d'une manière cartésienne (si c'est au-dessus d'une valeur donnée, tout fond) ou trop globale (quel que soit la face de l'objet, tout fond).

Je souhaitais être capable de transférer les particules d'un émetteur à un autre suivant leur distance par rapport à un point dans l'espace, un repère que je peux ensuite animer et déplacer dans la scène. Avoir un objet dans la scène sur lequel je puisse mettre des clés d'animations m'offre la possibilité de récupérer sa position et d'utiliser cette information dans mon code.

L'idée était de pouvoir recréer l'effet d'une torche s'approchant d'un bloc de glace, ne fondant que quand la source de chaleur est proche, puis durcissant de nouveau une fois cette dernière éloignée.

J'ai, en premier temps, créé un simple script pour faire fondre un objet de manière uniforme selon une valeur donnée, permettant d'intensifier ou stopper complètement la transformation. De cette base, j'ai affiné en ajoutant des conditions en plus, récupérant des informations supplémentaires, tout en espérant que la dernière modification ne fasse planter tout le script.

Pour expliquer rapidement les bases de ce script, je passe toutes les particules composant l'objet rigide dans une boucle qui récupère la position de la particule, le nombre de particules avoisinantes dans un rayon de 0.1, et enfin la distance entre cette particule et l'objet qui symbolise la source de chaleur.

Une condition détermine ensuite que si la particule en question est suffisamment proche de la source de chaleur symbolique, et que cette dernière n'a pas trop de voisins (ce qui indiquerait que la particule n'est pas sur la face extérieure de l'objet qu'elle compose), alors elle est transférée sur l'émetteur de particules qui représente le liquide.

Les premiers tests à petite échelle se révélèrent concluants. J'ai alors tenté d'augmenter la qualité de ma simulation. Malheureusement, le code produit ne réussissait pas à s'adapter au changement d'échelle. Le souci venant de valeurs qui n'augmentaient pas de manière linéaire lors de l'augmentation de la qualité, mais de manière exponentielle, provoquant des résultats non uniformes.

J'ai alors, par soucis de temps passé sur ce problème, décidé de changer de problématique.

Au lieu d'une source de chaleur dont la distance par rapport aux particules permettait de faire « fondre » de plus en plus rapidement les particules à mesure que cette distance se réduisait, je me suis limité à un volume sphérique les faisant fondre de manière uniforme selon un diamètre donné. L'objet en dehors de ce cercle ne fond pas, uniquement la partie présente à l'intérieur de ce rayon est affectée par le changement d'état. Pour l'expliquer de manière plus claire, à la place d'une torche émettant de la chaleur à distance, j'utilise une sphère métallique chauffée à blanc, qui n'affecte que les surfaces en contact direct.

Pour revenir à ce test, j'ai fait passer cette sphère à l'intérieur du crâne, du sommet jusqu'à travers l'un des yeux, liquéfiant tout sur son passage, et provoquant cette éruption liquide une fois que la source de chaleur a complètement traversée.

Intention artistique :

Je souhaite donner l'impression que ce crâne est fait de cire, une cire au reflet rouge, comme parfumée à la cannelle.

Afin d'arriver à un rendu convainquant, j'ai utilisé le Shader SSS de Vray pour la cire à froid, donc le crâne, et un shader transparent, similaire à un shader pour eau, mais avec une couleur de profondeur, afin de récupérer des reflets rouges de la cire, et lui donner plus de profondeur.

RESEARCH PAPER TEST 13, SCRIPTING:



FIGURE 24: RÉALISATION PERSONNELLE, TEST 13 SCRIPTING 02

Ce test est dans la lignée du précédent, c'est-à-dire qu'il n'est pas basé sur un script à part entière, mais sur une évolution de ce dernier.

De nombreux nouveaux défis ce sont présentés à moi au court de sa réalisation.

Premièrement, je souhaitais pouvoir déclencher le changement d'état à partir d'un objet en particulier, et de cet objet seulement. L'intérêt d'un tel code permet d'éviter les transformations intempestives, et offre également un intérêt réel pour de nombreuses scènes.

J'avais en tête l'idée de pouvoir faire fondre un cachet d'aspirine lorsqu'il entre en contact avec le fluide, ou bien de faire geler une étendue d'eau lorsque un personnage la touche du doigt. On peut également imaginer d'autre intérêt, puisque cela permet au même fluide de réagir différemment selon l'objet.

Pour cela, j'ai récupéré le nom de l'objet que j'ai choisi comme déclencheur.

A l'aide d'une liste contenant les particules étant en collision, c'est-à-dire TOUTES les particules, puisque ces dernières rentrent en collision les unes avec les autres, j'ai comparé le nom de l'objet entrant collision avec celui de mon objet choisi pour chaque particule.

Les particules détectées comme étant en collision avec cet objet sont alors transmises dans le second émetteur.

Deuxièmement, je souhaitais qu'une fois le contact effectué, la transformation se face de manière plus organique. J'entends par là que la nature tend à toujours prendre le chemin ayant le moins de résistance. Que l'on observe une mare qui gèle, une buche brulant, un rocher soumis à l'érosion, ou un simple glaçon fondant dans un verre, toutes ces transformations d'un état à un autre se font de l'extérieur vers l'intérieur.

Je devais donc introduire dans mon script la notion de réaction en chaîne, qu'une action à un endroit enclenche une réaction qui se propage de son épicerie le long de la surface de l'objet, affectant les zones les plus exposées en première, avant de concerner peu à peu le centre.

La solution technique réalisée pour résoudre est assez simple :

La collision de l'objet avec le premier système de particules a produit un transfert de ses particules vers un deuxième système de particules. Les particules de ce deuxième réseau sont cataloguées dans une liste, et cette liste est ensuite utilisée par une boucle les testant toutes.

Pour chaque particule, la boucle crée une liste des particules avoisinantes. Cette liste est ensuite examinée, et toute particule contenue dans ladite liste se voit examinée pour l'identité de l'émetteur à laquelle elle est rattachée. Si cet émetteur diverge de celui contenant les particules du second réseau, alors cette dernière est retirée de son système de particule initial et attribuée au second réseau, ajoutant alors son poids dans la réaction en chaîne, qui à chaque nouvelle frame a plus de particules, donc en scanne plus à chaque fois, et en récupère encore plus, etc. Tout cela continue jusqu'à épuisement de particules sources.

Ce test m'a permis d'apprendre à optimiser mon code. Quand une simulation commence à avoisiner le million de particules, avoir un script qui garde toutes les particules dans des listes peut devenir très rapidement impraticable en raison du temps de calcul qui augmente exponentiellement. J'ai mis en place de nombreuses conditions permettant de limiter au maximum les listes lourdes qui n'en finissent plus. Il m'a été également possible de découvrir une autre commande permettant de parcourir automatiquement toutes les particules d'un émetteur en particulier sans avoir à les implémenter dans une liste.

Intensions Artistique :

Travailler sur la réaction en chaîne m'a fait penser aux colonies de bactéries, évoluant au fur et à mesure, s'élargissant autour d'un point d'origine.

Je voulais donner l'impression d'une réaction chimique provoquée quand un liquide et un élément externe se rencontrent.

Je souhaitais travailler un peu plus la texture des fluides, leur donner un côté plus velouté, presque crémeux, mais les temps de rendu n'ont pas été appréciés, et il m'a fallu me résigner à un rendu plus fantaisiste.

J'apprécie cependant le résultat final, il reste sobre et se maintient, le béton ciré en fond lui donnerait presque la place d'une immense installation dans un musée d'art contemporain.

PROJET INTENSIF DE 3 SEMAINES :



FIGURE 26 PROJET INTENSIF DE 3 SEMAINES, PLAN 00

Je souhaite faire un court aparté sur le travail réalisé durant le projet de 3 semaines de janvier 2014, réalisé en collaboration avec Eddy Camilleri et Gaspard Imbert. Afin d'intégrer mon sujet de recherche à ce projet artistique, j'ai eu de pouvoir faire fondre des parties du cosmonaute afin d'y apporter un élément de simulation en complément du travail de Gaspard Imbert.

Le respect du cahier des charges et des délais envisagés pour la production d'un projet de 3 semaines sont rarement en corrélation avec la réalité d'un si court délai. Il m'a fallu reconsidérer mon objectif et assumer d'autres travaux que ceux prévus afin de privilégier la réussite technique et artistique de ce court métrage avant mes recherches de mémoire.

Bien qu'ayant dû m'occuper du lighting, shading, rendu et composition du film, j'ai pu cependant travailler un peu avec les simulations de fluide. Il n'y a hélas aucun changement d'état, mais j'ai cependant accru ma connaissance du logiciel en ayant à gérer son implémentation dans un workflow. J'ai appris à mieux gérer les imports et exports du logiciel ainsi que 'a optimiser la taille et le rendu des maillages générés.

Les nombreuses difficultés dont j'ai eu à faire face, bien que n'ayant pas un lien direct avec le sujet principal de mon mémoire, m'ont permis d'accroître ma maîtrise du logiciel, et par la même d'augmenter mes recherches et ma production pour les tests ultérieurs.

QUATRIEME PARTIE: LE CHANGEMENT D'ETAT DANS LA SIMULATION DE FLUIDES :

LES POINTS CLES DU CHANGEMENT D'ETAT DU LIQUIDE AU SOLIDE, APPROCHE NODALE ET SCRIPTEE:

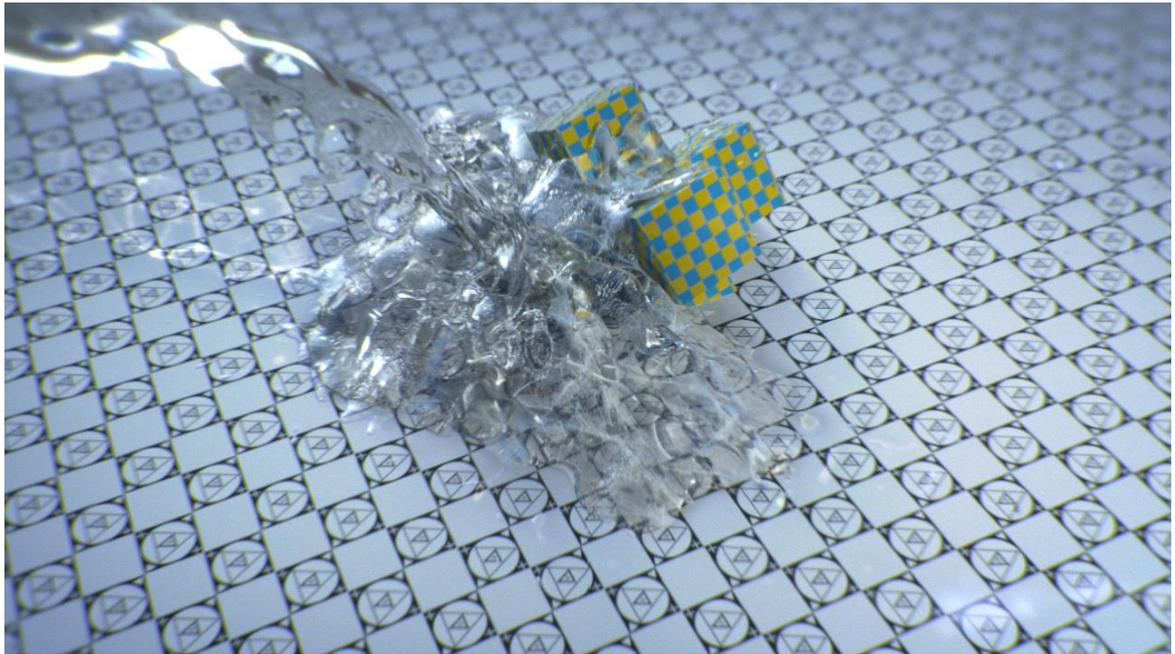


FIGURE 26 REALISATION PERSONNELLE, TEST ICING.

Dans cette section, je vais vous diriger pas à pas à travers les différentes étapes nécessaires afin de faire passer de l'eau ou tout autre liquide d'un état liquide à un état solide. J'expliquerai à partir d'une scène plus simple les étapes à effectuer pour parvenir au résultat présenté dans la vidéo suivante : <http://vimeo.com/rossetto/rpt09>

La société Next Limit a partagé cette vidéo auprès de ses utilisateurs sur internet. Cela me procure une immense satisfaction et un sentiment de reconnaissance sur le domaine de recherche que j'ai sélectionné.

Ce premier tutoriel ne nécessite aucune connaissance de programmation et n'utilise que les outils disponibles au sein de Realflow.

La version Anglaise de Realflow 2013 à été utilisé pour la réalisation de ce test, cependant les techniques utilisées sont également disponibles sous les versions précédentes. L'emplacement des boutons ainsi que leur dénomination peut avoir changé.

APPROCHE NODALE:

1 .COMPOSANTS NECESSAIRES DE BASE:

1 Emitter: J'utiliserai pour cet exemple le cercle Emitter, un émetteur émettant les particules à partir d'un cercle.

1 Container Emitter: Le Container Emitter est un émetteur n'émettant pas de particules. Il est possible de changer le comportement des particules qu'il contient, mais ces dernières doivent provenir d'un autre émetteur. Il est nécessaire d'avoir deux émetteurs dans cette scène afin de pouvoir appliquer des forces et des paramètres différents pour avoir un réseau agissant comme un liquide, et un autre agissant comme un solide.

1 Gravity Daemon: Ce daemon applique une force extérieure sur les systèmes de particules suivant un axe et une force donnés. Il est utilisé afin de simuler la Gravité dans notre scène, appliquant une force de 9.8 dans l'axe négatif de Y.

2 Filter Daemon: Ce daemon prend pour paramètres un réseau de particules sources et un autre réseau de destination. Il transfère les particules de l'un à l'autre suivant un paramètre donné.

1Kill daemon: Ce daemon est représenté par un cube redimensionnable. Toutes les particules appartenant à un émetteur connecté au Kill daemon sera supprimée à chaque nouvelle itération de la simulation.

1Kill Speed daemon: Ce daemon permet de limiter la vitesse des particules en supprimant ou plafonnant à une vitesse donnée toutes les particules dépassant cette valeur. Il sera utilisé afin de figer les particules représentant l'état solide.

1 Plane: Le plane est la géométrie la plus simple disponible dans Realflow, il s'agit d'une simple surface plate, idéale pour représenter le sol.

1 Noise daemon: Ce daemon n'est pas indispensable, mais permet d'ajouter des variations à l'émission des particules du Circle Emitter.

2. MISE EN PLACE:

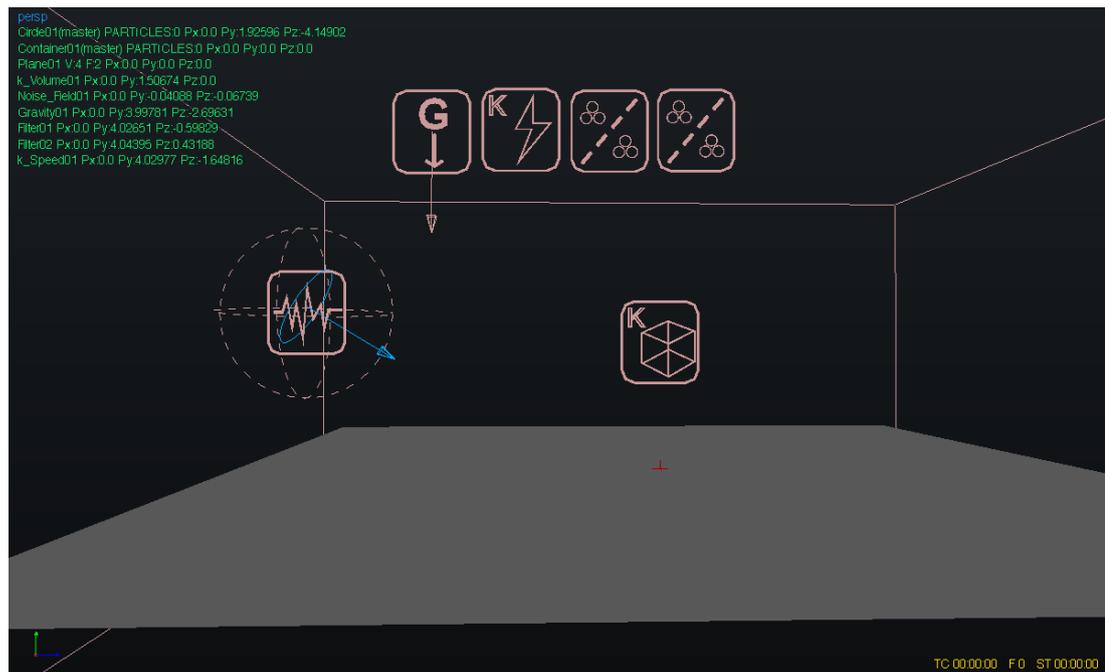


FIGURE 27: REALISATION PERSONNELLE, TUTORIEL 0

Créez tous les éléments énoncés ci-dessus, et renommez les si vous le souhaitez. C'est une bonne habitude à prendre quand le nombre d'éléments dans votre scène devient important. Pour la clarté de ce tutoriel, je vais utiliser les noms de base. Sélectionnez le circle Emitter (Circle01) et élevez le dans l'axe des Y afin de mieux le voir. Vous pouvez changer dans la fenêtre *Node Params* > *Particles* ses propriétés si vous souhaitez simuler un type de fluide en particulier. Ce tutoriel ne portera pas là-dessus.

Sélectionnez le Noise daemon (Noise_Field01). Dans la fenêtre *Node Params* > *Noise Field*, mettez le paramètre « Bounded » en Yes. Le Noise daemon est maintenant représenté par une bulle en pointillé, et n'influencera les émetteurs auquel il est attaché qu'à l'intérieur de ce cercle. Déplacez le vers le haut afin qu'il englobe Circle01, puis dans la fenêtre *Node Params* > *Node*, cliquez sur Parent to et sélectionnez Circle01. Noise_field01 est maintenant parenté à Circle01, quand vous bougez Circle01, Noise_field01 va suivre.

Enfin, sélectionnez le Kill volume (K_volume01). Dans la fenêtre *Node Params*, ouvrez K Volume, puis cliquez sur Fit to Scene. Le Kill volume englobe maintenant toute votre scène.

Les daemons K_speed, Gravity et Filter affectent toute la scène par défaut, l'icône les représentant chacun peut alors être déplacé et organisé comme bon vous semble dans la scène. Les aligner et les déplacer du centre de la scène aide à la lisibilité.

3. CONFIGURER LES DAEMONS ET NODE:

K_Speed01: Ce daemon va être utilisé afin de figer l'état solide. Dans la fenêtre *Node Params > k Speed*, attribuez la valeur 0.0 à Max Speed puis Yes à Limit & Keep.

Filter_01: Ce daemon est au centre du changement d'état, c'est grâce à lui que le liquide va se solidifier selon un paramètre donné. Dans la fenêtre *Node Params > Filter*, choisissez Circle01 comme *Source Emitter* et Container01 comme *Target emitter (true)*. Choisissez *lower than* comme condition et *Speed* comme *Attribute*. Une valeur de 0.1 est un bon début comme valeur de la condition, vous pourrez toujours le calibrer ultérieurement. Si vous souhaitez gagner en volume lors du changement d'état, comme les molécules d'eau se dilatant lors du gel, vous pouvez mettre l'option *split* sur *yes*. Cela va dupliquer la particule lors de son transfert.

Filter_02: Ce filter Daemon est ajouté à la scène pour consolider le transfert des particules, il va effectuer le transfert non pas selon leur vélocité mais leur âge. Filter_02 a les mêmes paramètres que Filter_01 pour *Source Emitters* et *Target Emitter (true)*. Pour la condition, choisissez *Greater than* et *Age* pour *attribute*. Une valeur de 10 peut être mise par défaut, mais cela dépendra de votre scène.

Plane_01: Pour éviter un effet Leidenfrost sur nos particules, nous allons augmenter la friction de Plane_01. Sélectionnez Plane_01, puis dans la fenêtre *Node Params > Particle Fluid Interaction*, passez le paramètre *Particle friction* à 0.1.

4. CONFIGURER LE RELATIONSHIP EDITOR

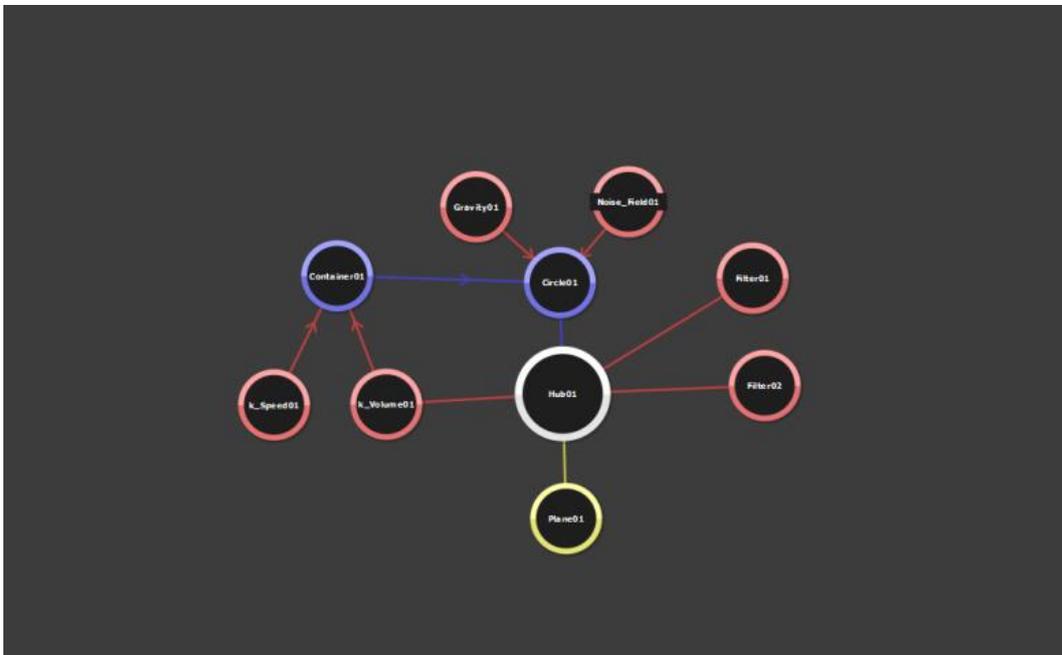


FIGURE 28 REALISATION PERSONNELLE, TUTORIEL 0

Une fois que nos nodes sont placés et configurés, il nous reste à définir leur relation. Certains daemon ne doivent affecter qu'un seul réseau de particules, certains deux, etc.

De base, tous les nodes créés sont reliés à un *hub* (Hub01). Ils interagissent avec tous les autres, et sont affectés de même.

Dans la fenêtre *Relationship Editor*, déconnectez Noise_Field01 et Gravity01 de Hub01 et connectez-les à Circle01. Déconnectez également Container01 et connectez-le à Circle01. Ainsi, Container01 influence Circle01, mais pas l'inverse. Enfin, connectez k_volume01 a Container01 et déconnectez k_speed01 de hub01 et connectez le à Container01.

La scène est mise en place, il ne vous reste plus qu'à tester la simulation, augmenter les substeps et la qualité de vos émetteurs si nécessaire.

APPROCHE SCRIPTEE:

Il est possible d'atteindre le même résultat avec un simple script python. Pour ce faire, gardez la scène crée pour l'approche nodale et supprimez les daemon Filter01 et Filter02 ainsi que k_Speed01. C'est précisément le rôle de ces trois nodes qui va être remplacé par un script.

Allez dans le menu *Layout*, puis cliquez sur *simulation Flow*. Vous pouvez également utiliser le raccourci Crtl + F2

Nous souhaitons que le script s'applique à chaque step. Dans la fonction « onSimulationStep » copiez le code suivant :

```
def onSimulationStep():
    liquid = scene.get_PB_Emitter("Circle01")
    solid = scene.get_PB_Emitter("Container01")
    liquidParticlesList = liquid.getParticles()
    solidparticlesList = solid.getParticles()

    for particle in liquidParticlesList:
        vel = particle.getVelocity()
        partAge = particle.getAge()
        if vel.module() < 0.2:
            solid.addParticle(particle.getPosition(),particle.getVelocity())
            liquid.removeParticle(particle.getId())
        if partAge > 1:
            solid.addParticle(particle.getPosition(),particle.getVelocity())
            liquid.removeParticle(particle.getId())

    for particle in solidparticlesList:
        particle.freeze()
```

Analysons ensemble le code pas à pas :

Afin que le code fonctionne, nous devons indiquer à Realfow quels sont les émetteurs sur lesquels nous souhaitons travailler :

```
liquid = scene.get_PB_Emitter("Circle01")
solid = scene.get_PB_Emitter("Container01")
```

Nous créons une variable, et allons chercher dans la scène l'émetteur correspondant. Une fois les émetteurs récupérés, et afin de pouvoir étudier particule après particule, il nous faut créer une liste les répertoriant :

```
liquidParticlesList = liquid.getParticles()
solidparticlesList = solid.getParticles()
```

Première étape, je récupère les émetteurs, deuxième étape, je récupère une liste des particules contenues dans cet émetteur. Nous avons à présent à disposition les informations nécessaires pour créer une boucle afin de passer en revue chaque particule émise par Circle01 afin de la figer ou non.

```
for particle in liquidParticlesList:
    vel = particle.getVelocity()
    partAge = particle.getAge()
    if vel.module() < 0.2:
        solid.addParticle(particle.getPosition(),particle.getVelocity())
        liquid.removeParticle(particle.getId())
    if partAge > 1:
        solid.addParticle(particle.getPosition(),particle.getVelocity())
        liquid.removeParticle(particle.getId())
```

Nous utilisons une boucle *for*, qui pour chaque élément d'une liste donnée, va effectuer une série d'action.

Pour chaque particule dans la liste des particules de Circle01, nous récupérons sa vitesse et son âge, stocké respectivement dans les variables *vel* et *partAge*.

Nous utilisons la condition *if*, afin de comparer ces variables avec une valeur donnée.

Si la vitesse de la particule est inférieure à 0.2 alors...

Si l'âge de la particule est supérieur à 1 alors...

```
solid.addParticle(particle.getPosition(),particle.getVelocity())
liquid.removeParticle(particle.getId())
```

La première ligne ajoute une particule au Container01 en récupérant la position et la vitesse de la particule contenue dans Circle01 et étant testé à ce moment dans la boucle. La deuxième ligne retire ensuite ladite particule de Circle01.

Le transfert est alors réalisé.

Il ne reste plus qu'à geler sur place les particules contenues dans Container01 afin de simuler un état solide.

```
for particle in solidparticlesList:
    particle.freeze()
```

Pour chaque particule dans la liste des particules de Container01, gèle la particule.

Cette dernière est alors fixée dans l'espace.

Ce code est simple et ne réalise pas le dédoublement des particules lors du transfert comme le faisait le filter Daemon, il permet cependant d'illustrer le potentiel de personnalisation que permet de faire la programmation.

DU SOLIDE AU LIQUIDE, APPROCHE NODALE ET SCRIPTEE :

APPROCHE NODALE :

Dans cette partie, je vais vous expliquer comment ai-je réalisé mon premier test de changement d'état.

Il est assez basique dans son fonctionnement, mais m'a permis de me de donner une base sur laquelle progresser et réaliser un test beaucoup plus poussé que j'aborderai dans l'approche scriptée du passage du solide au liquide.

Ce test est visible à l'adresse suivante : <http://vimeo.com/rossetto/rpt06>

Ce test a été réalisé comme une preuve de concept. J'ai réalisé cette création vers le début du mémoire, alors que j'expérimentais sur différents outils offert par Realflow. Il m'a permis de réaliser que non seulement un changement d'état utilisant ce logiciel était possible, mais également esthétiquement appréciable. C'est à la suite de sa création ainsi que du retour reçu par la communauté web, que j'ai décidé de choisir le changement d'état comme problématique de mémoire.

MISE EN PLACE DE LA SCENE :

Cette scène est très simple de par son fonctionnement, et n'a nécessité que 7 nodes différents :

- **Skull** : C'est un maillage de crâne humain réalisé sous Maya et triangulé pour permettre son utilisation sous Realflow. Il présente une plus grande complexité que les maillages proposés de base dans Realflow.
- **Fill_Object01** : C'est un node émetteur ayant pour caractéristique de permettre le remplissage d'une géométrie donnée de particules, me permettant ainsi d'avoir un volume de particules ayant la forme d'un objet défini, à condition d'avoir un niveau de résolution suffisamment élevé pour mettre en valeur tous les détails.
- **Container01** : Ce node émetteur est utilisé pour récupérer les particules transférées de Fill_Object01 lors du changement d'état. Il possède les mêmes caractéristiques de résolution (4.0) et viscosité (8.0) que Fill_Object01, afin d'obtenir un liquide uniforme et contigu, mais suffisamment fluide pour simuler la liquéfaction du crâne.
- **Sphere01** : Il s'agit d'un maillage d'une simple sphère. Bien que non visible au rendu, elle est utilisée comme déclencheur permettant le changement d'état des particules. Son rôle exact sera décrit plus en détail plus bas.
- **Filter01** : Ce daemon est utilisé pour réaliser le transfert des particules. Il prend comme émetteur source Fill_Object01 et comme émetteur de destination Container01. La condition du transfert est basée sur la vitesse des particules de l'émetteur source, qui au-dessus d'une valeur donnée, sont transférées vers l'émetteur de destination.
- **Gravity01** : Ce daemon est utilisé pour simuler la gravité sur le réseau de particule représentant l'état liquide (Container01).
- **Cube_top_Open** : Ce maillage fait office de réceptacle afin que le liquide reste sur la scène, il dispose de rebord pour éviter qu'il ne déborde. De plus, sa friction et viscosité ont été augmentées pour garder plus de volume et qu'il ne s'éparpille pas trop.

RELATIONSHIP EDITOR :

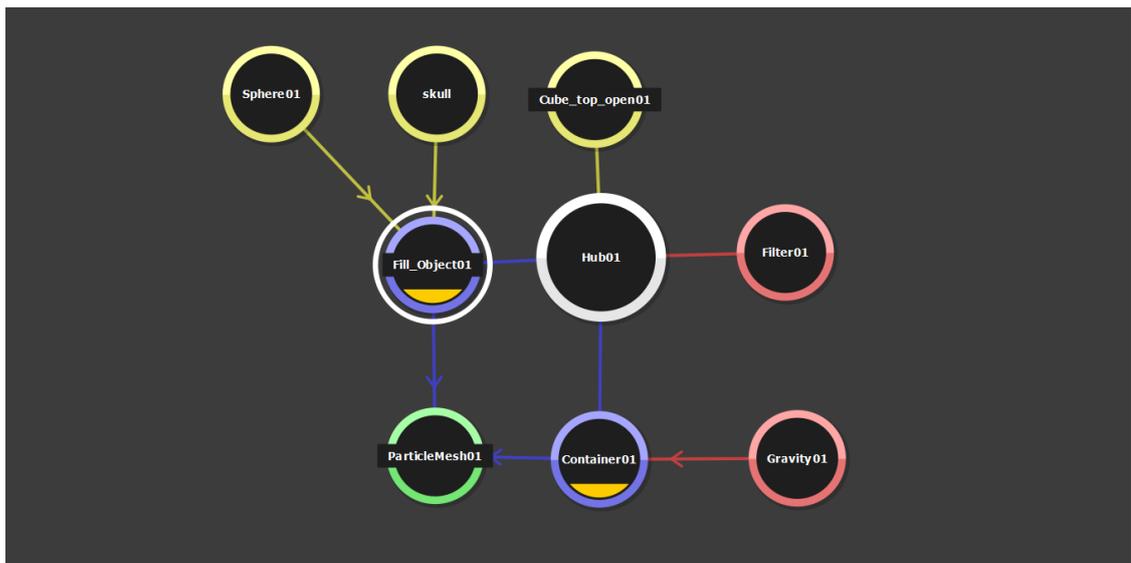


FIGURE 29 REALISATION PERSONNELLE, TUTORIEL 03

L'émetteur Fill_Object01 est connecté avec le maillage du crâne (skull) et à la sphère. Il n'est pas affecté par le Gravity daemon. Les particules étant contenues dans le crâne, elles ne peuvent sortir de ce dernier, et n'ayant pas de force externe les influençant, elles restent en place.

L'émetteur Container01, qui représente le liquide dans le changement d'état, n'est affecté que par le Gravity daemon et Cube_top_Open qui représente le sol.

FONCTIONNEMENT :

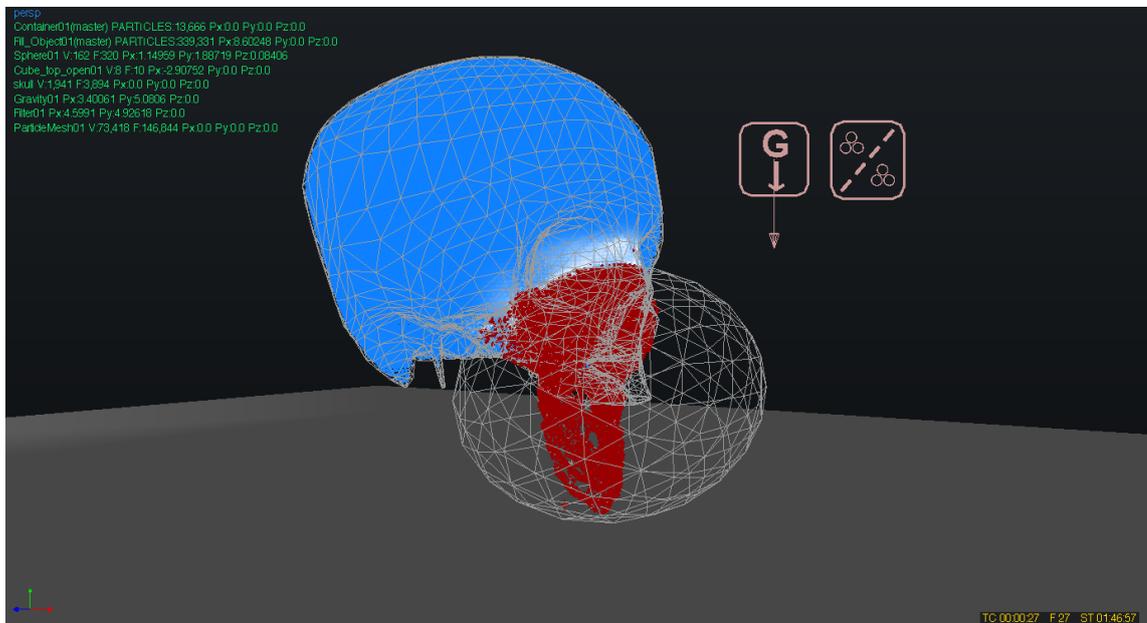


FIGURE 30 REALISATION PERSONNELLE, TUTORIEL 04

La sphere01 est animée, elle se déplace vers le maillage du crâne jusqu'à l'interpénétrer. Lorsque ceci ce produit, les particules de Fill_Object01 entrent alors en collision avec la sphère et sont repoussées.

Ce mouvement accentue leur vitesse jusqu'à rentrer dans la condition du filter, qui les transfère dans l'émetteur Container01. Une fois transférées, elles se retrouvent influencées par le gravity daemon, et ne rentrant plus en collision avec la sphère ou le crâne, elles se mettent à couler vers le bas et entre enfin en collision avec le sol. La valeur de condition du filter01 est alors graduellement abaissée, entraînant une réaction en chaîne, car les particules des deux émetteurs ayant une certaine viscosité, les particules « liquide » tirent légèrement sur les particules « solide », leur donnant suffisamment de vitesse pour que ces dernières soient à leur tour transférées.

Ce test présente cependant quelques soucis :

Les particules de Fill_Objects n'étant pas figées par l'utilisation d'un daemon ou d'un script, bien que non influencées par une force externe, subissent cependant l'action de la force interne du fluide, dépendant de la pression interne et externe, sa viscosité, sa tension de surface, etc... qui leur est attribuée.

Cela provoque des variations dans leur organisation, donnant alors au rendu final du crâne une fois meshé un volume qui change légèrement.

L'utilisation d'une sphère poussant sur les particules et les comprimant dans un espace restreint peut résulter en des forces trop grandes et non réalistes, provoquant un effet d'explosion sur certaines particules, qui se détachent du reste et s'éparpillent sur toute la simulation. L'utilisation d'un *Kill isolated particles* daemon ou l'augmentation des substeps peut résoudre le problème, mais rend la simulation plus gourmande en temps de calcul.

DU SOLIDE AU LIQUIDE, APPROCHE SCRIPTEE :



FIGURE 3.1: REALISATION PERSONNELLE, TEST 13, SCRIPTING 03

La scène étudiée dans cette partie est le résultat le plus avancé de mon année de recherche, c'est l'aboutissement de mes différents tests mais également le test le plus poussé techniquement.

Vous pouvez visualiser le test à l'adresse suivante : <http://vimeo.com/rossetto/rpt13>

Le but de ce test était de réunir en une seule scène, les trois états de la matière : solide, liquide, gazeux. J'ai voulu, comme l'expression anglaise le dit si éloquemment, « to end with a bang ».

Quoi de mieux, pour rendre honneur à cette expression, qu'une réaction en chaîne explosive ?

Partir d'une composition sobre et design dans la matière, monochrome de neutralité, et y insérer un élément perturbateur, insignifiant à première vue mais qui par son intrusion dans l'imperturbabilité de cette vanité placide, contrastant avec le calme plat apparent, déclenche une propagation d'énergie pure et insoupçonnée, aussi brutale qu'éphémère.

A peine a t'on le temps de saisir ce qu'il se passe qu'elle est déjà finie, ne laissant derrière elle qu'un nouvel ordre établi, un équilibre renouvelé jusqu'à la prochaine étincelle qui réveillera le brasier.

Il faut se méfier de l'eau qui dort.

MISE EN PLACE DE LA SCENE :

La scène est composée de trois émetteurs de particules, trois maillages, six daemons et un null :

- **Skull:** Ce maillage est utilisé comme moule pour disposer d'une représentation du crâne en particules, il n'est aucunement utilisé dans la simulation des particules.
- **Ground:** Ce maillage est utilisé pour simuler le sol.
- **Object :** Ce maillage est la sphère provoquant la réaction en chaîne.
- **Solid_PB:** Emetteur de type fill_Object, utilise le skull.
- **Liquid_PB:** Emetteur de type Container, reçoit ses particules de Solid_PB.
- **Liquid2_PB :** Emetteur de type Container, reçoit ses particules de Liquid_PB
- **Drag_Force01:** Daemon permettant de simuler la résistance de l'air, ralentissant exceptionnellement la vitesse des objets et particules dans la scène.
- **Filter01:** Daemon utilisé en parallèle du script pour transférer les particules de Liquid_PB à Container01
- **Gravity01:** Daemon Simulant la gravité
- **Noise_Field01:** Daemon Ajoutant du bruit afin d'ajouter des perturbations d'aléatoire à la simulation
- **K_isolated01:** Daemon supprimant les particules ayant été isolées.
- **Null01:** Ceci n'est qu'un simple locator, un repère dans la scène utilisé dans le script pour récupérer une valeur animée. Il n'a aucune influence directe sur les éléments présents dans la scène.

RELATIONSHIP EDITOR :

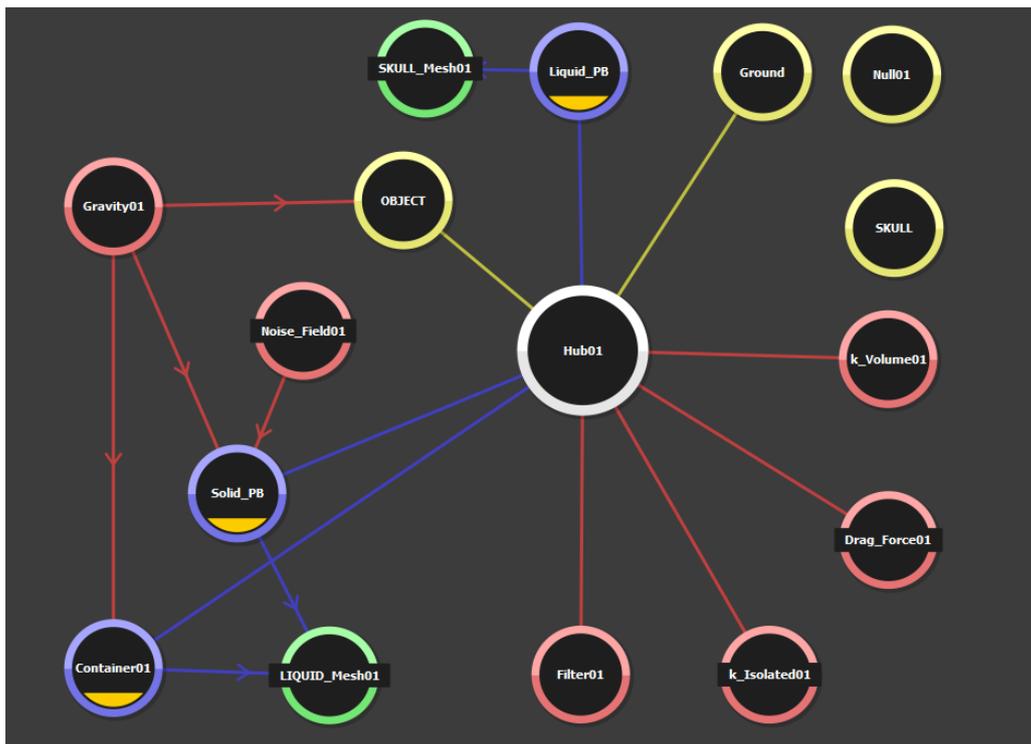


FIGURE 3: REALISATION PERSONNELLE, TEST SCRIPTING 3D RELATIONSHIP EDITOR.

Le daemon Gravity01 affecte Liquid_PB, Liquid2_PB et Object, ce dernier étant un maillage dynamique pouvant être soumis à des forces externes.
Liquid_PB est la phase explosive du changement d'état, il est le seul à être affecté par Noise_Field01.
Les autres nodes sont tous reliés les uns aux autres, à l'exception de Skull et Null01.

EXPLICATION DU SCRIPT :

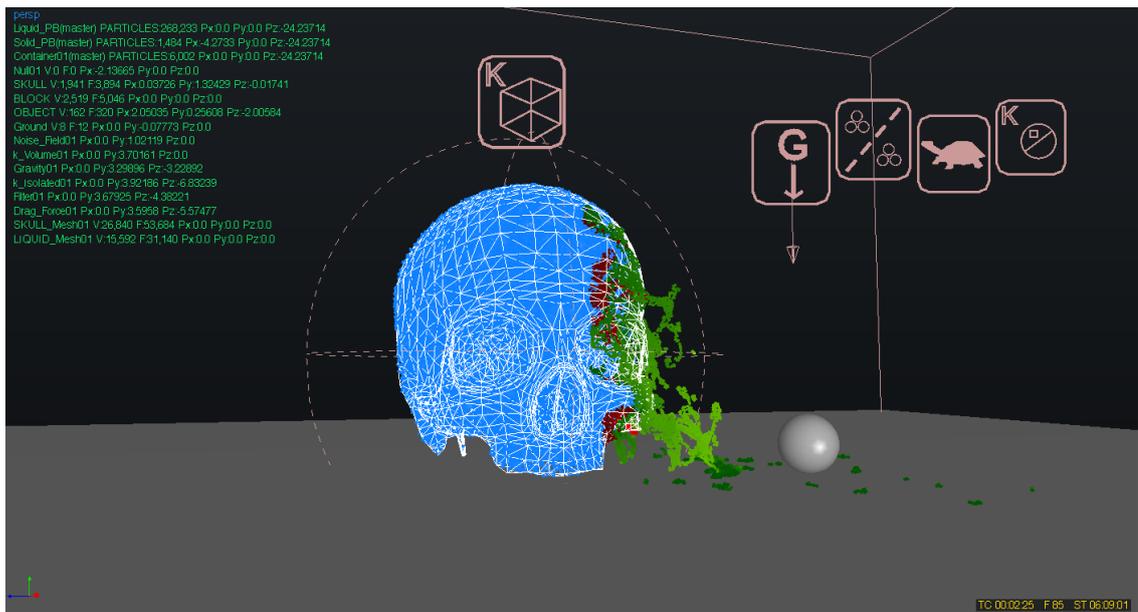


FIGURE 3B REALISATION PERSONNELLE, TEST SCRIPTING 3D PARTICLES VIEW.

C'est ici que la magie opère, c'est toujours une agréable surprise de voir un script relativement court mais dont le résultat dépasse ses attentes.

Je vais vous expliquer son fonctionnement section après section, le code en un block est disponible en annexe.

Ce script a été perfectionné au fur et à mesure de mes recherches, une version avec une fenêtre d'utilisation a été créée, mais afin de simplifier sa lecture, nous allons étudier ensemble sa version sans variables globales. Le script complet est disponible en annexe 1.

Ce code est séparé dans deux fonctions s'effectuant à un rythme différent au court de la simulation.

La fonction « onSimulationStep » s'effectue pour chaque substep composant une frame, soit pour cette scène entre 30 à 300 fois par images, soit de 750 à 22500 fois la seconde en 25 images par seconde.

```
#-----  
# Function: onSimulationStep  
#-----  
def onSimulationStep():
```

```
Obj1 = "OBJECT"
```

L'un des principaux objectifs de ce script était de ne pouvoir déclencher la réaction en chaine que si un maillage en particulier rentrait en contact avec le réseau de particules. Dans le cas du changement d'état du liquide au solide, le réseau de particules liquides doit être contenu dans un réceptacle, il est donc en contact avec un maillage et sans cette condition, il gèlerait de manière intempestive. C'est pourquoi une variable est créée contenant le nom du maillage provoquant la réaction.

```
source = scene.get_PB_Emitter("Solid_PB")  
target = scene.get_PB_Emitter("Liquid_PB")  
targetparticlesList = target.getParticles()  
sourceParticlesList = source.getParticles()
```

Tout comme le script utilisé dans le changement d'état du liquide au solide, on stocke dans des variables les émetteurs utilisés. Une liste des particules de chaque émetteur est respectivement créée.

```
for p in sourceParticlesList:
```

```
colObj = p.getCollidingObject()
```

Première boucle permettant de détecter les particules de solid_PB entrants en collision avec OBJECT. Pour chaque particule dans la liste des particules de solid_PB, stocke dans la variable *colObj* les maillages rentrant en collision.

```
if colObj != None:
```

```
Objname = colObj.getName()
```

Avec cette condition, on ne filtre que les particules entrant en collision avec un maillage.

Si la variable *colObj* n'est pas vide, stocke le nom du maillage entrant en collision dans la variable *Objname*.

```
if Objname == Obj1:
```

```
target.addParticle(p.getPosition(),p.getVelocity())
```

```
source.removeParticle(p.getId())
```

Enfin, si le nom contenu dans la variable *Objname* correspond au nom contenu dans la variable *Obj1*, copie la particule dans *Liquid_PB* et supprime la de *Solid_PB*.

```
for p in sourceParticlesList:
```

```
p.freeze()
```

Cette boucle assure que les particules de Solid_PB soient et restent figées.

La seconde partie du code n'est utilisée qu'une seule fois par frame, elle consiste principalement en la gestion de la réaction en chaîne. Cette réaction contrôlant le transfert des particules, ce code ne peut être mis dans la fonction *OnSimulationStep* sous peine d'avoir toutes les particules transférées en moins de 2 frames.

```
#-----
```

```
# Function: onSimulationFrame
```

```
#-----
```

```
def onSimulationFrame():
```

```
source = scene.get_PB_Emitter("Solid_PB")
```

```
sourceID = source.getId()
```

```
sourceNumber = source.getParameter("Existent Particles")
```

```
target = scene.get_PB_Emitter("Liquid_PB")
```

```
targetparticlesList = target.getParticles()
```

```
sourceParticlesList = source.getParticles()
```

Comme dans la précédente fonction, nous connectons l'émetteur Solid_PB à une variable, mais récupérons également l'id de l'émetteur ainsi que le nombre de particules contenues dans ce dernier. Nous associons également l'émetteur Liquid_PB à une variable et créons deux listes contenant leurs particules respectives.

```
thresholdLevel = scene.getObject("Null01").getParameter("Position").getY()
```

```
threshold = 45 + thresholdLevel
```

Il n'est pas possible dans le code d'animer une valeur avec des clés d'animation. Si nous souhaitons changer la sensibilité d'un paramètre, nous devons récupérer une valeur, ici la position dans l'axe des Y, d'un objet présent dans la scène, le locator Null01.

Nous stockons alors dans une variable une valeur de seuil, de base 45 mais pouvant évoluer en déplaçant le locator.

```
if sourceNumber < 250 :  
for p in sourceParticlesList:  
target.addParticle(p.getPosition(),p.getVelocity())  
source.removeParticle(p.getId())
```

Cette section de code permet de donner un nombre minimum de particules contenues dans l'émetteur source à partir duquel toutes particules restantes sont transférées dans l'émetteur cible. Cette condition permet d'éviter de se retrouver en fin de simulation avec quelques particules par ci par là, figées en l'air, la réaction en chaîne étant passée à côté, ridicule.

La section du code qui suit est le cœur même de la réaction en chaîne.

```
for Sp in targetparticlesList:  
targetID = Sp.getEmitterId()  
Neighbors = Sp.getNeighbors(0.1)  
numNeighbors = len(Neighbors)
```

Pour chaque particule de *Liquid_PB*, on récupère l'id de son émetteur dans une variable, et on stocke les particules présentes dans un rayon de 0.1 autour de la particule dans une liste (*numNeighbors*). On compte alors le nombre de particules présentes dans cette liste et on stocke le résultat dans la variable *numNeighbors*.

```
if numNeighbors < threshold:  
for p in Neighbors:  
emitterID = p.getEmitterId()  
if emitterID == sourceID :  
target.addParticle(p.getPosition(),p.getVelocity())  
source.removeParticle(p.getId())
```

Si le nombre de particules entourant la particule analysée est inférieure au seuil fixé, on passe alors toutes les particules contenues dans la liste des particules voisines dans une boucle qui vérifie l'id de la particule. Si cet identifiant est le même que l'émetteur source, alors la particule est copiée dans l'émetteur cible et l'originale est supprimée.

Le tri basé sur le nombre de particules voisines est nécessaire afin de rester à la surface de l'objet. Les particules situées à la périphérie ont moins de voisines, elles sont donc affectées par le changement d'état, ce qui permet à la réaction en chaîne de rester sur la surface et de s'enfoncer au fur et à mesure que les couches supérieures ont été transférées.

Le contrôle d'identifiant des émetteurs est nécessaires lorsque plus de deux émetteurs sont présents dans la scène, cela permet d'éviter que les particules de *Liquid2_PB* ne soient réintégréés à *Liquid_PB* et permet ainsi de ne pas créer une boucle infinie.

La fumée a quant à elle été réalisée avec Maya Fluids en utilisant les particules importées de Realfow. Afin de n'émettre de la fumée et du feu que sur la partie la plus récente du transfert de particules, il me fallait pouvoir isoler ces dernières, d'où l'utilisation d'un filter daemon basé sur l'âge et l'existence d'un troisième émetteur.

CONCLUSION :

Il est toujours amusant de retracer le fil d'une conversation. Comment sommes-nous en arrivés à parler de ça ?

De même, comparer le premier et le dernier test effectués durant cette année montre qu'ils n'ont pas ou peu de lien. Pourtant, si je retrace tout ce qui s'est passé entre les deux, chaque recherche et expérimentations, lecture et visualisation, le lien qui les unit devient alors manifeste.

Mis en rapport avec mes années d'études à Paris 8, il est plaisant d'observer mon évolution.

N'ayant aucune connaissance en 3D au début de mon année de L3, j'ai pu voir mon habileté évoluer, contempler des tâches insipides de complexités devenir seconde nature.

Mon regard sur le monde du cinéma a changé, j'ai commencé à voir au-delà des pixels, de l'autre côté du miroir, là où la vraie magie s'opère, ou là beauté d'un plan cinématographique est au combien embellie par l'effort fourni pour le rendre réel.

J'ai vu ma vision du monde se muer également. Prendre le temps d'admirer la lumière jouer avec la translucidité des feuilles d'un grand chêne, m'émerveiller des voluptueuses nuances de la fumée de mon cigare, danse exquise et éphémère. Devoir recréer le monde me pousse à en apprécier au combien plus sa subtile et complexe beauté.

J'ai vu, au court de cette année de recherche, une problématique se transformer en solutions. De cette incertitude technique, j'ai su me forger le fer de lance de mes connaissances qui me permettra de percer dans le domaine professionnel.

Les résultats de mes recherches, bien que présentant des solutions viables et prouvant qu'il est possible de transmuter la matière à l'air de systèmes de particules, ne sont que la pointe de l'iceberg : Il reste encore beaucoup de travail et de simulations à faire sur ce sujet.

Le futur est une notion versatile, au combien instable. On ne sait pas ce que ce que le futur réserve une fois ce dernier devenu présent.

C'est donc avec confiance dans les enseignements, la polyvalence et l'esprit de d'adaptation que j'ai reçu en Arts et technologies de l'image et que j'ai su faire mien, que je me dirige vers mon futur, ne sachant ce qu'il me réserve, mais étant près à tout.

GLOSSAIRE:

Camera Obscura: Chambre noire; C'est un appareil optique projetant une image de son environnement sur un écran. Cet appareil fut très utilisé pour le dessin.

George Méliès: 1861-1938, Illusioniste et cinéaste Français.

James Cameron : 1954-, Cinéaste, producteur Canadien.

Voxel : Représente une valeur sur une grille dans un espace 3D.

Steps /substeps : Sous étapes d'une simulation. Nombre de fois qu'une frame sera calculé.

Workflow: Suite de tâches ou action à effectuer par une ou plusieurs personnes.

Maya : Plateforme de création 3D.

Nuke : Logiciel de composition et effets spéciaux.

After Effect : Logiciel de motion design, compositing et effets spéciaux.

Maillage / Mesh : Un objet tridimensionnel composé de polygones.

Displacement map : Texture en niveau de gris permettant de déplacer la géométrie d'un objet 3D en corrélation avec la valeur de la texture à un point donnée. Cela permet d'ajouter du détail sur une modélisation sans augmenter le nombre de faces.

Compositing : Processus consistant à combiner plusieurs éléments visuels en une seule image.

Frame rate/per second (fps): Nombre d'images composant une seconde de film. 25fps : 25 images par secondes : Chaque seconde de ce film / simulation est composé de 25 images / 25 étapes.

Shading : Correspond à l'application de textures et de paramètres physiques sur un modèle 3D, comment la lumière réagit, son niveau de transparence, sa couleur, sa réflexion, etc...

Leidenfrost effect : L'effet Leidenfrost est un phénomène dans lequel un liquide, entrant en contact avec une surface bien plus chaude que sa température d'ébullition, créé une couche de vapeur isolant le liquide et évitant qu'il ne boue trop vite. Une goutte d'eau dans une poêle chauffé à blanc subie cet effet.

BIBLIOGRAPHIE:

LIVRES ET ARTICLES:

Jon P., (2013), *The History of Visual Magic in Computers*, Springer-Verlag London, ISBN 978-1-4471-4932-3 – 2013

Bill D. 'Spider-Man 3': *Breaking Down Sandman*, May 2007, VFXWorld, www.awn.com

Daily Mail Reporter, *Melting men: Thousand ice sculptures left to thaw in the sun to highlight climate change in Arctic*, September 2009, Mail Online, www.dailymail.co.uk

VIDEO ET SUPPORTS DE COURS MULTIMEDIA :

Sunder L., *Scripting in Reelflow*, www.digitaltutors.com

Dan L., *Introduction to Reelflow 2013*, www.digitaltutors.com

William S., *Viscious Liquids in Reelflow and Cinema 4D*, www.digitaltutors.com

Chris G., *Using Morph and Magic Daemons in Reelflow*, www.digitaltutors.com

Yuval K., *Pixelpro videos*, <https://vimeo.com/user595972>

Brandong Y., *Reelflow object Melt Tutorial*, <http://www.byvfx.com/>

NextLimit Technologies, *Reelflow 2013 Documentation*, <http://www.support.nextlimit.com/>

FILMS ET COURT MÉTRAGES :

James C., *Terminator 2 : Judgment Day*, 1991, USA

James C., *Abyss*, 1989, USA

Chris B. & Jennifer L., *Frozen*, 2013, Walt Disney Pictures, USA

Nicolas M. *Star Treck II: The Wrath of Khan*, 1982, Paramount Pictures, USA

TABLE DES ILLUSTRATIONS

Figure 1: Amnesty Commercial, Pleix & Digital District.....	9
Figure 2: Creep, 2011. Photo par Alison Fairley	10
Figure 3: La persistance de la mémoire, Salvador Dali. 1931, peinture à l'huile.	11
Figure 4: Expansion au pot a lait, César, 1989	11
Figure 5 : Melting Men, Berlin.....	12
Figure 6: Extrait du film Spider-Man 3, Colombia pictures & Marvel studio.....	13
Figure 7: Extrait du film Terminator 2 de James Cameron (1991).....	14
Figure 8: Réalisation personnelle, comparaison de densité, 1000 et 4000.	18
Figure 9: Réalisation personnelle, comparaison de viscosité, 3, 100 et 500.	18
Figure 10: Realisation personnelle, comparaison de tension de surface, 0, 100, 1000.	19
Figure 11:: Réalisation personnelle, Gravity Daemon.....	20
Figure 12: Réalisation personnelle, Noise Daemon.	21
Figure 13: : Réalisation personnelle, Attractor Daemon.....	20
Figure 14: Réalisation personnelle, Magic Daemon.....	22
Figure 15: Réalisation personnelle, Comparatif Geometrie.....	23
Figure 16: Réalisation personnelle, Test 02, WaterBall_01.	27
Figure 17: Réalisation personnelle, Test 03, Hybrido_01.....	28
Figure 18: Réalisation personnelle, Test 04, Bubble_01.....	29
Figure 19: Réalisation personnelle, Test 05, Skull_01.	30
Figure 20: Réalisation personnelle, Texture Emission.	31
Figure 21: Réalisation personnelle, Test 08, WetMap.....	32
Figure 22: Réalisation personnelle, Test 10.....	34
Figure 23: Réalisation personnelle, Test X, Head Melt	35
Figure 24: Réalisation personnelle, Test 13, scripting 02.....	37
Figure 25: Projet Intensif de 3 semaines, Plan 040.....	39
Figure 26, Réalisation personnelle, test 09, Icing.	41
Figure 27: Réalisation personnelle, Tutoriel 01	43

Figure 28: Réalisation personnelle, Tutoriel 02.....	44
Figure 29: Réalisation personnelle, Tutoriel 03.....	48
Figure 30: Réalisation personnelle, Tutoriel 04.....	48
Figure 31: Réalisation personnelle, Test 13, Scripting 03.	49
Figure 32 : Realisation personnelle, test 13, scripting 03, Relationship Editor.	51
Figure 33: Réalisation personnelle, Test 13, Scritping 03, Particles view.	52

ANNEXES:

ANNEXE 1 :

```
#-----  
# Function: onSimulationStep  
#-----  
def onSimulationStep():  
    Obj1 = "OBJECT"  
    source = scene.get_PB_Emitter("Solid_PB")  
    target = scene.get_PB_Emitter("Liquid_PB")  
    targetparticlesList = target.getParticles()  
    sourceParticlesList = source.getParticles()  
    for p in sourceParticlesList:  
        colObj = p.getCollidingObject()  
        if colObj != None:  
            Objname = colObj.getName()  
            if Objname == Obj1:  
                target.addParticle(p.getPosition(),p.getVelocity())  
                source.removeParticle(p.getId())  
    for p in sourceParticlesList:  
        p.freeze()  
  
#-----  
# Function: onSimulationFrame  
#-----  
def onSimulationFrame():  
    source = scene.get_PB_Emitter("Solid_PB")  
    sourceID = source.getId()  
    sourceNumber = source.getParameter("Existent Particles")  
    target = scene.get_PB_Emitter("Liquid_PB")  
    targetparticlesList = target.getParticles()  
    sourceParticlesList = source.getParticles()  
    thresholdLevel = scene.getObject("Null01").getParameter("Position").getY()  
    threshold = 45 + thresholdLevel  
  
    if sourceNumber < 250 :  
        for p in sourceParticlesList:  
            target.addParticle(p.getPosition(),p.getVelocity())  
            source.removeParticle(p.getId())  
  
    for Sp in targetparticlesList:  
        targetID = Sp.getEmitterId()  
        Neighbors = Sp.getNeighbors(0.1)  
        numNeighbors = len(Neighbors)  
        if numNeighbors < threshold:  
            for p in Neighbors:  
                emitterID = p.getEmitterId()  
                if emitterID == sourceID :  
                    target.addParticle(p.getPosition(),p.getVelocity())  
                    source.removeParticle(p.getId())
```