Université Paris 8

Master Création Numérique
parcours : Arts et Technologies de l'Image Virtuelle

Plantes virtuelles grimpantes et rampantes

Modélisation procédurale de structures ramifiées sur des surfaces

Fanny Cabé

Mémoire de Master 2, 2018 - 2019

« Va prendre tes leçons dans la nature. »
Léonard de Vinci

Résumé

Les plantes virtuelles grimpantes ou rampantes sont dépendantes du décor sur lequel elles évoluent. Les créer à la main sur d'immenses décors représente une tâche fastidieuse, répétitive et chronophage pour les graphistes. Mon ambition au fil de ce mémoire est donc de comprendre comme créer des plantes grimpantes ou rampantes crédibles sans les modéliser à la main, de manière procédurale, avec l'aide d'algorithmes, tout en conservant un contrôle sur leur aspect esthétique final, dans un contexte de production de film d'animation. Dans ce cadre, différentes méthodes de modélisations seront présentées et mises en application sur des ronces, des racines, des lianes et plusieurs versions d'un outil de génération de plante grimpante.

Summary

Vegetation is very important for the credibility of virtual environments and the presence of climbing or creeping plants helps for it. These plants are dependent of the environment on which they evolve. Creating them by hand is a tedious, repetitive and time-consuming task for 3D artists. My ambition throughout this thesis is to understand how to create credible climbing or creeping plants without modeling them by hand, with the help of algorithms, while maintaining control over their final aesthetic aspect, in a context of production of animation film. In this context, different modeling methods will be presented and applied on brambles, roots, lianas and several versions of a climbing plant tool.

Remerciements

J’adresse mes remerciements chaleureux à tous les membres de l’équipe FX d'Illumination Mac Guff qui m’ont aidée dans la réalisation des projets exposés dans ce mémoire, avec patience et bienveillance. Ensuite, je tiens à remercier pour ses encouragements M. Jean-François Jégo, professeur à l'université de Paris 8 et directeur de mémoire. Je salue également la merveilleuse famille ATI et surtout mes proches qui ont traqué vaillamment les coquilles dans ces pages.

Table des matières

Introduction....1

1 État de l'art....2

1.1 Les végétaux dans l'art....2

1.2 Historique des techniques de modélisation....3

1.2.1 Automates cellulaires....3

1.2.2 L-systèmes et fractales....3

1.2.3 Algorithme de nervation....5

1.2.4 Systèmes de particules....5

1.2.5 Autres méthodes....6

1.3 Les L-systèmes....6

1.3.1 Définition....6

1.3.2 Exemple simple de l'algue....7

1.3.3 Interprétation graphique....7

1.3.4 Améliorations du système....8

1.4 L'algorithme de nervation....9

1.4.1 Principe en 2D....10

1.4.2 Principe en 3D....11

1.5 Les systèmes de particules....12

1.6 Phyllotaxie....12

2 Expérimentations....14

2.1 Premiers projets : mousse et bouquet de fleurs....14

2.1.1 Moss and Flowers....14

2.1.2 Evolve....15

2.2 Racines dorées....15

2.2.1 La méthode du shortest path....15

2.2.1.1 Définition....15

2.2.1.2 Application en 3D....16

2.2.2 Nouvelle mise en place....17

2.2.3 Remarques....18

2.3 Buisson de ronces....19

2.3.1 L'algorithme de nervation....19

2.3.2 L'astuce des volumes imbriqués....20

2.3.3 Finalisation....21

2.4 Lianes au Valhalla....22

2.4.1 Présentation du projet....22

2.4.2 La méthode du système de particules....23
2.4.3 Les UV procéduraux....24
2.4.4 Remarques....25
2.5 Arabesques végétales....27
2.5.1 Descriptif du process....27
2.5.2 Remarques....29
2.6 Générateur de plantes grimpantes....30
2.6.1 Les plantes grimpantes dans la nature....30
2.6.1.1 Observations sur les feuilles....30
2.6.1.2 Observations sur la structure du lierre....31
2.6.2 Outil 1ère version : le problème d'orientation le long d'une branche....32
2.6.2.1 L'algorithme de nervation ?....32
2.6.2.2 Projection sur une surface....33
2.6.2.3 Orientation le long d'une courbe....34
2.6.2.4 Présentation de l'outil à mes superviseurs....35
2.6.3 Outil 2ème version : le décor comme champ de force....36
2.6.3.1 Calcul du champ de croissance....36
2.6.3.2 Premier niveau de branches....37
2.6.3.3 Bruit et décollement....38
2.6.3.4 Un décor bien coiffé....38
2.6.4 Outil 3ème version : retour aux bases et regard extérieur....40
2.6.4.1 Simplification....40
2.6.4.2 Les feuilles....41
2.6.4.3 Critique de l'outil....42
2.6.4.4 L'outil après Valhalla....43
2.6.5 Outil 4ème version : l'approche botanique....45
2.6.5.1 L'outil idéal....45
2.6.5.2 Les forces en présence....46
2.6.5.3 Deux comportements différents....48
2.6.5.4 Embranchements et rayons....50
2.6.5.5 L'interface....51
2.6.5.6 Remarques....53
Conclusion....55
Bibliographie....57
Webographie....58

Introduction

Dans les films d'animation et les jeux vidéo de ces dernières années, la végétation joue un rôle clé dans la crédibilité des environnements virtuels. La présence de plantes grimpantes ou rampantes, qui épousent les formes d'un décor, peut même avoir un fort pouvoir narratif, évoquant la nature qui reprend ses droits.

La modélisation de végétaux virtuels est un sujet largement étudié par les scientifiques, qu'ils soient botanistes, mathématiciens, informaticiens ou même artistes. Aujourd'hui, il est assez simple de modéliser différentes plantes ou arbres de manière indépendante et d'en remplir un décor, en les déplaçant dans l'espace à la guise du directeur artistique.

Les plantes grimpantes et rampantes, quant à elles, sont dépendantes du décor sur lequel elles évoluent car elles ont besoin de ce support pour s'élever vers la lumière. Les créer à la main pour chaque environnement d'un film d'animation représente une tâche fastidieuse, répétitive et chronophage pour les graphistes, bien que cela permette un contrôle artistique absolu sur le résultat.

La modélisation procédurale se présente comme une solution à ce problème. Ici, le terme « procédural » se définit comme l'opposé de « manuel ». Son utilisation implique donc, grâce à des procédures ou des algorithmes, une accélération et, en partie, une automatisation du processus de modélisation.

Mon ambition au fil de ce mémoire est donc de comprendre comme générer des plantes grimpantes ou rampantes crédibles sans les modéliser à la main, avec l'aide d'algorithmes, tout en conservant un contrôle sur leur aspect esthétique final, dans un contexte de production d'un film d'animation.
L'étude de ce sujet ne traitera pas de rendu, de *texturing*, de *shading*, de simulations dynamiques ni de modélisation des feuilles; seule la modélisation de la structure des plantes grimpantes et rampantes sera considérée.

Dans les pages qui suivent, avant de les confronter à mes propres expérimentations, c'est l'ensemble des techniques artistiques et algorithmiques actuelles traitant du sujet qui seront abordées.

1 État de l'art

Les lianes, les plantes grimpantes et les racines font partie de la très grande famille des végétaux et je me suis appliquée à être aussi exhaustive que possible dans la rédaction de cet état de l'art, tant d'un point de vue artistique que technique.

1.1 Les végétaux dans l'art

Les plantes, les fleurs, les paysages, sont des sujets très largement représentés dans l'art [43]. Certaines peintures murales de l'antiquité égyptienne mettent en scène des champs cultivés et des jardins. Les enluminures des livres du Moyen-Âge sont souvent des entrelacs de tiges et de fleurs. Dans la peinture de la renaissance, la végétation est porteuse de symboles forts, comme la pomme et le péché originel. Au XIX^{ème} siècle, le courant Art Nouveau et Alfons Mucha avec lui, recouvrent d'arabesques végétales et de motifs de fleurs la peinture, l'architecture et le mobilier. La nature est également au cœur du courant impressionniste et de tout le travail de Claude Monet. Dans l'art contemporain, le *land art* prend même la nature comme matière première de création. De leur côté, Christa Sommerer et Laurent Mignonneau font pousser des plantes virtuelles en réponse au contact avec de vraies plantes dans leur installation artistique *Interactive Plant Growing* (1992), aujourd'hui dans la collection permanente du ZKM Media Museum, à Karlsruhe en Allemagne.

La représentation de la végétation dans les arts numériques est à l'origine de nombreux défis qui ont été patiemment relevés, puisque l'on est passé en l'espace de 30 ans de la forêt très simple du court-métrage, révolutionnaire à l'époque, *The Adventures of Andre and Wally B*. (1986) (cf Illus. 1) réalisé par the Lucasfilm Computer Graphics Project (qui deviendra Pixar) à la jungle hyper réaliste du film *Jungle Book* de Jon Favreau (2016) (cf Illustration 2).

Illustration 1: Premier plan de TheAdventures of Andre and Wally B (1986)

Illustration 2: Scène extraite du Jungle Book (2016) où seul Mowgli n'est pas virtuel

1.2 Historique des techniques de modélisation

1.2.1 Automates cellulaires

La modélisation des structures de végétaux et donc des structures ramifiées a une histoire relativement longue. Le premier modèle a été proposé par Ulam en 1966 [1][2] : il utilisait le concept d'automate cellulaire, développé la même année par lui-même et von Neumann[3].

Le motif de ramification était créé de manière itérative. On commençait par une cellule de couleur unique dans une grille triangulaire, puis à l'itération suivante, les cellules annexes qui ne touchaient qu'un et un seul sommet d'une cellule colorée étaient à leur tour colorées et ainsi de suite.

En 1989, Greene [4] a étendu ces automates cellulaires 2D aux trois dimensions, pour simuler des processus de croissance réagissant à l'environnement. Par exemple, l'illustration ci-contre présente la croissance d'une vigne sur une maison.

Illustration 3: Organic Architecture par Greene (1989)

La relation entre végétation et environnement est ici très claire. Cependant, prendre en compte ce dernier paramètre complexifie énormément les modèles. Cela peut expliquer pourquoi des modèles plus simples ignorant l'environnement ont été développés par la suite en priorité, laissant de côté la modélisation particulière des plantes grimpantes ou rampantes.

1.2.2 L-systèmes et fractales

En 1968, le biologiste hongrois Aristid Lindenmayer invente les L-systèmes ou systèmes de Lindenmayer. Ils lui servaient initialement à étudier le développement et la prolifération d'organismes multicellulaires simples ; il les a ensuite généralisés aux plantes plus complexes. C'est en 1990 qu'apparaît son application infographique avec la publication de « The algorithmic beauty of plants » [14]. Les auteurs y soulignent d'ailleurs que dans de nombreux processus de croissance de végétaux, la répétition de certaines structures est facilement perceptible. Dans le cas d'un vieil arbre par

Illustration 4: Le chou romanesco présente une structure fractale

exemple, certaines de ses branches ont la même forme que ce même arbre au début de sa vie. Cette répétition se nomme auto-similarité et cette caractéristique est propre aux fractales.

Les fractales sont définies par le mathématicien franco-américain Benoît Mandelbrot en 1974 : il crée ce néologisme « à partir de l'adjectif latin fractus, qui signifie irrégulier ou brisé » [12]. Cet objet mathématique, que ce soit une courbe ou une surface, se caractérise par sa structure similaire quelle que soit l'échelle d'observation. L'auto-similarité est caractérisée par Mandelbrot comme suit: « Lorsque chaque partie d'une forme est géométriquement similaire à l'ensemble, la forme et la cascade qui la génère s'appellent auto-similaires » [12].

Les L-systèmes, qui permettent donc la génération de formes fractales, sont aujourd'hui très répandus pour modéliser les structures de ramification des plantes. On les retrouve dans plusieurs logiciels.

SpeedTree [46] est un ensemble de logiciels de modélisation de végétation qui se base essentiellement sur les L-systèmes. Il est développé par Interactive Data Visualization, Inc. (IDV) et trouve ses applications dans les domaines du cinéma d'animation, de l'architecture et du jeu vidéo. Conçu en 2000, l'édition SpeedTree Cinema est publiée en 2009 et voit sa première utilisation majeure dans le film *Avatar* par James Cameron (2009). Dans ses toutes dernières versions, le logiciel permet d'animer la croissance d'arbres et de plantes, le vent dans les feuilles et les changements saisonniers.

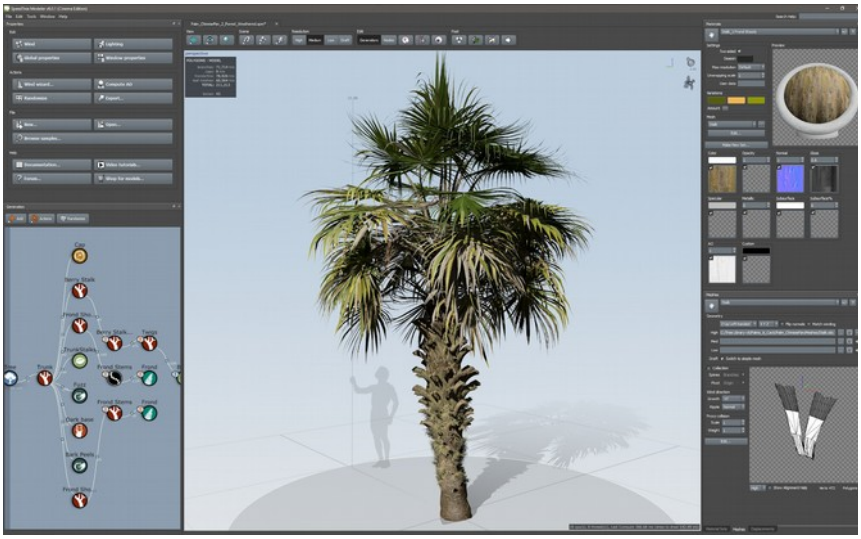


Illustration 5: Interface de Speedtree

Speedtree est très puissant pour créer des arbres pour des plans rapprochés ou à moyenne distance et propose énormément de paramètres, tout en restant très intuitif d'utilisation. Il semble cependant peu fiable s'il doit prendre en compte un environnement particulier, et il faut bien sûr tout reparamétrer si l'environnement change. Speedtree permet de modéliser des arbres mais aussi toute sorte de plantes, dont les plantes grimpantes [18]. C'est majoritairement ce logiciel qui est utilisé à Illumination Mac Guff.

Xfrog [49] est un autre logiciel de modélisation de végétation à base de L-systèmes. Il existe en plugin pour Maya, entre autres, et est aujourd'hui moins utilisé que Speedtree.
Le logiciel Terragen [44] peut générer des paysages numériques magnifiques, par exemple recouverts de forêts, à partir de règles fractales.

1.2.3 Algorithme de nervation

Plus récemment en 2007, Runions et al. [16] parviennent à modéliser la structure d'un arbre en partant d'un algorithme capable de recréer la nervation d'une feuille [15], c'est-à-dire le délicat réseau de veines à sa surface. D'autres scientifiques complètent cette méthode avec des scans de vrais buissons ou arbres [8]. Cela permet de reproduire des plantes de manière très réaliste, mais présente peu d'intérêt d'un point de vue direction artistique.

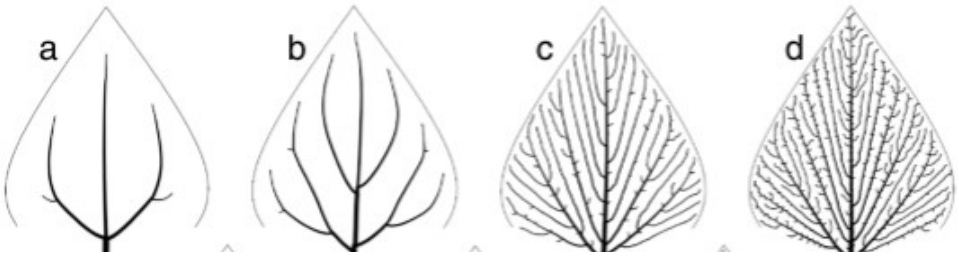


Illustration 6: Différents paramétrages pour l'algorithme de nervation d'une feuille

1.2.4 Systèmes de particules

Pour créer des structures ramifiées, des systèmes de particules peuvent être utilisés, comme l'a fait Thomas Luft pour son générateur de lierre de 2007 [39] .
Chez Disney et depuis le film *Raiponce* (*Tangled* en VO) de Byron Howard et Nathan Greno en 2010, les graphistes utilisent Dendro [17], outil interne de génération de végétation procédurale qui utilise des particules. Une étudiante en thèse Marta Feriani propose aussi une version améliorée de cet outil [6].

Chez Pixar, les paysages fascinants de *Zootopia* (2016, par Byron Howard et Rich Moore) ont été construits avec l'outil interne de modélisation procédurale de végétation « Bonsai » [9], également à base de particules, selon toute vraisemblance. Leur outil est très puissant car il permet de produire différentes espèces d'arbres et de plantes, mais également différents modèles pour une même espèce. Il contient aussi un module qui contrôle la croissance des plantes grimpantes aux arbres ou autres surfaces, mais ils gardent jalousement leurs informations sur les méthodes précisément employées.

1.2.5 Autres méthodes

Les **Paint Effects** [37] sont une fonctionnalité interne à Autodesk Maya,. Ils sont utiles pour répartir des végétaux choisis sur une surface. C'est dans Windows > General Editors > Content Browser ("Visor" dans les versions antérieures à 2017) qu'ils se cachent. Après avoir sélectionné l'outil Paint Brush, on peut choisir un type de plantes et en remplir de manière interactive l'espace 3D, selon une courbe dessinée par l'utilisateur. L'effet est similaire à celui de coups de tampons qu'on applique les uns à la suite des autres.

Il existe aussi des méthodes de modélisation où la réalité physiologique des plantes est prise en compte ; la quantité d'hormones de croissance qui les traverse, l'énergie récupérée par photosynthèse ou l'influence du temps sont calculées. Cette approche se retrouve dans le plugin Blender **The Grove** [31] ainsi que dans le travail de R&D d'Anastasia Opara avec son projet « **Procedural Realistic Tree R&D »** [40]. Ce dernier lui permet notamment de générer des biomes complets, avec les arbres qui poussent les uns par rapport aux autres.

Les végétaux peuvent aussi être modélisés par **photogrammétrie** [8], c'est-à-dire qu'une vraie plante est prise en photo sous tous ses angles pour estimer la position de tous les points de sa surface dans l'espace et en extraire un modèle 3D fidèle.

Anastacio et al. [5] proposent de modéliser les plantes à partir d'une **esquisse** de l'utilisateur, en mélangeant les règles issues de la géométrie des plantes et ce geste humain, potentiellement spontané et imprécis.

Les méthodes de modélisation sont donc multiples mais je vais présenter plus en détail celles qui sont principalement utilisées aujourd'hui, à savoir les L-systèmes, l'algorithme de nervation et les systèmes de particules [10].

1.3 Les L-systèmes

1.3.1 Définition

Le L-système est un système de grammaire formelle, c'est-à-dire un système de réécriture itérative de chaînes de caractères, qui permet de créer des motifs complexes et des structures de ramification de qualité fractale. Un L-système est donc composé :

- d'un alphabet, soit l'ensemble des symboles pouvant être utilisés pour écrire des chaînes de caractères,

- de règles de production, qui décrivent comment produire une nouvelle chaîne de caractère ou une chaîne plus grande à partir de la chaîne précédente,
- et d'un axiome, qui est utilisé comme point de départ pour la production, état initial du système ou encore « graine ».

Prenons l'exemple simple du système original de Lindenmayer pour modéliser la croissance d'une algue.

1.3.2 Exemple simple de l'algue

Les cellules de cette algue sont représentées par les symboles A et B. À chaque nouvelle génération, les cellules se divisent, ce qui se traduit par le remplacement récursif d'un symbole par un ou plusieurs autres symboles consécutifs, permettant la croissance de la structure.

Alphabet: A B
Axiome de départ: A
Règles de production:

- (A → B), A devient B
- (B → AB) B devient AB

Ce L-système produit la chaîne de caractères suivante, n étant le nombre d'itération :
n = 0 : A
n = 1 : B
n = 2 : AB
n = 3 : BAB
n = 4 : ABBAB
n = 5 : BABABBAB
n = 6 : ABBABBABABBAB
n = 7 : BABABBABABBABBABABBAB

Remarque : si on compte la longueur de chaque chaîne, on obtient la célèbre séquence de Fibonacci :1 1 2 3 5 8 13 21 34 55 89… Notons que les quotients de deux termes consécutifs de la suite de Fibonacci Fk_{+1} / Fk sont les meilleures approximations du nombre d'or [14].

1.3.3 Interprétation graphique

La chaîne de caractères résultante d'un L-système peut être interprétée graphiquement, en 2D comme en 3D. En 2D, il suffit d'imaginer une main qui déplace un stylo sur une feuille en suivant des instructions, comme « trace un trait tout droit », puis « tourne à droite de 15 degrés »… Pour

traduire ces instructions, on introduit de nouveaux symboles qui font partie de la « *Turtle interpretation* », du nom de la « tortue » du langage de programmation Logo qui fonctionne sur le même principe. En fait, on peut visualiser la tortue comme étant la main qui tient le stylo.

Pour mettre en pratique, prenons un exemple un peu plus complexe pour modéliser des plantes fractales :
Axiome : F
Règle de production : F→F[+F]F[-F]F
Angle de ramification : 25,7°

En langage Turtle, F signifie "dessine un trait en avant", - signifie "tourner à gauche de 25 °" et + "tourner à droite de 25,7°". Le crochet "[" correspond à la sauvegarde des valeurs courantes de position et d'angle, qui sont restaurées lors de l'exécution du "]" correspondant.

Cet exemple est illustré en (a) ci-dessous, au stade de la 5ème génération.

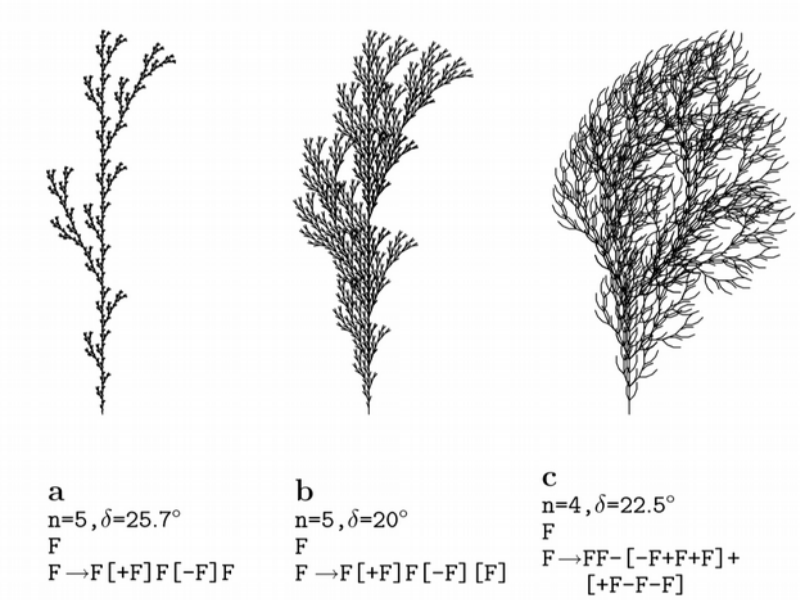


Illustration 7: Résultat graphique de L-systèmes de plantes fractales

1.3.4 Améliorations du système

Ces exemples sont des D0L-Systèmes. C'est la classe la plus simple du L-système, avec D pour déterministe (il offrira toujours le même résultat) et 0 (zéro) pour signifier « indépendant du contexte ».
Ces systèmes produisent des résultats graphiquement satisfaisants, mais la répétition du motif reste très visible et ne donne pas un effet naturel.

Prusinkiewicz [14] introduit alors le S0L-système, avec S pour stochastique, c'est-à-dire que des probabilités entrent en jeu. Il y a un choix, pondéré par une probabilité, entre plusieurs règles de productions différentes pour une même génération. Il introduit aussi des L-systèmes sensibles au contexte, c'est-à-dire que dans une chaîne de caractère, ce qui précède ou ce qui suit le symbole à remplacer est pris en compte. Les L-systèmes paramétriques permettent aussi de faire varier, par exemple, la longueur de l'internœud (c'est-à-dire la longueur d'un segment unitaire). Ces différents ajouts donnent un aspect plus crédible aux plantes générées.
Beaucoup d'autres ont travaillé à l'amélioration de ce système, notamment Palubicki et al. [13] en 2009, qui mélangent L-systèmes et algorithme de nervation (décrit au paragraphe suivant) pour produire des arbres virtuels très convaincants.

Illustration 8: Exemples d'arbres créés en modifiant un seul paramètre dans le système de Palubicki

Les L-systèmes ont cependant été très critiqués car, comme beaucoup de systèmes récursifs, ajouter une toute petite modification dans les règles impacte l'intégralité de la structure du végétal, rendant le contrôle artistique de l'ensemble difficile, mais pas impossible comme le démontre Palubicki [13] dans son travail.

Par ailleurs, réussir à déterminer des règles de productions, qui généreront la structure de la plante qu'on imagine, peut être difficile voire totalement contre-intuitif.

1.4 L'algorithme de nervation

Cet algorithme plus récent et très à la mode parmi la communauté scientifique [42][7][15][16][28] pour modéliser la structure des végétaux est l'algorithme de nervation (*venation* en anglais) aussi appelé algorithme de colonisation de l'espace (*space colonization algorithm*).

Cet algorithme est tiré des travaux de Runions et al. [15] et permet de reproduire la nervation des feuilles, c'est-à-dire le fin réseau de veines présent sur la surface des feuilles de beaucoup de végétaux.

Cette méthode qui s'applique d'abord en 2D a été extrapolée en 3D pour générer des arbres et permet notamment à l'utilisateur de contrôler leur forme en contraignant l'espace dans lequel ils poussent [16].

1.4.1 Principe en 2D

Le principe de l'algorithme peut être caractérisé en 5 grandes étapes. Il existe sur internet des explications animées de ce process pour mieux comprendre [48][28].

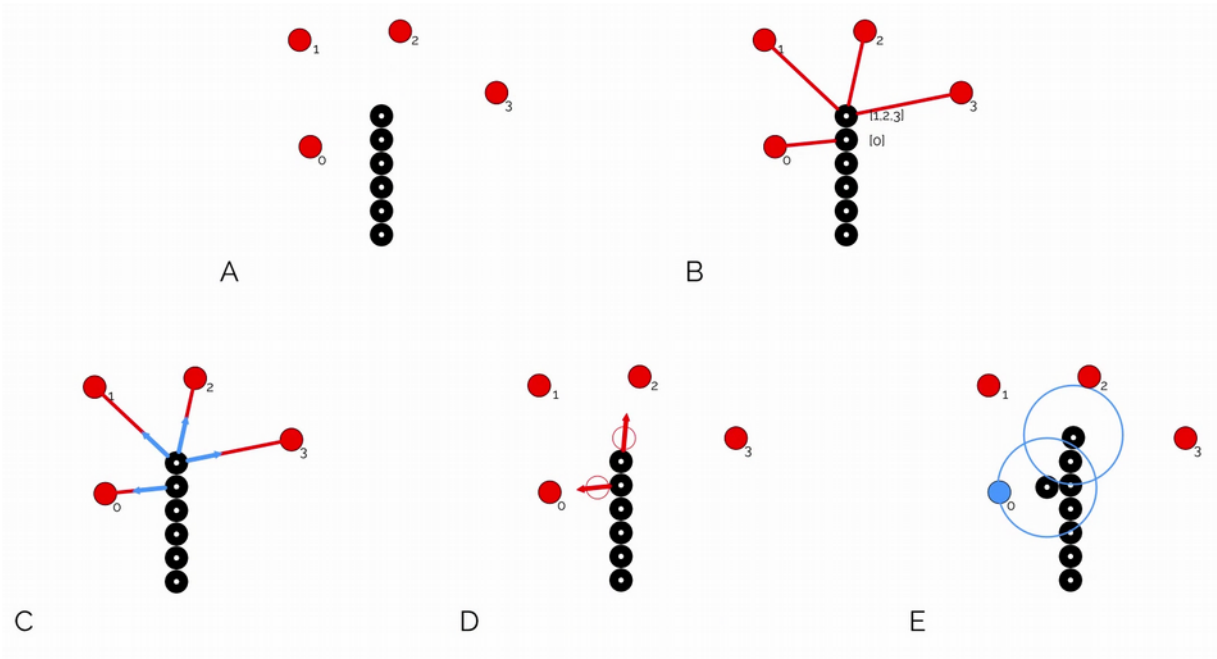


Illustration 9: Schéma explicatif des 5 étapes de l'algorithme de nervation

Étape A : Initialisation

Des points sont répartis sur la surface à remplir. Appelons-les points "attracteurs". Dans le schéma ci-dessus, ce sont les points rouges numérotés de 0 à 3 et ils représentent les directions où la structure est autorisée à grandir.

Un ou plusieurs autres points sont placés sur cette même surface et sont appelés « racines ». Ces points représentent la branche de base d'où vont ensuite partir toutes les ramifications. Dans le schéma ci-dessus, ce sont les points noirs.

Étape B : Recherche de proximité

Chaque point attracteur (en rouge) cherche quel est le point racine (en noir) le plus proche. Chaque point noir enregistre ensuite quels points rouges le considèrent comme le plus proche. La suite de l'algorithme ne s'applique ensuite qu'aux points racines qui sont associés à des points attracteurs.

Par exemple sur le schéma, le plus haut point noir est proche de 3 points rouges et le deuxième point noir est proche d'un seul point rouge. Les autres points noirs ne sont associés à aucun attracteur, donc ils restent « dormants ».

Étape C : Calcul de la direction moyenne

On considère donc les points noirs qui ont été associés à des attracteurs. Dans notre schéma, il y en a deux. Pour chaque point noir, on calcule la direction entre lui-même et chaque point attracteur rouge associé. Puis, on fait une moyenne normalisée de toutes les directions pour chaque point racine noir.
Ici, pour le 2ème point noir, c'est simple, la direction moyenne est uniquement la direction normalisée entre le point noir et son unique point rouge associé.
Pour le premier point noir en revanche, on calcule la direction entre le point noir et chaque point rouge et on normalise leur somme.

Étape D : Création de nouveaux points

Pour chaque point racine qui a été associé, un nouveau point racine est créé juste à côté, dans la direction moyenne calculée et à une distance prédéterminée.

Étape E : Suppression des points attracteurs trop proches

Chaque nouveau point racine généré cherche les points attracteurs les plus proches, dans un rayon prédéterminé. Les points rouges trouvés dans ce rayon sont supprimés. Dans notre schéma, le point bleu est trop proche d'un des derniers points racines créés et va donc être supprimé.

L'algorithme se termine ici et reboucle à partir de **l'étape B**, du nombre d'itération désiré pour créer la structure ramifiée. L'illustration 10 ci-dessous permet peut-être de mieux comprendre.

1.4.2 Principe en 3D

En 3D et dans l'objectif de créer un arbre, un volume est initialement défini et les points attracteurs sont répartis à l'intérieur (les points bleus sur le cas *a* de l'illustration 10 ci-dessous). La graine de départ est placée en dessous de ce volume et la forme de l'arbre apparaît alors au fur et à mesure des itérations de l'algorithme (b et c).

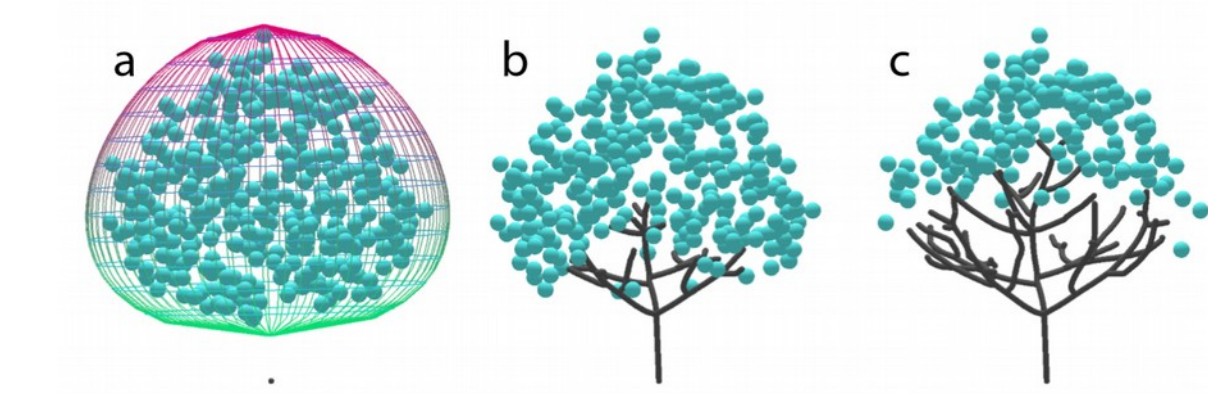


Illustration 10 : L'algorithme de nervation en 3D

La forme générale de l'arbre peut donc être contrôlée en modifiant le volume de départ.

1.5 Les systèmes de particules

Un système de particules est une technique employée dans les logiciels graphiques pour simuler, à l'aide d'un grand nombre de petits éléments, certains phénomènes naturels comme le feu, l'eau, la fumée… et sur lesquels vont s'appliquer des forces telles que la gravité, le vent etc.

Un système de particules est habituellement composé :

- d'un émetteur de particules,
- d'un module permettant de prendre en compte d'éventuels obstacles,
- de forces dans une direction donnée,
- et d'un *solver* qui va donc calculer en fonction de ces éléments et à chaque image (ou « frames ») les nouvelles positions dans le temps et dans l'espace des particules considérées.

Pour créer des structures qui poussent, on peut appliquer des forces spécifiques aux particules, comme la gravité ou l'influence du soleil et les regarder évoluer. En fusionnant chaque position de particules avec sa position précédente, on peut créer des chemins, des « branches ». A certains endroits, une particule peut en générer une deuxième qui deviendra à son tour la base d'une nouvelle branche [30].

1.6 Phyllotaxie

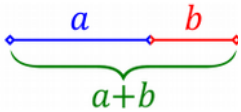


Illustration 11: Schéma de la divine proportion ou nombre d'or

Dans la nature, il est important de noter la disposition étonnamment régulière des fleurs, feuilles, branches le long d'une tige centrale ; cette disposition est appelée **phyllotaxie** [14] . Elle est intimement liée au **nombre d'or,** aussi nommé proportion dorée ou divine proportion. On retrouve aussi l'**angle d'or**

entre la pousse de deux branches. Ces notions sont très importantes pour modéliser des plantes virtuelles crédibles.

En mathématiques, deux quantités sont dans la **divine proportion** (ou nombre d'or) si leur rapport est le même que le rapport entre leur somme et la plus grande des deux quantités, c'est-à-dire :

$$\frac{a+b}{a} = \frac{a}{b}.$$

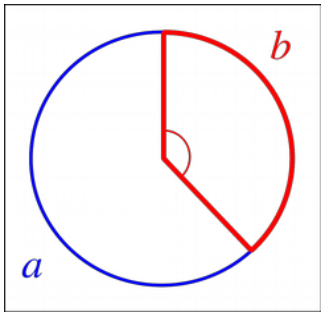


Illustration 12: l'angle d'or

Ce nombre d'or est communément désigné par la lettre grecque φ et vaut $\frac{1+\sqrt{5}}{2}$ soit 1.61803398875 …

L'**angle d'or** est l'angle sous-tendu par l'arc plus petit (en rouge sur l'illustration ci-contre) lorsque deux arcs formant un cercle sont dans la divine proportion. Il vaut en degrés environ **137,5°**.

Illustration 13: Cœur d'une fleur de tournesol

On retrouve notamment cet angle dans la disposition des graines (ou fleurons) dans le cœur d'une fleur de tournesol (ci-contre).

Par extension, on peut dessiner une **spirale d'or**, qui est une spirale logarithmique dont le facteur de croissance est le nombre d'or. C'est-à-dire qu'elle s'élargit (ou s'éloigne de son origine) d'un facteur φ pour chaque quart de tour qu'elle effectue. Son équation polaire est $r = ae^{b\theta}$ avec *a* une constante arbitraire, *e* comme base des logarithmes naturels, et *b* défini comme suit : $b = \frac{\ln \varphi}{\pi/2}$, φ le nombre d'or.

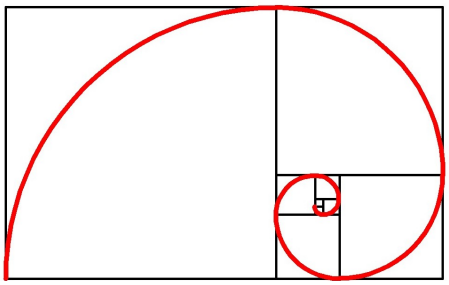


Illustration 14: Spirale d'or

On retrouve cette spirale dans les vrilles de certaines plantes grimpantes [47].

C'est donc à la lumière de toutes ces connaissances que mes propres tests ont pu débuter.

2 Expérimentations

Au cours de cette année, de nombreuses occasions d'explorer la végétation rampante procédurale se sont présentées. D'un côté, mes projets personnels ont permis de mettre en pratique certaines techniques et d'en étudier l'aspect artistique. De l'autre, la création d'un outil de plante grimpante, dans le cadre de mon alternance à Illumination Mac Guff, a mis en lumière de nombreux problèmes techniques.

L'intégralité de mes travaux ont été réalisés sur le logiciel SideFX Houdini en version 16, 16.5 et 17. Tous les fichiers sont consultables sur le DVD joint si besoin.

Ce mémoire ne présente pas le descriptif technique précis de mes projets dans ce logiciel ; ce serait trop indigeste à la lecture. C'est avant tout un partage de toutes mes réflexions et découvertes sur le sujet, avec dans l'idée qu'elles puissent être reproduites dans un autre logiciel.

2.1 Premiers projets : mousse et bouquet de fleurs

2.1.1 Moss and Flowers

Mon tout premier projet Houdini, intitulé Moss and Flowers (« mousse et fleurs ») a été réalisé avec Florentin Luce en tout début de M1 (novembre 2017). C'était aussi ma première confrontation avec la végétation procédurale, notamment rampante, bien que non ramifiée.

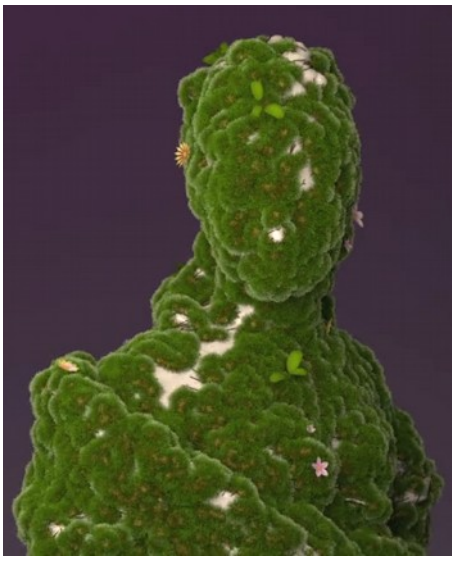


Illustration 15: Image de fin du projet Moss and Flowers (2017)

La mousse est ici élaborée avec les outils de fourrure de Houdini : des poils verts et légèrement marrons à la base recouvrent des sphères déformées qui grossissent et se propagent progressivement sur le buste. Le résultat est convaincant sur une image fixe, l'animation présentait cependant des problèmes que nous n'avons pas eu le temps de résoudre.

La mousse que l'on trouve dans la nature est évidement bien plus complexe et ne ressemble pas à des poils, mais cette approximation reste possible tant que nous ne faisons pas de gros plans.

2.1.2 Evolve

Le projet intensif de M1 « Evolve », réalisé en étroite collaboration avec Isabela Stoppa Guerreiro, a été ma deuxième confrontation à la végétation procédurale.

Les structures ramifiées étant absente de ce projet, ce dernier ne sera donc pas commenté. Il nous a cependant permis de comprendre la puissance d'harmonie de la phyllotaxie : le bouquet virtuel de fleurs et de rameaux, que nous avons créé et animé, n'est en réalité qu'une suite de copies aléatoires de pétales ou de feuilles le long de courbes, respectant l'angle d'or de 137,5°. Tous mes projets suivants l'ont donc utilisé.

Illustration 16: Le projet Evolve (2018)

2.2 Racines dorées

C'est aussi dans le projet Moss and Flowers que nous avons découvert la méthode du *shortest path* ou chemin le plus court, qui nous a permis de créer des racines. Ce sont elles qui entraînent la pousse de la mousse sur le buste.

2.2.1 La méthode du shortest path

Pour notre projet, cette méthode du « chemin le plus court » nous a permis de créer ce réseau de racines qui ne se chevauchent pas. Étonnamment, je n'ai pas trouvé trace de cette méthode dans la littérature scientifique liée à la végétation virtuelle mais seulement dans des tutoriels [26]. Le principe est expliqué ci-après.

2.2.1.1 Définition

En théorie des graphes, le problème du plus court chemin (ou *shortest path*) souligne la difficulté de trouver un chemin entre deux sommets (ou nœuds) dans un graphe de manière à minimiser la

somme des poids (ou 'coûts' ou longueurs) de ses arêtes constituantes. Le plus court chemin entre deux sommets est celui qui offre la plus petite somme des valeurs des arêtes traversées.

Sur le graphe pondéré ci-contre, le chemin le plus court entre les sommets A et F est A, C, E, D, F car la somme des poids des arêtes y est inférieur au chemin A, B, D, F qui est un chemin qui « coûte plus cher ».

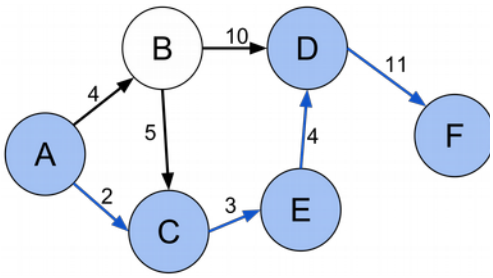


Illustration 17: Graphe pondéré

Les algorithmes qui permettent de trouver une réponse à ce problème sont l'algorithme de Dijkstra, découvert en 1959, et son extension l'algorithme A* (prononcer A étoile ou A star à l'anglaise) qui est présent dans la plupart des logiciels aujourd'hui, donc dans le logiciel utilisé pour mes projets, SideFX Houdini.

2.2.1.2 Application en 3D

Cet algorithme fonctionne donc sur un graphe composé de sommets et d'arêtes. On peut donc faire correspondre ce concept à une géométrie polygonale composée de sommets de polygones et d'arêtes de polygones. L'algorithme va donc générer, dans ce cas, une unique courbe brisée qui relie exactement le ou les point(s) de départ avec le ou les point(s) d'arrivée les plus proches en passant par les arêtes des polygones. On peut aussi assigner à certaines arêtes un « coût » ou poids plus important : si un point de départ est relié à un point d'arrivée par une arrête qui a un coût très élevé, l'algorithme préférera relier les deux points par un plus long chemin mais dont les arêtes ont un coût total plus faible. S'il y a un unique point de départ et beaucoup de points d'arrivée, un réseau va se construire, avec des courbes qui partent toutes du même endroit pour atteindre différents points. Certaines courbes vont potentiellement utiliser les mêmes chemins au début pour se séparer ensuite, ce qui va créer un effet de ramification.

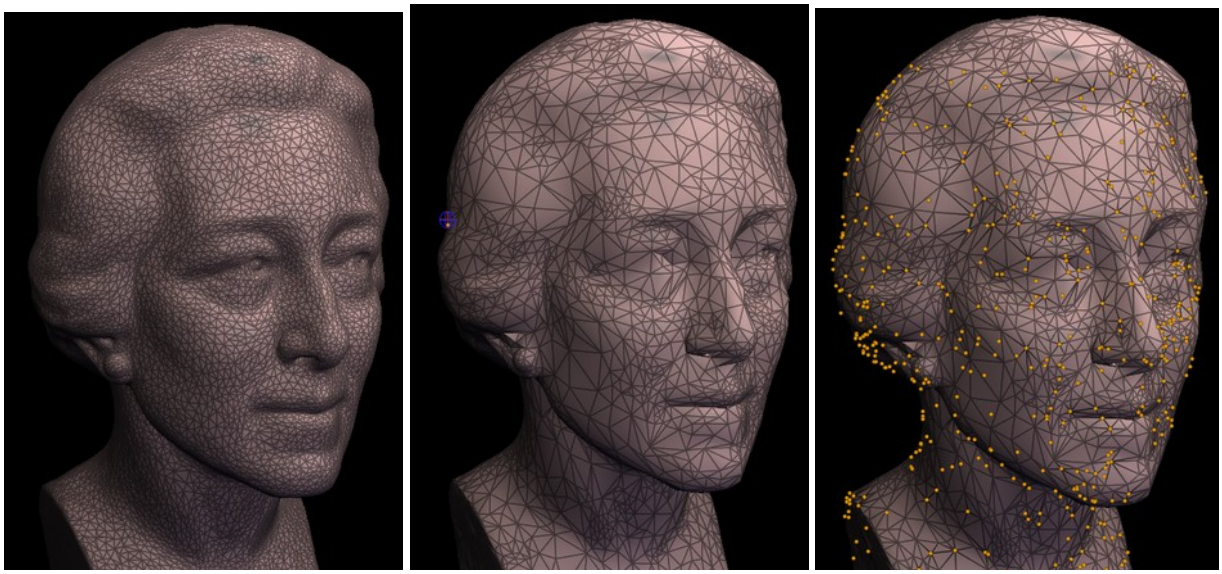
Illustration 18: Image du début du projet Moss and flowers (2017)

Dans notre projet Moss and Flowers et comme on peut le voir sur l'image ci-contre, nous n'avions pas utilisé la pleine puissance de cet algorithme : les arêtes du modèle 3D du buste n'ont pas de coûts particuliers, produisant des branches toutes droites. Par ailleurs, ces racines ont des rayons peu crédibles et leur aspect est beaucoup trop anguleux pour être organique.

2.2.2 Nouvelle mise en place

Une deuxième chance a donc été donnée à cette méthode, toujours à partir du scan 3D d'un buste de femme. Ma méthodologie est expliquée dans les lignes qui suivent.

On commence par réduire le nombre de polygones en laissant un bruit (dans le sens informatique du terme) influencer cette réduction pour obtenir un maillage non homogène, irrégulier, qui nous donnera ainsi un réseau de racines un peu plus organique. Sur ce nouveau maillage, l'unique point de départ (cf l'illustration ci-dessous, au milieu) et les points d'arrivée (ci-dessous, à droite) sont sélectionnés, plus ou moins aléatoirement.



llustration 19: De gauche à droite : le buste de base, le buste avec réduction aléatoire de polygones et sélection du point de départ, et enfin le même buste avec sélection des points d'arrivée

On applique aussi un coût aléatoire aux arêtes des polygones du buste. Toutes ces informations sont entrées dans l'algorithme de *shortest path* d'Houdini. Le point de départ sélectionné est alors relié à chaque point d'arrivée (illustration 20 ci-dessous, gauche) et les courbes créées sont lissées pour casser l'effet anguleux donné par les polygones.

Un rayon est attribué à chaque point des courbes : plus gros si le point est proche du point de départ et plus petit si le point est proche du point d'arrivée. Chaque point est ensuite repositionné exactement sur la surface du buste initial (non déformé par la réduction des polygones et qui conserve donc ses détails). Il est ensuite décalé selon la normale (ou direction orthogonale à la surface) de la géométrie pour créer des racines élégamment décollées (cf. illustration ci-après, milieu).

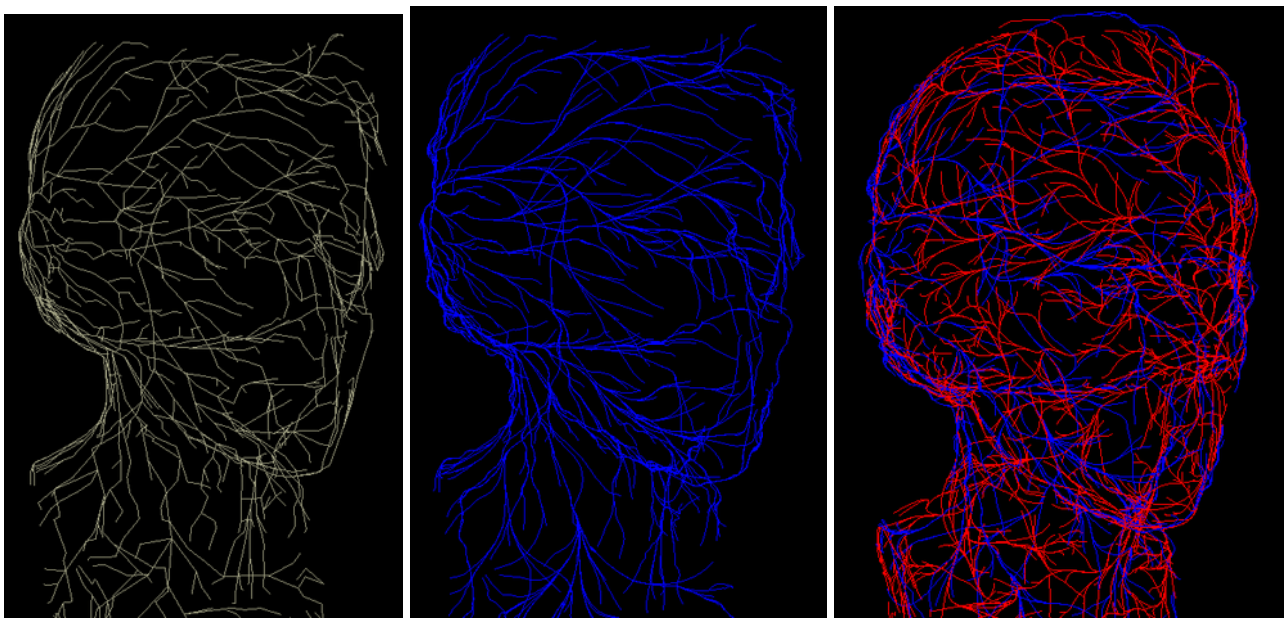


Illustration 21: De gauche à droite : les courbes en sortie du shortest path, puis les mêmes lissées et légèrement décollées et enfin les mêmes recouvertes d'un deuxième système de courbes

Pour augmenter le nombre de racines, un deuxième système est créé avec un point de départ et des points d'arrivée différents (en rouge sur l'illustration ci-dessus, à droite).

Pour le rendu, les courbes ne sont pas converties en tubes de polygones afin de garder une scène légère, sachant que Mantra (le moteur de rendu d'Houdini) est capable d'interpréter correctement leurs épaisseurs.

L'animation résultante de cette création de racines est présentée ci-contre.

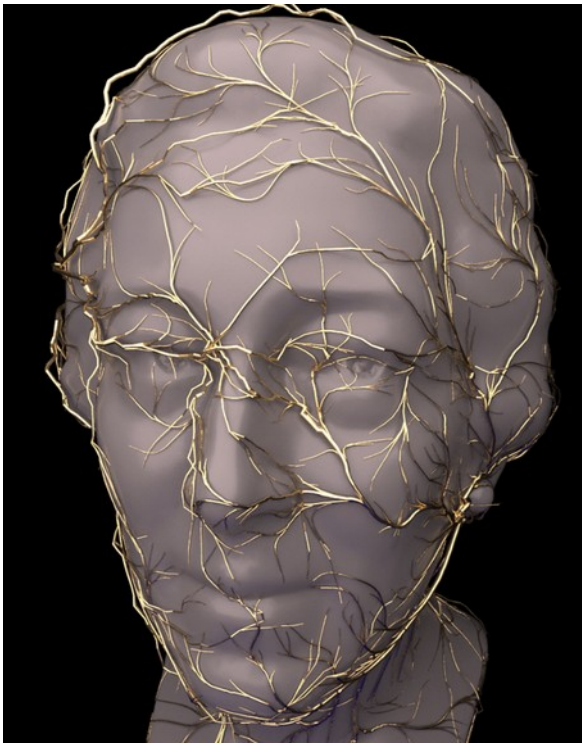


Illustration 20: le résultat rendu (et animé)

2.2.3 Remarques

On peut voir avec cette méthode une certaine similitude avec le comportement des racines dans la nature : les points de départ sélectionnés peuvent correspondre aux toutes premières pousses de racines, les points d'arrivée sont les points d'eau, et les arêtes avec des coûts élevés sont des zones avec de gros cailloux qui bloquent le passage.

Finalement, le rendu est satisfaisant, organique et fluide mais assez compliqué à manier si la direction artistique est précise. Il est en effet difficile de modifier l'aspect d'une racine en

particulier, à moins de changer à la main la courbe générée par l'algorithme. Par ailleurs, si la modélisation change d'un pouce, il est très probable que le nouveau réseau de racines résultant n'aura pas grand-chose à voir avec le précédent, à cause de l'utilisation du bruit aléatoire.

2.3 Buisson de ronces

Au cours de mon stage à Illumination Mac Guff pendant l'été 2018, ma principale mission a été de construire un buisson de ronces, avec plusieurs contraintes :

- il devait être modélisé procéduralement, afin de pouvoir être modifié facilement si besoin et d'éviter la lourdeur d'une modélisation manuelle,
- il devait rendre accessible les lignes constituant la structure des ronces, pour en faire des simulations dynamiques si un personnage s'y accroche,
- il ne devait pas y avoir d'interpénétrations entre les différentes branches de ronces.

2.3.1 L'algorithme de nervation

Après recherches, l'utilisation de l'algorithme de nervation décrit dans la première partie de ce mémoire, s'est vite imposée.

Pour commencer, un grand nombre de points attracteurs sont répartis à l'intérieur d'un volume grossier délimitant l'espace que doit occuper le buisson. Une poignée de points racines sont également posés à différents endroits de la base de ce volume. Avec ces éléments, l'algorithme de nervation est lancé, un rayon progressif est appliqué aux différentes branches et des épines sont dispersées dessus selon l'angle d'or. On obtient alors une première version du buisson de ronces présentée dans l'illustration ci-dessous :

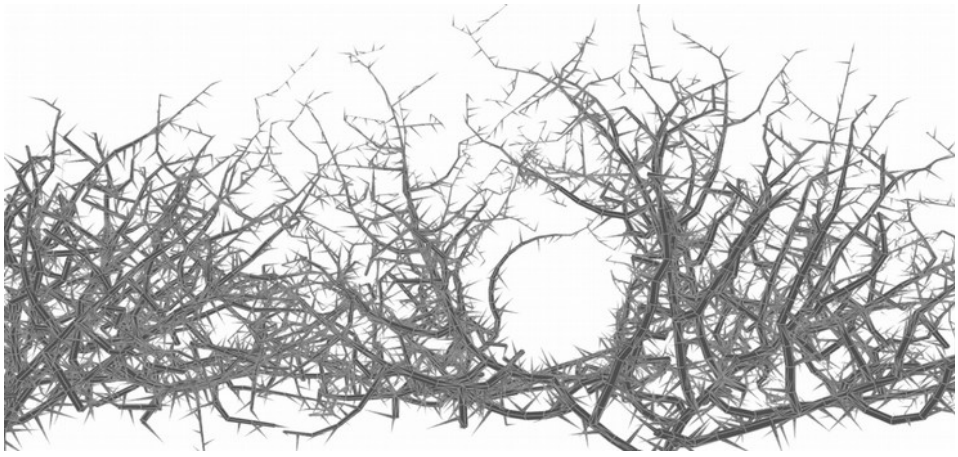


Illustration 22: Toute première version du buisson de ronces

Voilà, objectivement semble-t-il, un buisson, mais pas un buisson de ronces comme sur la photo ci-dessous. Dans la nature, les ronces poussent selon une unique tige qui monte vers le ciel puis retombe vers le sol. Là, cette tige se plante et se développe ensuite en une ou plusieurs autres tiges, envahissant très vite un espace. De loin, on constate qu'elles créent comme un amas d'arcs de cercle. Dans Houdini, on peut tout à fait remplir un volume avec des arcs de cercle mais on se confronte très vite à des problèmes de collisions entre chaque tige. Comment éviter cette impression de « sac de nœuds »?

Illustration 23: Photo d'un buisson de ronces

2.3.2 L'astuce des volumes imbriqués

La solution que l'on m'a suggérée est de partir de plusieurs volumes imbriqués les uns dans les autres, comme des poupées russes, et de faire courir sur leur surface l'algorithme de nervation en 2D (et non en 3D comme avant), ce qui empêcherait tout chevauchement. Ainsi, comme les volumes ne se touchent pas par ailleurs, cette méthode évite strictement toute collision entre les tiges tout en donnant cet effet de végétation qui fait des arcs de cercle, naturellement donné par les volumes emboîtés comme présenté ci-dessous à gauche dans l'illustration 24.

Concrètement, pour obtenir ces volumes emboîtés, un premier volume très grossier (par exemple une sphère aplatie) aux dimensions inférieures à celles du buisson désiré, est défini. Ensuite, on lui applique une force (avec des variations aléatoires) qui le fait grossir principalement vers le haut. Pour finir, les différentes « étapes » de cette transformation sont récupérées pour obtenir nos volumes imbriqués, 6 dans notre cas (cf. illustration 24 gauche).
Les points attracteurs et racines sont répartis sur chaque surface des volumes et l'algorithme génère les tiges des ronces, comme présenté sur l'illustration ci-après, au milieu et à droite :

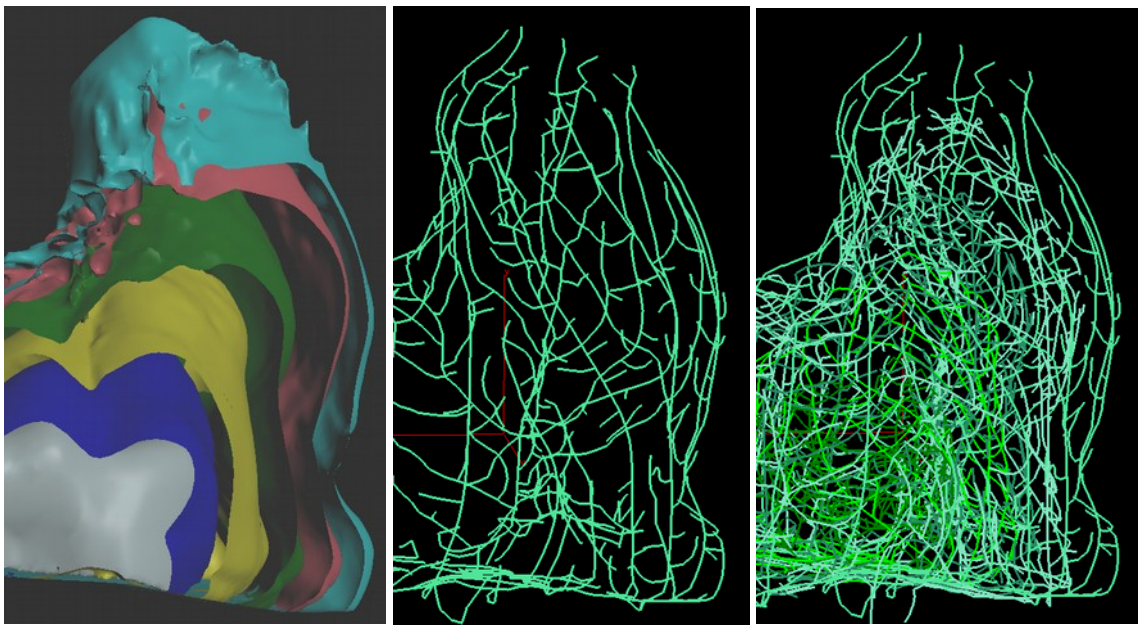


Illustration 24: A gauche : les volumes imbriqués qui servent de support à l'algorithme de nervation. Au milieu : Le plus grand volume recouvert de ronces produites par l'algorithme. A droite : tous les volumes de ronces imbriqués.

2.3.3 Finalisation

Pour accentuer l'effet ronces, des arcs de cercles créés à partir de spirales d'or sont répartis aléatoirement sur les deux derniers niveaux de ronces et orientés vers l'extérieur (illustration ci-contre). Étant peu nombreux, ils ont peu de chance d'entrer en collision les uns avec les autres.

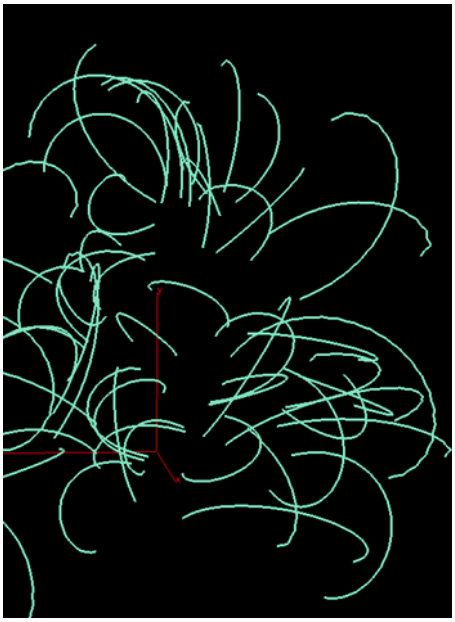


Illustration 25: Répartition aléatoire d'arcs de cercles sur les derniers volumes de ronces

La spirale devait être « légère » dans le sens où il ne fallait pas que le petit tourbillon central apparaisse au risque d'avoir des ronces un peu trop graphiques ou d'un style « conte de fées » (et pas du tout réaliste, accessoirement).

Ces arcs et les ronces précédemment obtenus sont ensuite fusionnés. Des épines, dont on peut paramétrer la quantité, sont réparties sur l'ensemble pour produire le buisson ci-dessous.

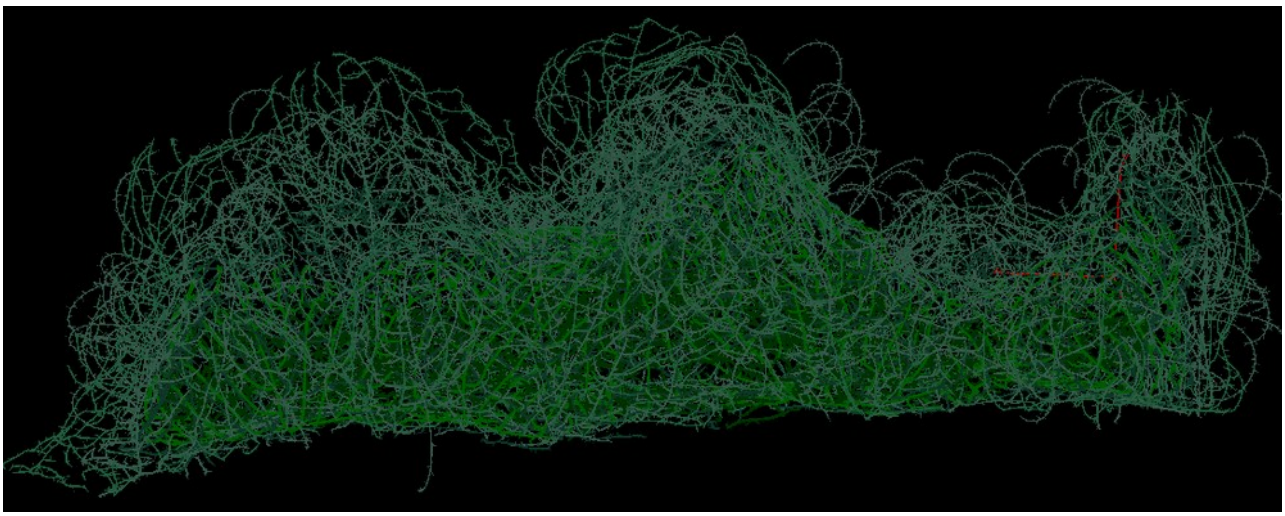


Illustration 26: Modélisation procédurale quasi finale du buisson de ronces

A ce moment-là de la création du buisson de ronces, mon stage a pris fin. Un autre graphiste a donc pris la suite pour insérer des feuilles, un autre a fait des simulations dynamiques dessus, puis un autre département s'est chargé du shading et encore un autre du rendu. Cette modélisation de base a en tout cas été utilisée pour une séquence d'un long-métrage d'animation. A l'heure où ce mémoire est rédigé, le film n'est pas sorti donc il m'est interdit de montrer le résultat final.

Notons que les ronces ne sont pas ce qu'on pourrait qualifier de végétation rampante (mais plutôt envahissante), cependant l'utilisation du très intéressant algorithme de nervation m'a amenée à l'inclure dans ce mémoire.

Le contrôle artistique était ici faisable, mais non procédural : la forme générale des volumes a été modifiée avec un *lattice* (un déformeur particulier de géométrie) pour obtenir l'aspect désiré par le directeur artistique.

2.4 Lianes au Valhalla

2.4.1 Présentation du projet

Le projet « Valhalla » a été réalisé en collaboration avec Isabela Stoppa Guerreiro, Akriti Bacha et Vincent Noé. Il s'agit d'un clip sur une musique Viking, rendu avec le moteur temps réel Unreal Engine et entièrement modélisé dans Houdini. L'univers imaginaire dans lequel nous évoluons est composé d'une suite de cavernes aux multiples tunnels, dont les surfaces sont recouvertes de lignes-lianes qui s'entremêlent et se superposent. Différentes matières leurs sont attribuées pour donner différentes ambiances, au fil de la traversée des 6 cavernes différentes générées procéduralement.

Pour ce projet, l'utilisation de mon outil de plante grimpante (en version 3 au moment du projet, voir paragraphes suivants) était prévue mais il s'est révélé très vite totalement inadapté. En effet, mon outil a besoin pour fonctionner du tracé préalable, par l'utilisateur, d'une courbe directionnelle, matérialisant le tronc principal de la plante grimpante. Cette courbe ne doit pas traverser le support, donc il faut faire attention quand on la dessine, et ce support nécessite d'être plutôt plat. Ainsi donc, tracer des courbes manuellement dans toutes les cavernes gigantesques aux larges surfaces biscornues aurait pris bien trop de temps et n'était pas envisageable. Les techniques de systèmes de particules ont donc été mises en oeuvre.

Ici, la création des lianes est identique pour chaque caverne, seuls quelques paramètres changent d'une caverne à l'autre. Mes connaissances des particules proviennent essentiellement de la mine d'or qu'est Cgwiki [30] et du tutoriel « Houdini Particle flow on surface » [36].

2.4.2 La méthode du système de particules

Voici le déroulé de la création des lianes d'une caverne, tel que réalisé au cours de ce projet.

D'abord la caverne est construite procéduralement (illustration ci-contre). Ensuite des points sont positionnés sur sa surface, avec une densité plus ou moins forte.

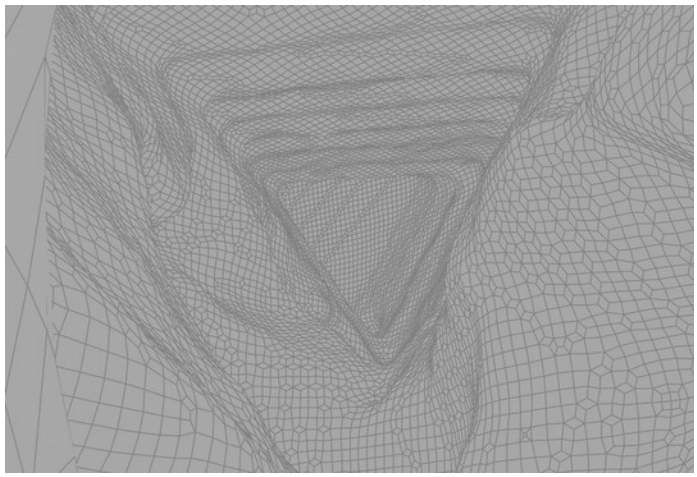


Illustration 27: Couloir d'une caverne générée procéduralement

Dans le système de particules utilisé, l'émetteur est paramétré de sorte que chaque point réparti sur les parois génère une unique particule. Une force (ou vent) dans une direction donnée est appliquée aux particules pour les faire avancer. Elles sont ensuite recollées à chaque image sur la surface de la caverne. La position de chaque particule est ensuite fusionnée avec sa position précédente, donnant ainsi de longues lignes qui parcourent la surface.

En jouant avec les paramètres du vent et notamment le contrôle du bruit (*noise*), on peut obtenir différents aspects de lignes-lianes, par exemple en leur faisant suivre un axe, ou bien un ensemble précis de courbes (comme sur l'illustration ci-dessous, les lianes suivent la courbure du tunnel), ou la rotation autour d'un axe, ou sous forme de petites boucles ou encore de manière totalement aléatoire.

Les longues lianes élaborées à partir des particules (cf. ci-dessous) sont ensuite « nettoyées » : les points trop proches sont fusionnés, les courbes sont lissées et les plus petites d'entre elles sont supprimées car elles correspondent souvent à une particule qui se coince dans un recoin de caverne et qui n'avance pas.

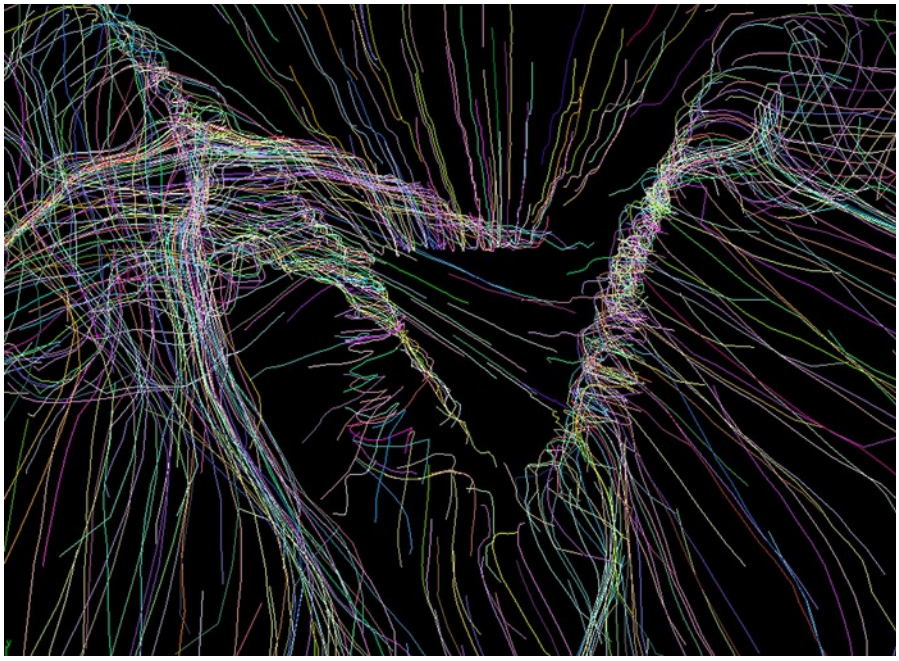


Illustration 28: Lignes créées suite à la simulation de particules

Ensuite, à chaque nouvelle courbe est attribué un rayon aléatoire et ses extrémités sont enfoncées dans le sol, pour de pures raisons esthétiques.

2.4.3 Les UV procéduraux

Les UV correspondent, en gros, au « papier peint » que l'on applique sur un objet 3D. Pour les obtenir, nous avons donc besoin de « déplier » ce modèle 3D pour en faire une version 2D, comme un patron, sur laquelle il est plus facile d'appliquer une texture. U et V désignent les axes de ce plan 2D, comme XYZ le sont pour un objet 3D. Dans ce domaine, toute la difficulté réside dans le fait de placer les coutures de ce patron au bon endroit. Cela doit souvent être fait à la main mais ici, vu la quantité des lianes, il était primordial que ce soit automatisé.

Mes connaissances dans le domaine des UV procéduraux proviennent essentiellement de deux tutoriels [20][34]. Ces derniers préconisent un dépliage UV sur toute la longueur de la ligne-liane (donc d'abord sur la seule dimension U) et de copier à chaque point de la courbe un cercle au rayon prédéterminé (lui aussi déplié mais sur la dimension V) qui va former une « peau ». Un long tube est donc obtenu, aux UV toujours dépliés, qui s'adaptent au diamètre de la liane ainsi qu'à sa longueur, avec un résultat toujours régulier et non étiré, comme démontré ci-dessous.

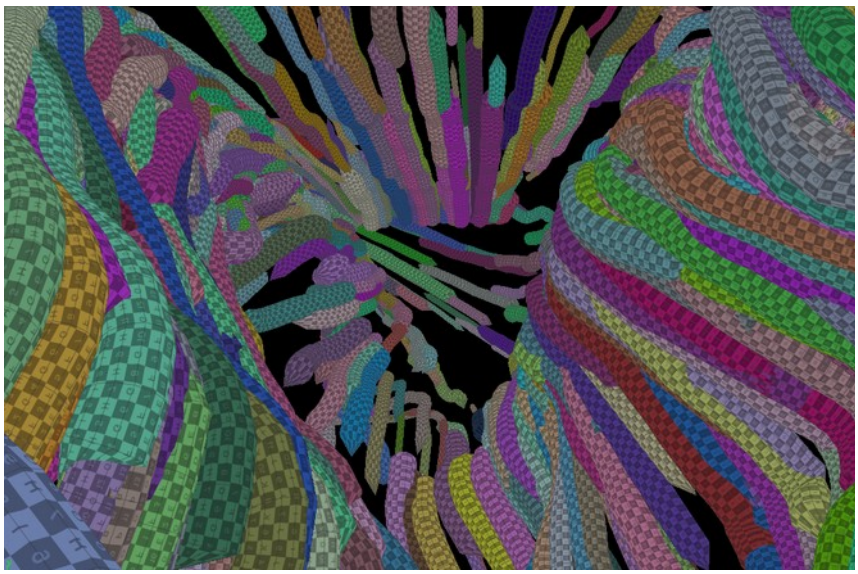


Illustration 29: Lianes-tubes générées avec des UV correctement dépliés, puisque chaque carré de la texture d'exemple reste bien carré.

Ces lianes sont ensuite exportées par paquets dans le format de fichier Alembic vers le moteur de jeu Unreal Engine. Là, mes collègues ont pu appliquer sereinement les matériaux sur chaque liane, placer les lumières et créer les mouvements de caméra afin d'obtenir le rendu ci-dessous :

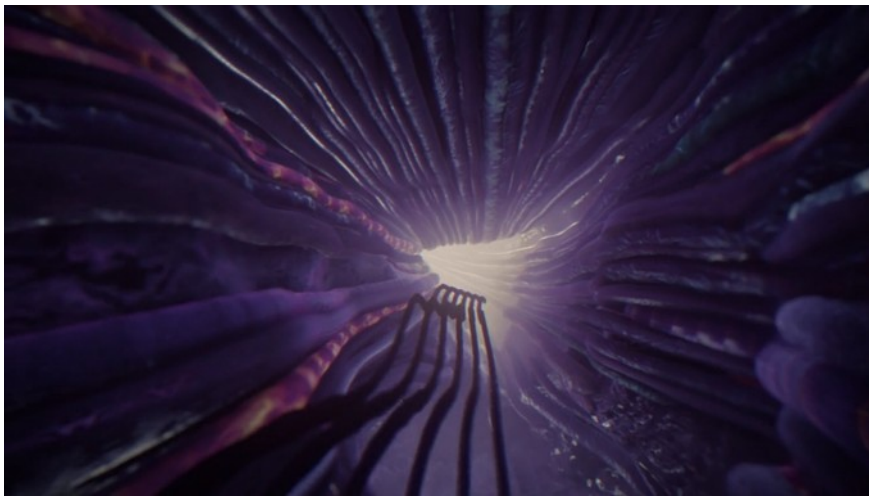


Illustration 30: Rendu final des lianes sur les parois de la caverne dans Unreal Engine

2.4.4 Remarques

- Pendant le projet, nous avons découvert par hasard comment fabriquer des racines biscornues décollées de la surface. Dans notre *workflow*, une caverne est associée à ses lianes. Cependant, au moment de l'intégration dans Unreal Engine, une caverne d'une version précédente a été placée par erreur sous les lianes : nous obtenons alors cette impression de grosses racines qui se soulèvent par endroit, alors qu'en réalité elles courent simplement sur une surface que l'on ne voit pas.

- Il y a beaucoup de collisions entre les lianes car mon système de particules ne prend pas en compte les autres particules : toutes les lignes s'interpénètrent donc… En vérité, ces interpénétrations plaisaient à Isabela notre directrice artistique sur le projet et dans la nature, les racines des arbres proches fusionnent entre elles... donc nous ne nous sommes pas trop attardés sur ce problème. Les interpénétrations trop flagrantes ont été supprimées à la main ou masquées.

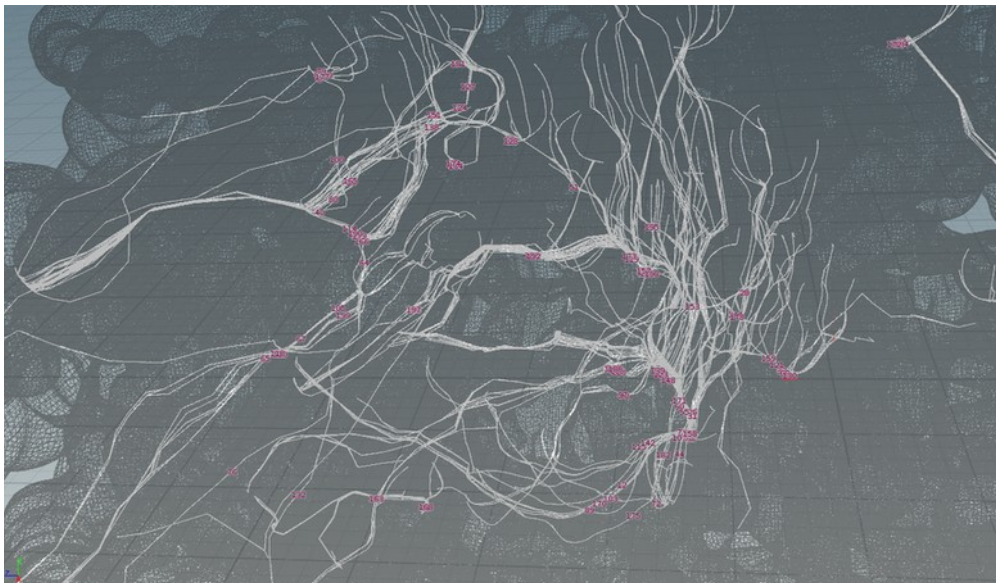


Illustration 31: Les particules (et donc les lignes-lianes en blanc) peinent à recouvrir toutes les surfaces de cavernes (en grisé)

- Mon système de particules ne peut empêcher que ces dernières se coincent dans les petits coins ou finissent par passer par les mêmes tunnels et donc ne recouvrent pas toute la surface de la caverne, comme nous pouvons le deviner sur la capture d'écran ci-dessus. De très grandes quantités de points doivent alors être prévus au départ pour espérer recouvrir toute la surface. Avec le recul, cette méthode n'est pas la plus efficace. La méthode de nervation était exclue car la direction artistique du projet demandait des lianes sans embranchement, cependant j'aurais pu explorer les techniques de découpe comme sur les cartes topologiques ou les « isolines », découvertes notamment dans le tutoriel (non testé) de Rohan Dalvi deux mois après l'intensif [22].
- Ce projet me fait prendre conscience de la nécessité d'améliorer mon outil de plante grimpante, toujours en cours de développement à ce moment de l'année, pour lui permettre de couvrir de larges surfaces et de s'adapter à toutes les dénivellations. Pour le moment mon outil manque d'automatisation, il y a trop de contrôles et de paramètres à régler et c'est fastidieux.

- Néanmoins, le « tout procédural » qui s'adapte à toutes les dénivellations sans vraiment de contrôle esthétique final n'est pas forcément pratique en production. Comme l'a très bien souligné Akriti en plein milieu du projet :« Le procédural c'est pénible». Un contrôle artistique plus fin exige plus de temps.
- Notons que travailler avec Unreal Engine est très satisfaisant car, étant donné sa capacité de moteur de rendu temps réel, nous voyions immédiatement le résultat de notre travail. Nous avons cependant rencontré beaucoup de difficultés, notamment à cause de la différence d'échelle d'un facteur 100 entre ce logiciel et Houdini, l'axe Y qui devient Z dans Unreal, les limitations de taille des fichiers importés en format alembic, et les digital assets (sortes de petits bouts de scripts Houdini qui peuvent fonctionner dans d'autres applications) qui sont très instables, entre autres.

Ce projet intensif a été très riche d'enseignements. L'étude des systèmes des particules a été poursuivie avec le projet personnel décrit ci-après, dans le but de créer une structure ramifiée.

2.5 Arabesques végétales

Pendant le projet intensif, le temps m'a manqué pour comprendre comment gérer des collisions et construire des embranchements avec des systèmes de particules. Les recherches de Matt Estela sur Cgwiki, « Road » et « Tree » [30] m'ont directement inspirée.

Mon objectif ici était de reproduire les arabesques d'inspiration végétale que l'on peut retrouver dans beaucoup d'œuvres affiliées au courant Art Nouveau. Le résultat final ressemble à un masque finement ciselé sur un visage et les étapes pour y arriver sont décrites ci-dessous.

2.5.1 Descriptif du process

Mon support est un scan du buste de la Pieta de Michel-Ange (disponible sur https://www.myminifactory.com/). Le nombre de polygones est alors drastiquement diminué pour garder une scène légère.
Le visage est sélectionné, car c'est la seule zone sur laquelle les particules vont se déplacer, ainsi qu'un point dessus, qui va servir de point de départ aux particules et donc aux arabesques. Ces deux informations, le visage et le point, sont entrées dans le système de particules d'Houdini (appelé « popnet »), où une succession de processus vont être appliqués et recalculés à chaque image.

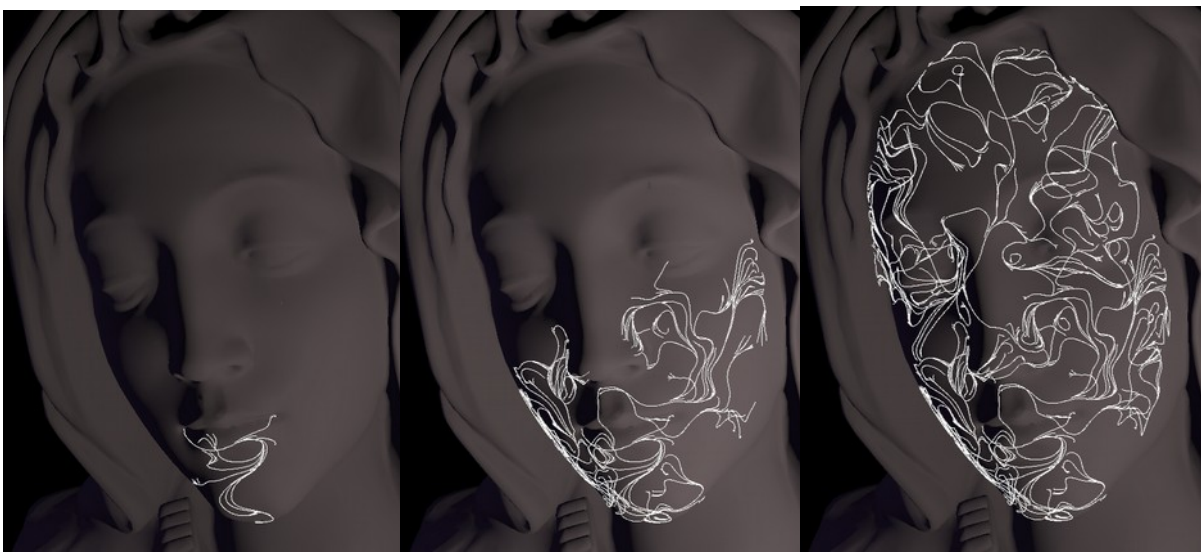


Illustration 32: Calcul du système de particules sur 40 images (gauche), 140 images (milieu) et de 240 images (droite).

Comme pour Valhalla, une force avec un « bruit bouclé » (*curl noise*) est appliquée à chacune des particules (notons qu'au moment du calcul de la première image, il n'y en a qu'une) qui vont ensuite être plaquées sur le visage de la Pieta.

Puis, l'âge (c'est-à-dire le temps depuis sa création) de la particule est considéré. S'il dépasse une certaine valeur, plusieurs autres particules sont générées à l'endroit de la première avec une certaine vélocité initiale, établissant les nouvelles bases de plusieurs branches.

Ensuite, chaque particule compte le nombre de particules proches d'elle dans un certain rayon. S'il y en a trop, la particule au centre est « tuée », elle disparaît et sa branche s'arrête.

Enfin, les particules sont légèrement repoussées entre elles et la dernière étape conduit chaque particule à générer une particule immobile à l'endroit où elle est à cet instant, comme un caillou du petit Poucet. Cette particule « chemin » sort de la boucle de calcul et reste là où elle est, alors que toutes les autres particules encore en mouvement vont continuer et repasser par toutes les étapes décrites précédemment.

Illustration 33: Dernière image du rendu animé du projet Arabesques

En sortie de cette boucle de simulation, tous les points émis par l'avancée d'une même particule sont reliés pour obtenir les branches qui forment de belles arabesques. Ces dernières sont

légèrement décollées du visage pour un effet masque et un diamètre aléatoire est appliqué à chaque courbe.

La technique de rendu est similaire à celle utilisée pour le projet de racines dorées (en moins bien) et peut être visible sur l'illustration 33 ci-dessus.

2.5.2 Remarques

Cette méthode de ramification donne un résultat plutôt satisfaisant, plusieurs problèmes sont cependant constatés :

- A chaque embranchement, cinq ou six particules sont émises en même temps et non une seule. Ce qui fait que les particules détectent un grand nombre de voisines et ont tendance à se « tuer »: impossible de faire la différence entre des particules tout juste générées et celles qui proviennent de branches voisines…
- Malgré la fonction qui repousse les particules entre elles, les collisions existent toujours et certaines branches se recouvrent.
- Par cette méthode, le contrôle artistique est complexe : cet aspect est obtenu en modifiant plusieurs paramètres au hasard et il est très difficile de contrôler plus précisément les embranchements, à moins de retoucher les courbes à la main après la simulation.

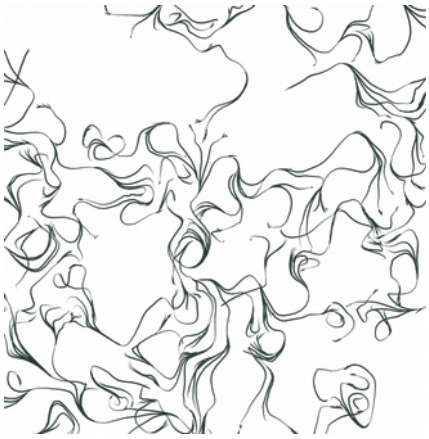


Illustration 34: Arabesques sur support plat

Ci-contre une version de ce masque d'arabesques à plat.

2.6 Générateur de plantes grimpantes

Dans le cadre de mon alternance au département FX d'Illumination Mac Guff, ma principale mission était de développer un outil, dans Houdini, de génération de plante grimpante le plus contrôlable possible artistiquement. L'objectif était d'habiller un décor, plutôt en arrière-plan, avec de la végétation grimpante, dans le contexte d'un long métrage d'animation. Mes recherches pour cet outil se sont déroulées tout au long de l'année et elles ont été grandement alimentées par mes découvertes faites lors de mes projets personnels parallèles.
Plusieurs consignes m'ont été données pour la conception de cet outil :

- en entrée, il devait recevoir, d'un côté, une courbe-guide dessinée par l'utilisateur pour donner la direction générale de la croissance du lierre, et de l'autre, le décor à recouvrir,
- en sortie, une plante grimpante crédible, qui suit cette courbe-guide, devait être modélisée, et l'accès aux courbes formant sa structure devait être facile d'accès afin de réaliser des simulations dynamiques ultérieures,
- autant que possible, mes études devaient partir de plantes grimpantes réelles et s'affranchir de la simple application de tutoriels,
- et enfin, des ramifications et des feuilles devaient être placées le long de cette courbe dessinée par l'utilisateur.

Les quatre versions que j'ai développées de cet outil sont détaillées dans les pages qui suivent et chacune d'entre elles présente une logique de construction différente.

2.6.1 Les plantes grimpantes dans la nature

C'est par de nombreuses recherches sur l'aspect des plantes grimpantes que mon travail a débuté. Elles existent en une infinité de variétés, c'est donc le lierre grimpant (ou *Hedera Helix*) qui a d'abord retenu mon attention, puisqu'il est très présent dans nos contrées.

2.6.1.1 Observations sur les feuilles

Sur les photographies ci-après, nous constatons que les feuilles de lierre sont globalement orientées dans le même sens, c'est-à-dire avec une pointe vers le bas, mais présentent de légères variations dans tous les axes de rotation. Le plan de la feuille est souvent parallèle à celui du décor sur lequel le lierre grimpe. La petite tige à la base d'une feuille (ou pétiole) n'est pas dans le même plan qu'elle et lui est pratiquement orthogonale.

Illustration 35: Photographies de feuilles de lierre

Souvent, nous observons une alternance de position de la feuille entre la gauche et la droite (en quinconce) de la tige principale (ci-dessus, photographie du milieu).

La taille des feuilles semble dépendre de leur position sur la tige principale et probablement de l'âge de cette tige. Il peut y avoir de grosses variations de taille au sein d'un même plant de lierre.

2.6.1.2 Observations sur la structure du lierre

Illustration 36: Photographies de structure de lierre

La structure du lierre peut être assimilée à celui d'un arbre ou même d'une structure fractale, c'est-à-dire que sur une première branche va pousser d'autres branches sur lesquelles vont pousser d'autres branches et ainsi de suite.

Nous constatons que certaines branches ne collent pas parfaitement au mur, qu'elles peuvent se recouvrir les unes les autres pour remplir un espace, voire parfois qu'elles pendent dans le vide (voir photographies ci-dessus) et qu'elles sont plus ou moins apparentes sous les feuilles.

Dans toute la suite de ce document, les termes « plante grimpante » ou « lierre » seront employés indifféremment pour désigner l'un comme l'autre. Mon intention est de fabriquer un outil qui permette autant de générer du lierre (caractérisé par son aspect général et ses feuilles) que n'importe quelle autre plante grimpante. Plusieurs tutoriels m'y ont aidée [21][32].

2.6.2 Outil 1ère version : le problème d'orientation le long d'une branche

2.6.2.1 L'algorithme de nervation ?

Mes premières pistes ont consisté en l'application de l'algorithme de nervation, déjà utilisé pour les ronces. Le résultat (ci-contre) est plutôt crédible à première vue, cependant :

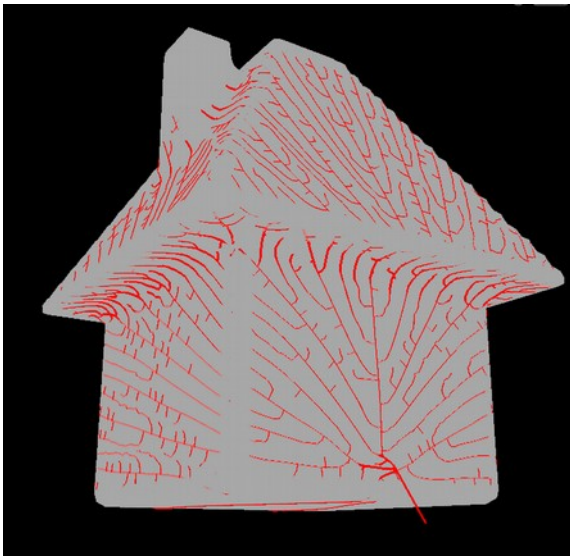


Illustration 37: Application de l'algorithme de nervation comme structure de lierre

- les angles de la géométrie du décor (ici une petite maison) ne sont pas bien pris en compte. Il faudrait que des points attracteurs soient placés très précisément tout le long des arêtes...
- Cette méthode ne permet quasiment aucun contrôle de l'aspect général des branches. Nous pouvons à peu près indiquer à l'algorithme les zones à remplir mais aucun moyen n'est à notre disposition pour indiquer la croissance générale du lierre et de ses ramifications.
- L'algorithme de nervation ne permet pas de recouvrement entre les branches, alors que des enchevêtrements sont observés dans la nature.

Cette méthode est donc écartée.

La courbe tracée par l'utilisateur pour donner la direction de croissance du lierre, est alors prise comme point de départ de la suite de ma réflexion.

2.6.2.2 Projection sur une surface

Cette courbe doit être projetée sur une surface, puisque les plantes grimpantes ont besoin d'un support pour pousser. Différentes méthodes de projection sur un décor sont étudiées, chacune consistant soit à :

- chercher la surface la plus proche et coller les points de la courbe dessus,
- appliquer une direction aux points de la courbe et « les arrêter » quand ils rencontrent la surface (projection selon un vecteur),
- chercher la normale (c'est-à-dire la droite perpendiculaire à la surface) du polygone du décor le plus proche et plaquer les points de la courbe selon cette direction,
- détecter la direction du « champ de force » qui entoure un décor et plaquer le point sur le décor suivant cette direction.

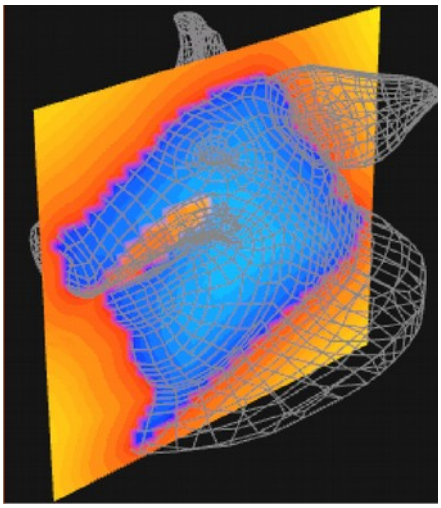


Illustration 38: Visualisation du champ de force orienté

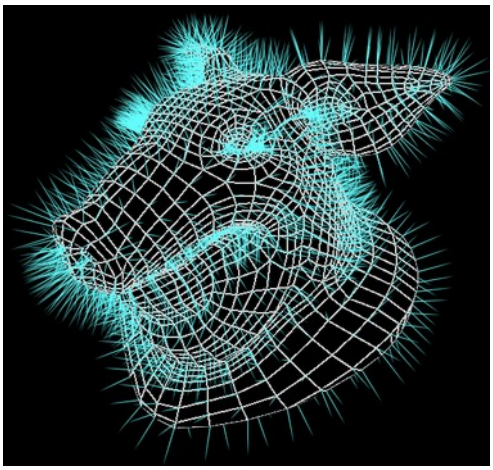
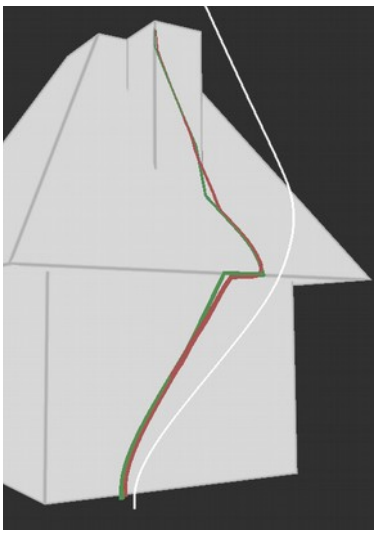


Illustration 39: Les normales des points sur une modélisation de tête de cochon

Cette dernière solution implique la conversion du décor (qui est un assemblage de polygones) en un volume appelé « champ de distance orienté » (*Signed Distance Field* ou SDF en anglais) que l'on peut visualiser dans l'exemple sur l'illustration 38 ci-dessus. Sur cette modélisation de tête de cochon, un plan de coupe représente le champ de distance, dans les tons orangés s'il est à l'extérieur du volume et dans les tons bleus à l'intérieur.

L'avantage par rapport aux autres méthodes est que ce champ est continu et qu'il n'y aura pas d'écart entre deux directions de projections. Comme exposé sur l'illustration 39 ci-dessus, si nous choisissons la méthode de la normale, il y a une grande différence entre les directions de deux normales (traits bleus) dans les zones anguleuses, comme le bout de l'oreille. Une projection dans cette zone donnerait des résultats incohérents.

Illustration



40: Comparaison des méthodes de projection

S'agissant de l'élaboration de l'outil, une courbe-guide est tracée (la ligne blanche sur l'illustration 40 ci-contre) et la méthode de projection avec champ de distance (la ligne rouge) est comparée à celle qui cherche la surface la plus proche (ligne verte). La méthode qui semble la moins angulaire et potentiellement plus organique est de toute évidence la première, par champ de distance : c'est elle qui sera conservée dans la suite du développement de l'outil, la plupart du temps. C'est aussi la méthode la plus longue à calculer (à cause de la conversion en volume), donc si les environnements sont imposants, la méthode de projection simple sera préférée.

La possibilité de dessiner plusieurs courbes est très vite incluse dans cet outil en plein développement.

2.6.2.3 Orientation le long d'une courbe

Une fois les courbes projetées sur une surface, des feuilles sont réparties dessus. Mais pourquoi vont-elles dans n'importe quel sens, devant ou derrière les courbes (cf. ci-contre)?

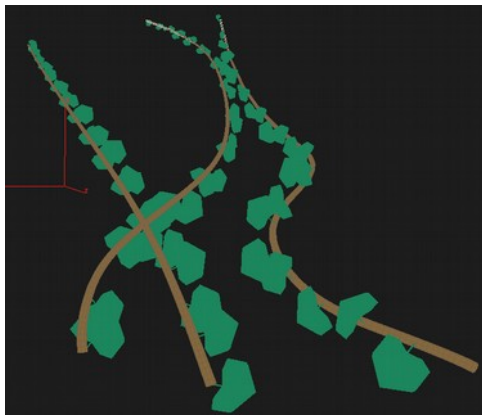


Illustration 41: Feuilles réparties dans n'importe quel sens le long des branches

C'est que les courbes, ici, ont des directions un peu complexes et Houdini attend qu'à chaque point accueillant une feuille, un repère orthonormé soit défini. Ce repère traduit l'orientation précise de la feuille.

Dans notre cas, nous souhaitons que les feuilles (dont le repère personnel est défini comme sur l'illustration 43 ci-contre) soient toujours parallèles au décor et pointent globalement vers le bas comme sur le schéma de l'illustration 42.

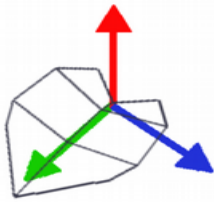


Illustration 43: Repère orthonormé de la feuille

Nous pouvons récupérer l'information sur la direction du décor grâce au champ de distance orienté, qui se traduit par la flèche rouge sur le schéma de l'illustration 42 ci-contre. Pour la flèche verte, nous pouvons partir d'un vecteur pointé vers le sol et le recalculer pour qu'il soit

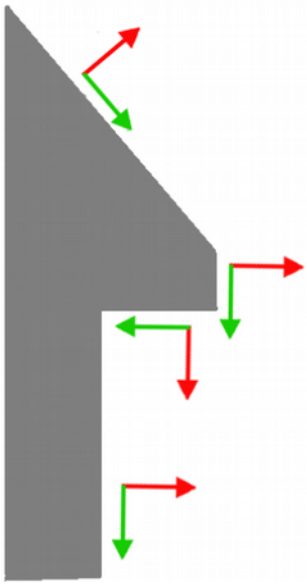


Illustration 42: Orientation souhaitée des feuilles le long du décor

toujours orthogonal à la flèche rouge grâce à un double produit vectoriel (l'excellent tutoriel d'Entagma explique très bien ce sujet délicat [27]). Le troisième vecteur de notre repère se déduit grâce à un produit vectoriel des deux premiers, et voilà nos trois vecteurs qui définissent en chaque point l'orientation que doit avoir la feuille.

Ces 3 vecteurs sont proprement rangés dans une matrice de rotation qui est elle-même transformée en quaternion, outil mathématique non-intuitif dont Houdini a besoin pour faire ses calculs.

Une fois maîtrisé ces problèmes complexes de repères dans l'espace, il est facile de faire en sorte qu'il y ait alternativement des feuilles à droite et à gauche de la branche. De même, grâce à cette méthode d'orientation, des branches secondaires, modélisées à part avec des formes aléatoires, sont copiées de chaque côté de la branche principale puis projetées sur le décor.

Des variations infimes d'angle dans l'orientation des feuilles et des branches apportent un peu plus de crédibilité à l'ensemble et la taille des feuilles est diminuée au fur et à mesure qu'elles s'éloignent du sol, comme sur la capture d'écran ci-dessous (illustration 44).

2.6.2.4 Présentation de l'outil à mes superviseurs

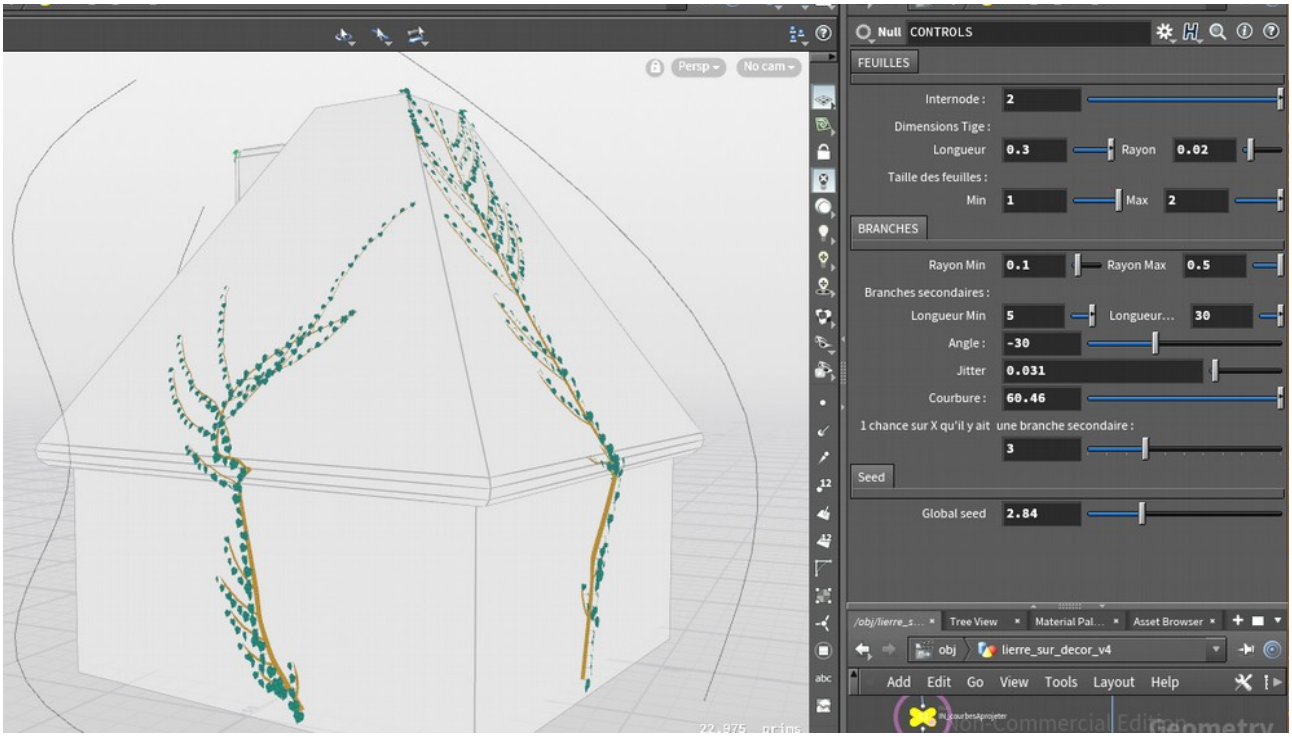


Illustration 44: Interface dans Houdini de la 1ère version d'outil de plante grimpante

Au bout d'un mois de recherches en tant qu'alternante, cette première version de mon outil de plante grimpante est présentée à mes superviseurs : à partir d'une ou plusieurs courbes dessinées par l'utilisateur et d'un décor/support (mur, maison, arbre…), un lierre couvert de branches secondaires et de feuilles se colle au décor donné. Beaucoup de paramètres sont accessibles et

peuvent être modifiés : angles, longueurs, rayons, taille des feuilles, courbure des branches secondaires….

Mes superviseurs déplorent que l'outil a trop de paramètres inutiles et produit une organisation anarchique des branches le long du tronc principal. Certes, mais toutes ces recherches m'ont néanmoins permis de comprendre beaucoup de choses sur les rotations de repères et l'utilisation pratique des quaternions. Une 2ème version de mon outil avec une approche totalement différente a ensuite été mise au point.

2.6.3 Outil 2ème version : le décor comme champ de force

Comme la forme des branches secondaires semblait échapper à mon contrôle, une nouvelle approche m'a été suggérée: contrôler le décor. L'idée est de créer une sorte de « champ de croissance» et d'indiquer, à chaque fois qu'une branche s'approchera d'un certain endroit du décor, dans quel sens elle devra pousser ensuite.

2.6.3.1 Calcul du champ de croissance

Tout d'abord, la question est de calculer ce champ de croissance à chaque point de la surface du décor, en fonction des directions données par les courbes dessinées par l'utilisateur. Peindre à la main tout un décor serait trop long, une base calculée automatiquement est nécessaire.
Dès lors, mon support de test est assimilé à un mur plat pour simplifier.
Cinq ou six algorithmes différents, pour obtenir la direction propre à chaque point du mur (les traits en bleu sur les illustrations ci-dessous) ont été testés en fonction de la direction des courbes dessinées (en blanc). Voici les deux principaux :

- direction des points sur le mur = direction de la courbe en son point le plus proche (ci-dessous à gauche) ;
- direction des points sur le mur = interpolation linéaire entre chaque direction des points les plus proches de chaque courbe, avec pondération par rapport à la distance (ci-dessous à droite).

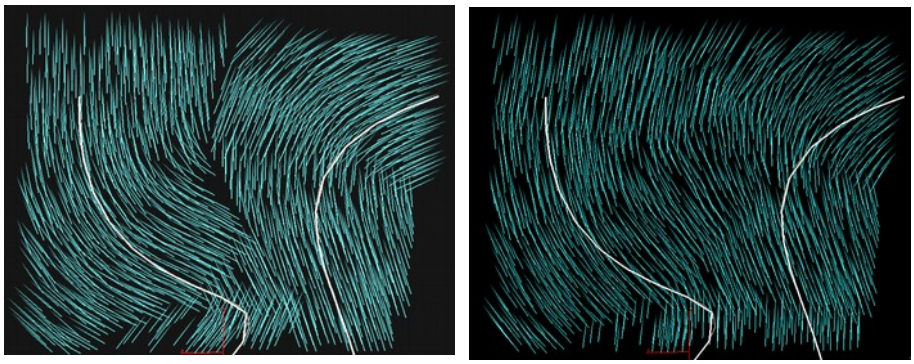


Illustration 45:A gauche, direction de points calculée par proximité avec les courbes utilisateur. A droite, direction des points calculée par interpolation linéaire

L'interpolation linéaire donne les résultats les plus cohérents et homogènes et sera donc utilisée dans la suite du développement de l'outil.

2.6.3.2 Premier niveau de branches

Sur les courbes principales, dessinées par l'utilisateur et qui font office de **tronc**, des curseurs peuvent régler les zones de croissance: en blanc ça poussera, en rouge aucune branche ne poussera (Illustration 46 ci-dessous).

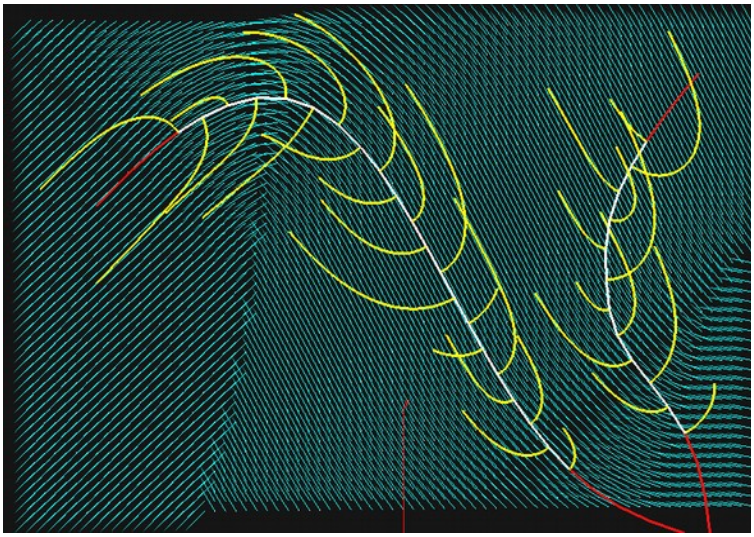


Illustration 46: Tronc et branches secondaires sur mur orienté

Ensuite, tout au long de cette courbe blanche, des points sont ajoutés les uns derrière les autres pour créer la base des branches (en jaune) qui alternent de chaque côté des courbes et qui sont de longueurs différentes.

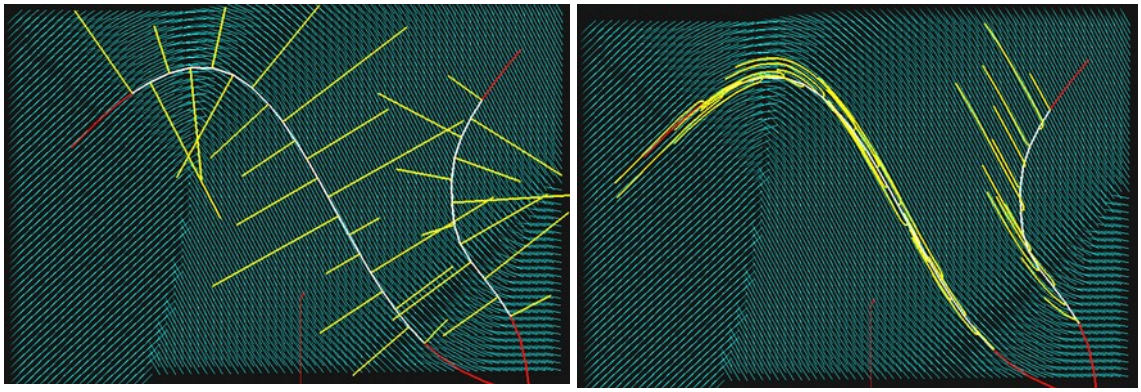


Illustration 47: à gauche les branches suivent la bitangente, à droite elles suivant la courbe principale

Les branches grandissent au fur et à mesure que la position de chacun de leurs points est calculée : leur prochaine direction est établie en faisant une interpolation linéaire entre la perpendiculaire à la courbe principale (qu'on appelle bitangente) et la direction présente sur le mur (en bleu toujours). Une variable aléatoire est ajoutée dans la longueur des branches pour un rendu plus organique. L'illustration de gauche ci-dessus présente ce qui se passe si les points qui

forment la branche suivent uniquement la direction de la bitangente et l'illustration de droite si seule la direction du décor est prise en compte.

2.6.3.3 *Bruit et décollement*

Sur les branches jaunes, du bruit (c'est-à-dire de petites variations aléatoires) est ajouté dans la position de chaque point (illustration 48). Elles sont ensuite plaquées plus ou moins proches du mur suivant la rampe (modifiable par l'utilisateur) présentée sur l'illustration 49, pour un effet plus ou moins décollé. Sur cette rampe, la droite correspond au début de la branche et la gauche à la fin de la bra nche. Si la valeur est à 1 (c'est-à-dire que la courbe est au point le plus haut), les points de la branche dans cette zone restent au même niveau que le tronc principal. Si la valeur est à zéro, ces points sont collés au mur. Cet effet fonctionne en récupérant seulement la normale du point du décor le plus proche et sa distance avec le point de la branche.

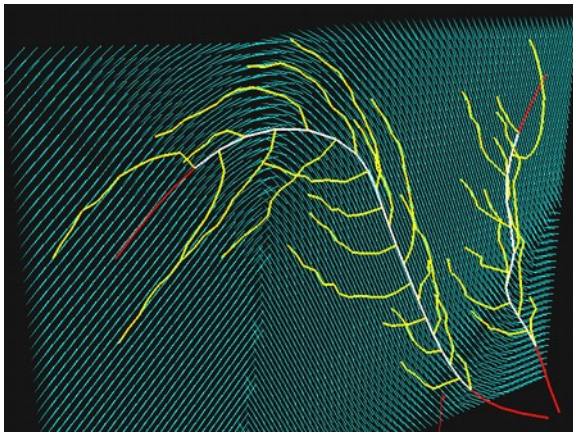


Illustration 48: Branches bruitées et plaquées sur le mur

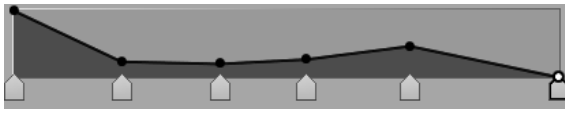


Illustration 49: Forme de la rampe qui détermine comment les branches vont se coller au mur

2.6.3.4 *Un décor bien coiffé*

De plus petites branches sont ajoutées sur les branches existantes, comme ci-dessous.

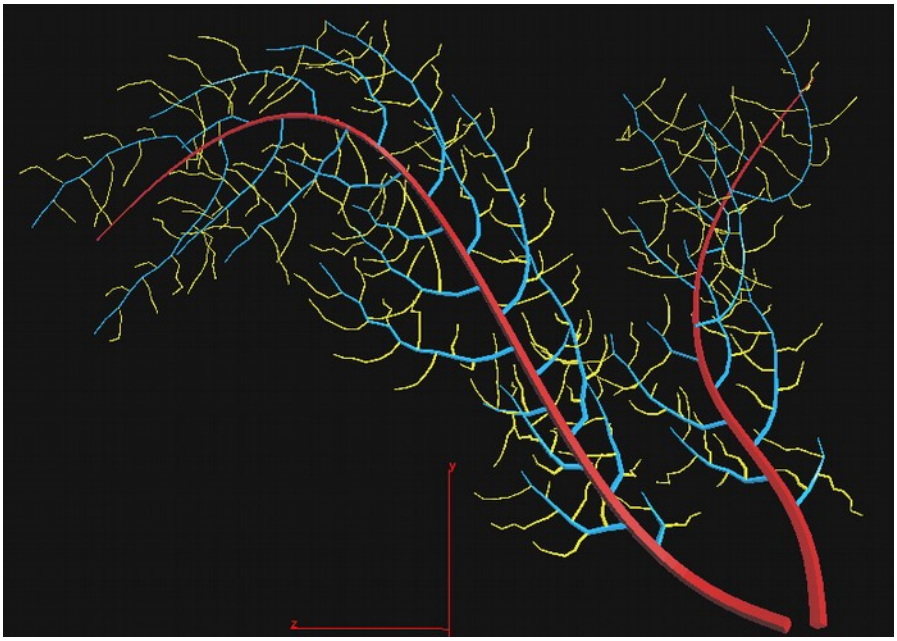


Illustration 50: Un tronc, des branches et des brindilles

Grâce à divers paramètres exposés dans une interface utilisateur, la forme des courbes principales, secondaires et tertiaires, leur rayon, leur longueur… peuvent être modifiés. Tout est pris en compte en temps réel et le résultat apparaît aussitôt qu'un paramètre est changé.
Pour modifier la direction de croissance donnée par le décor (et calculée automatiquement), des outils qui gèrent habituellement la pose de poils, cheveux et fourrures dans Houdini (« hair grooming ») sont utilisés. Ils nous permettent de «coiffer» directement à la souris les directions des points du mur dans la direction souhaitée. Nous pouvons constater l'impact sur la croissance des branches sur l'illustration 51, à droite, à comparer avec l'illustration 50.

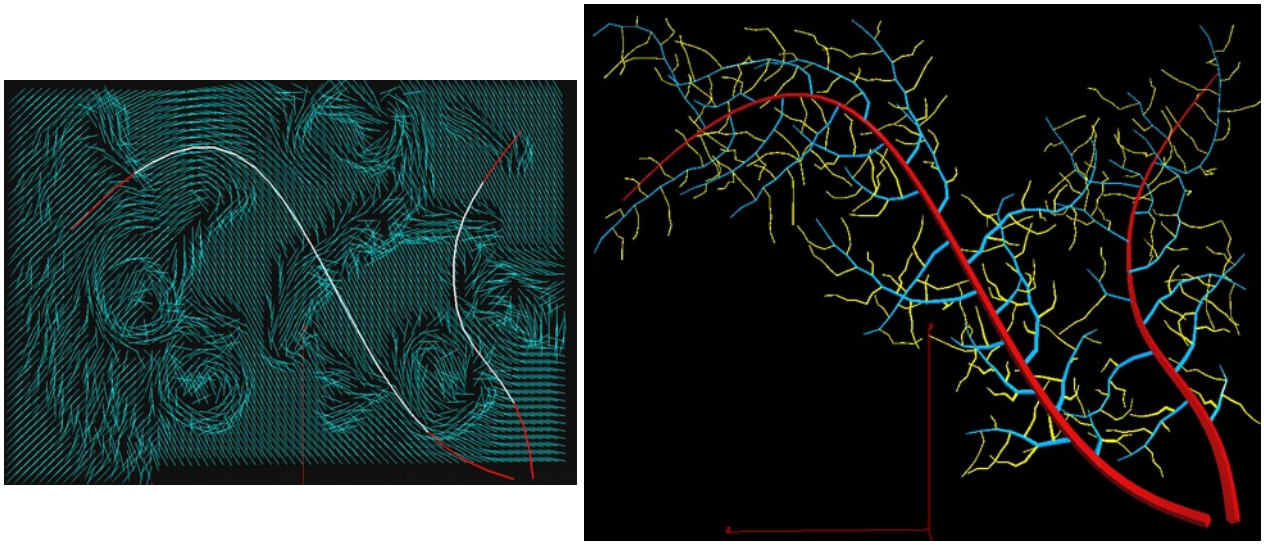


Illustration 51: A gauche, modifications manuelles dans les directions de pousse, à droite l'impact sur la structure globale

Pendant la conception de l'outil, il m'a été conseillé de séparer les problématiques de branches (la structure) et de feuilles (qui sont réparties sur cette structure). Dans cette version de l'outil, la question des feuilles n'a pas eu le temps d'être traitée.

En effet, l'idée de champ de force à peindre est vite abandonnée. Elle est certes très originale mais ne donne pas forcément de bons résultats et demande beaucoup de temps avant d'obtenir un rendu harmonieux.

2.6.4 Outil 3ème version : retour aux bases et regard extérieur

Pour cette nouvelle version, simplicité et épuration des paramètres ont été recherchées pour un résultat plus convaincant.

2.6.4.1 Simplification

Mon outil devient alors un outil de modélisation procédurale simple à 4 niveaux (troncs où sont réparties les branches sur lesquelles sont réparties les brindilles, les feuilles étant réparties sur l'ensemble).
Tout ce qui se rapproche d'une « pousse » en fonction de l'environnement est supprimé, puisqu'en réalité une branche ne doit être cohérente en direction qu'avec sa branche parente et non le reste du décor. C'est l'enseignement principal à retenir de la précédente version de l'outil, à ce moment-là de l'exploration de mon sujet. Seule l'étape de projection sur le décor est conservée.

Par ailleurs, travaillant dans un contexte de long métrage d'animation, l'esthétique prime sur le réalisme botanique. Ainsi, plutôt que d'avoir des branches tordues qui partent dans tous les sens, des branches esthétiques et procédurales sont créées, grâce à une NURBS (courbe qui permet une représentation exacte de fonctions polynomiales) avec quatre points de contrôle. Les coordonnées en Y des deux points centraux sont modifiées aléatoirement pour donner des branches à la forme variable mais à l'esthétique contrôlée, réparties ensuite sur le tronc.

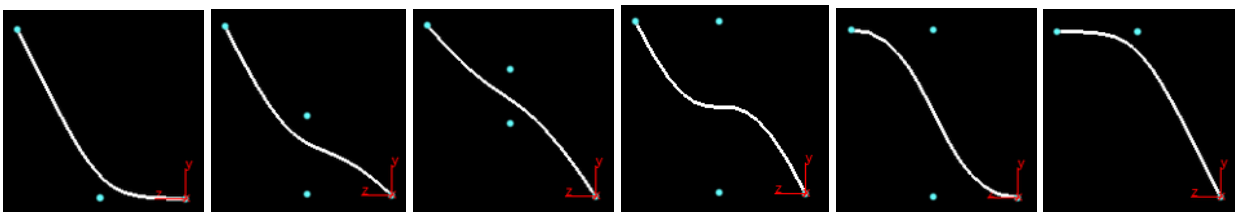


Illustration 52: Différentes variations de courbes-branches esthétiques

Des outils précédents, le système de décollement du mur contrôlé par rampe est conservé, le bruit sur les positions des points aussi, ainsi que tout ce que nous avons appris sur les rotations et les quaternions. Toutes les branches, les brindilles et les feuilles se copient alors dans le bon sens.

Les brindilles (les petites branches sur les branches) n'ont pas de forme esthétique particulière et sont simplement des lignes tordues.

Sur l'illustration ci-dessous, après mise à jour de l'outil, la structure de la plante grimpante se compose d'un tronc en rouge, de branches en bleu et de brindilles en jaune. La courbe blanche

correspond à celle initialement dessinée par l'utilisateur. Nous constatons ici que le tronc est bien évidemment trop large pour du lierre, mais pourrait être crédible sur un rosier grimpant. De toute façon, le diamètre du tronc est un paramètre facile à régler.

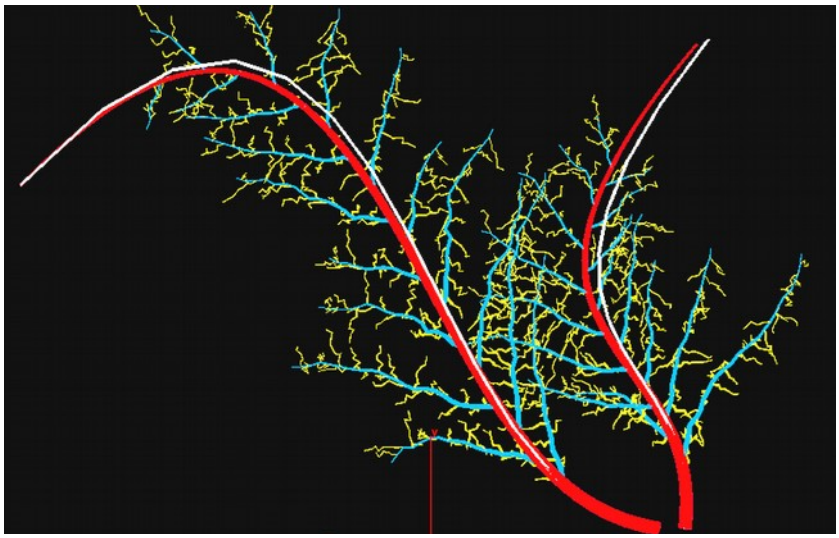


Illustration 53: Outil simplifié

2.6.4.2 Les feuilles

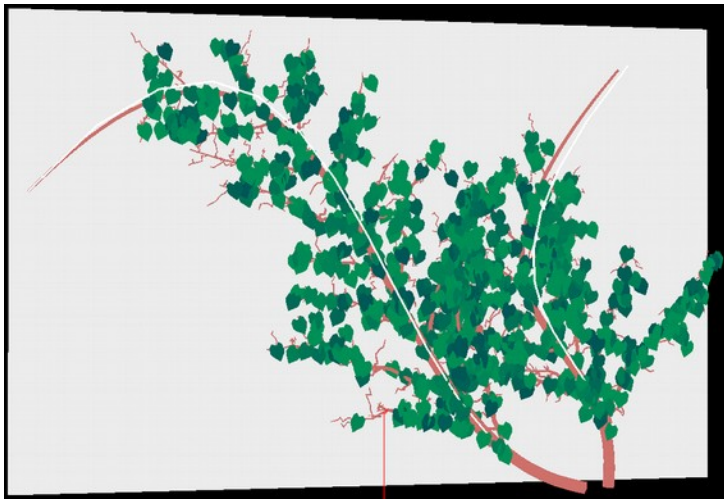


Illustration 54: Outil avec les feuilles par défaut

Tout ce que nous savons déjà sur les feuilles depuis la 1ère version de l'outil est implémenté ici. Une petite tige à la feuille (ou pétiole) est ajoutée, pour la décaler des branches et du décor. Ces tiges sont globalement orthogonales au plan des feuilles, comme constaté lors de mes recherches sur le lierre. Les feuilles sont contrôlables en rotation selon tous les axes et de l'aléatoire généré par l'ordinateur est appliqué pour donner un peu plus de vie et moins d'uniformité.

Il est possible de « brancher » en entrée de l'outil une feuille de son choix, à appliquer sur toute la structure, ou alors de sélectionner la feuille de base fournie par défaut.

La différence principale avec les précédentes versions est l'utilisation *d'instances* pour placer les feuilles. Auparavant, l'orientation de chaque géométrie de feuille était modifiée avant d'être copiée sur la structure : c'était très lourd à calculer, car des milliers de feuilles différentes étaient présentes dans la scène. A présent, toutes les informations sont contenues dans les points à l'extrémité des tiges qui vont accueillir les feuilles : ce sont les points qui sont orientés et c'est la

même feuille qui est copiée partout. La feuille devient une *instance* et la scène ne contient plus que des points et une seule géométrie, la rendant bien plus légère et rapide à calculer. De plus, cette méthode permet un export facile vers un autre logiciel ou moteur de rendu, tant qu'il gère aussi les *instances*.

Pour contrôler la densité des feuilles en certains endroits du lierre, une technique à base de *metaballs* est utilisée. Une *metaball*, dans notre cas et pour simplifier, est une sphère en 3 dimensions qui présente la particularité d'être plus « dense » en son centre et moins en allant vers sa surface, donnant ainsi un dégradé radial naturel. Les *metaballs* ont aussi la capacité de fusionner entre elles comme des « blobs », créant des formes très organiques. Ces objets sont présents par défaut dans Houdini et peuvent être manipulés facilement.

Dans mon outil, avant de générer la plante grimpante, l'utilisateur peut donc placer des points dans l'espace, proche du décor, vers les zones où il veut voir plus de feuilles qu'ailleurs. Une *metaball* est copiée sur chacun de ces points et influence en douceur la densité des feuilles à ces endroits.

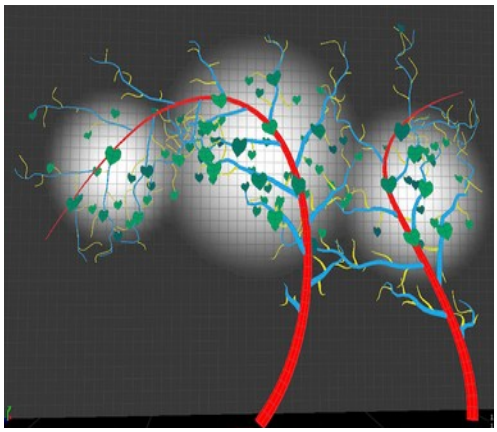


Illustration 55: Variation de densité de feuilles grâce aux metaballs

Dans l'illustration ci-contre, les taches blanches sont la représentation de trois *metaballs* qui commandent la densité des feuilles sur le mur. Ici, leur influence est très forte car il n'y a plus aucune feuille en dehors de ces taches. Cette influence peut être dosée, offrant ainsi des îlots légèrement plus denses de feuilles sur l'ensemble du lierre.

2.6.4.3 Critique de l'outil

Mon travail est présenté à un graphiste modeleur du département *modeling* d'Illumination Mac Guff. Habitué à Speedtree, il avait essayé de modéliser du lierre à l'aide de ce logiciel, avec des résultats mitigés et trop aléatoires. Il était dont très intéressé par mon outil qui, selon lui, offre plus de contrôle, plus rapidement ; il est aussi plus ciblé pour la tâche. Plusieurs améliorations m'ont été suggérées, notamment avoir un meilleur contrôle du bruit sur les branches et les brindilles (en amplitude et fréquence) et un contrôle plus complet sur les rotations aléatoires possibles des feuilles, surtout que ce sont elles qui se voient le plus.

Sur la capture d'écran ci-dessous, les feuilles sont masquées.

Cette capture d'écran montre aussi (en bas à droite) que l'on peut « brancher » en entrée de l'outil la ou les courbe(s) dessinée(s) par l'utilisateur, le décor à recouvrir et ainsi qu'une ou des forme(s) de feuilles différentes. Ici le décor est un pseudo-tronc d'arbre modélisé rapidement pour le test.

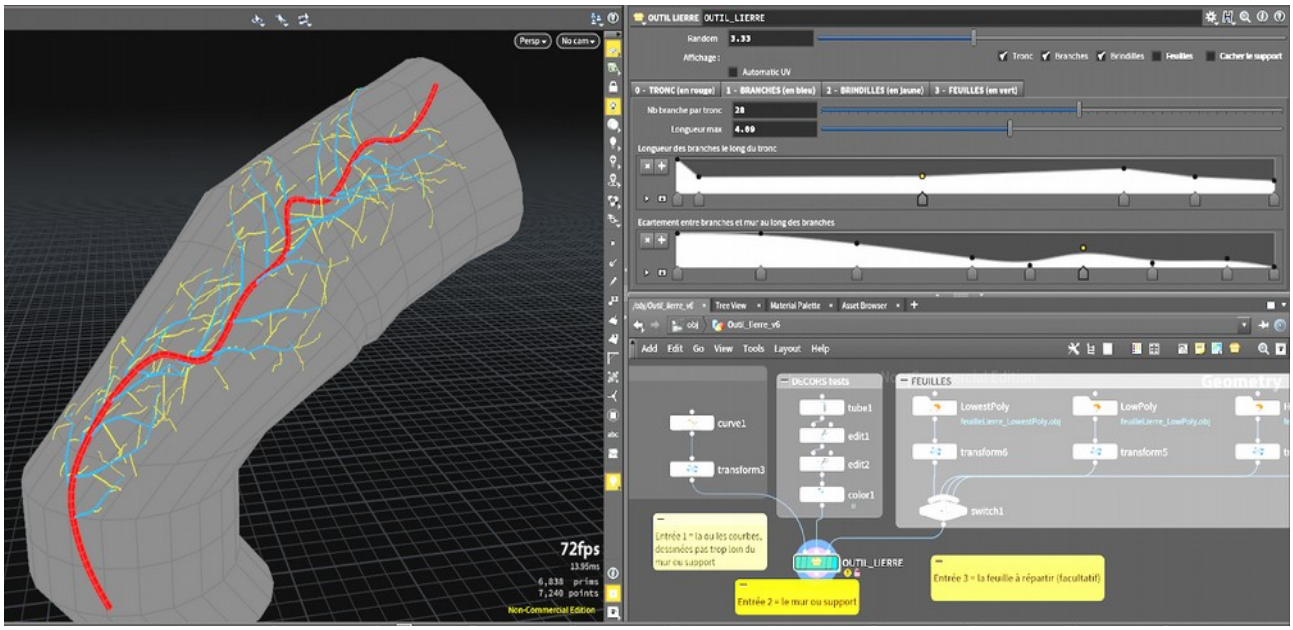


Illustration 56: Capture d'écran de l'outil présenté au modeleur

Notons que la courbe que le modeleur a utilisée en entrée (et qu'on ne voit pas à l'écran) était parfaitement lisse et droite, alors que sa projection sur le décor (le tronc en rouge) est complètement tordue. On pourrait croire que c'est normal mais ce n'est pas le cas : mon algorithme de projection ne correspond pas à ce qu'on attendait et mériterait quelques ajustements.

2.6.4.4 L'outil après Valhalla

Notre projet intensif Valhalla me fait aussi prendre conscience des difficultés de mon outil à gérer correctement les surfaces irrégulières. Comme montré ci-contre, si une courbe principale passe au milieu de deux parties d'un décor (ici un gros arbre mort en vert), les branches, qui sont construites sans se soucier de l'environnement, se plaquent au plus proche, perdant tout réalisme botanique.

Valhalla m'apprend aussi la logique des systèmes de particules et leurs techniques spécifiques pour se plaquer sur des surfaces.

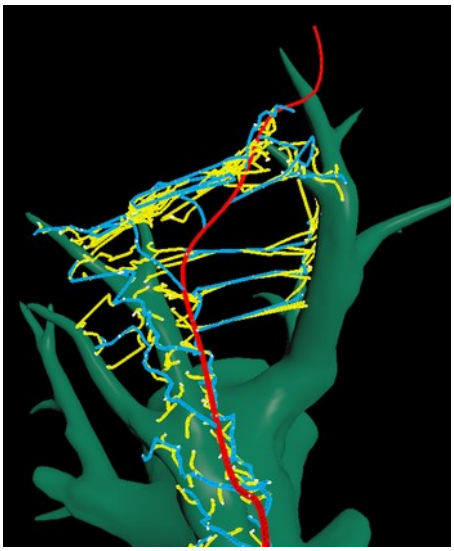


Illustration 57: Problèmes de projection des branches sur un décor biscornu

Au début de la conception de cette nouvelle version de l'outil, supprimer tout ce qui avait un lien avec la pousse de branches n'était donc pas du tout pertinent.

Comme ce qui avait commencé à être implémenté dans la version 2 de l'outil avec le décor coiffé, les branches, initialement modélisées de façon « rigide » et indépendamment du décor, sont remplacées par des particules qui sont plaquées dès leur création sur le décor. En réalité, aucun système de particules n'est utilisé ; sa logique à base de forces influentes est cependant recodée en VEX, le langage de programmation de Houdini, pour s'affranchir de la simulation (qui doit normalement être relancée à chaque modification de paramètres) et garder ainsi un contrôle en temps réel sur l'outil. Les branches épousent alors parfaitement le décor, donnant un résultat très organique tout à fait satisfaisant. L'illustration ci-dessous présente le résultat sur le même arbre mort que précédemment.

Illustration 58: Rendu basique de la structure seule (à gauche) puis avec feuilles (à droite) du lierre sur un décor

Mes supérieurs étaient satisfaits et la mise au point de l'outil aurait pu s'arrêter là.

2.6.5 Outil 4ème version : l'approche botanique

C'était sans compter sur la découverte soudaine et tardive de la vidéo de présentation de l'outil « Ivy Taming » (ou lierre apprivoisé), réalisé par le français Marc Chevry [19].

2.6.5.1 L'outil idéal

Illustration 59: Captures d'écran issues de la vidéo de présentation de l'outil Ivy Taming

L'outil présenté possède exactement toutes les fonctionnalités de tout ce que je souhaitais d'implémenter… Mon nouvel objectif est donc de comprendre son fonctionnement et de l'imiter, simplement en analysant la vidéo de présentation et en procédant à ce qu'on appelle du « reverse engineering ».
Cet outil semble avoir une approche beaucoup plus botanique, c'est-à-dire que son système recrée les forces subies par une plante grimpante dans la nature, comme la gravité ou l'influence du soleil.

Marc Chevry part d'une thèse du suédois Johan Knutzen datée de 2009 intitulée *Generating Climbing Plants Using L-Systems* (Générer des plantes grimpantes en utilisant des L-systèmes).
C'est avec enthousiasme que je découvre cette thèse: elle présente des éléments que j'avais déjà compris via mes projets et outils, confirme quelques idées que j'avais eues sans réussir à les implémenter et m'en apprend de nouvelles.
Par ailleurs, mon travail avait laissé de côté l'approche « L-systèmes », jugeant, comme beaucoup dans le milieu de la 3D, sa grammaire trop peu intuitive [6][9][17]. Ce que propose cette thèse est de garder la logique fractale du L-système mais de recoder (à base de calculs sur des vecteurs) toute la partie implémentation géométrique, donnant un bien meilleur contrôle sur la plante grimpante générée et surtout une prise en compte de l'environnement.

La thèse propose une implémentation dans un langage dérivé du java et présente des concepts génériques de mathématiques. Il n'est pas trop compliqué de les implémenter dans Houdini et de les fusionner en un grand algorithme itératif principal qui va décider de la croissance de la plante grimpante.
La logique générale exposée dans la thèse ressemble à celle d'un système de particules sur lesquelles s'appliquent des forces et dont on va calculer, à chaque itération, la nouvelle direction, après avoir laissé derrière elles un point à l'endroit où elles sont passées .

Étant donné tout le savoir accumulé au cours de l'année, cette étape se fait très rapidement. En cours de développement, je m'aperçois que je suis en train de recoder un L-système basique, stochastique et paramétrique qui s'adapte à son environnement (voir illustration ci-contre). Ici, les pointes rouges sont les « têtes chercheuses » qui font pousser la plante et les traits verts correspondent au chemin qu'elles laissent derrière elles.

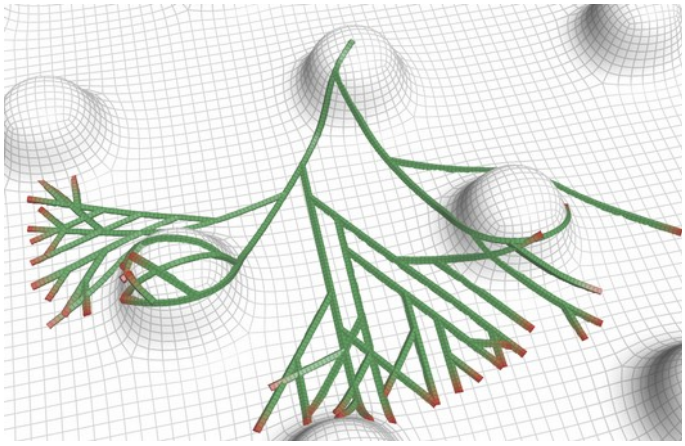


Illustration 60: Prototypage de l'outil qui se présente comme un L-système

2.6.5.2 Les forces en présence

L'algorithme ainsi développé peut prendre en entrée :

- des points qui vont servir de 'premières graines' pour construire les branches du lierre,
- plusieurs courbes, dessinées par l'utilisateur qui donneront la direction de croissance du lierre.

En botanique, on appelle tropismes les forces qui s'appliquent aux plantes. Nous nous concentrerons ici sur le gravitropisme et l'héliotropisme, respectivement la force d'attraction vers le sol et celle vers le soleil (vers le « haut »).

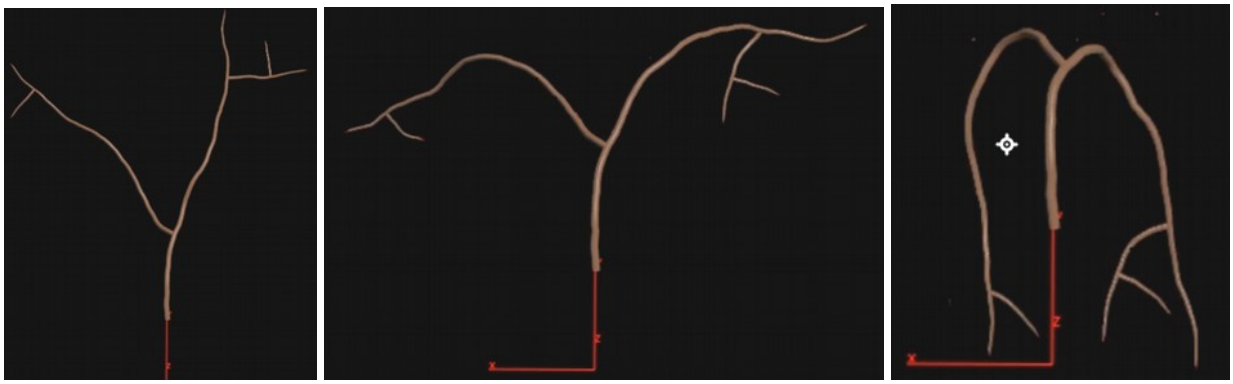


Illustration 61: Influence de la gravité : d'aucune (gauche), à moyenne (milieu) à forte (à droite).

A ces forces 'naturelles', on en joute d'autres qui nous permettent de contrôler le comportement de la particule qui, par son avancée, va composer la branche :

- la force d'influence d'une direction aléatoire, pour obtenir des branches plutôt droites ou plutôt noueuses (cf. illustration 63 ci-dessous),
- la force d'attraction du décor, pour un lierre très collé aux parois ou au contraire très 'aérien' (cf. illustration 64 ci-dessous, gauche et milieu),
- la force d'influence des guides, ces courbes dessinées par l'utilisateur pour guider la direction de croissance du lierre (illustration 62 ci-contre),
- la force d'évitement des obstacles (optionnelle, par exemple, si le lierre court sur une maison, cela permet de contourner une fenêtre).

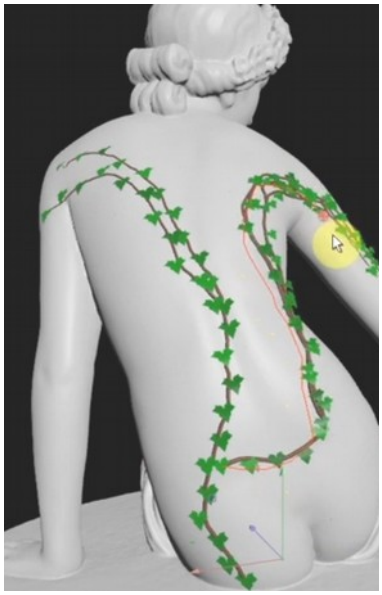


Illustration 62: Influence très forte de deux guides tracés par l'utilisateur (courbes rouges à peine visible)

Chaque force est pondérée par un facteur que l'utilisateur peut modifier à sa guise.

Illustration 63: influence du vecteur aléatoire : de nul (à gauche) à très fort (à droite)

Comme dans les versions précédentes de l'outil, et donc indépendamment de la thèse découverte, une variation aléatoire est appliquée ensuite aux branches pour créer un décollement par rapport au décor (cf. illustration 64 ci-dessous, à droite).

Illustration 64: influence de l'attraction du décor (faible à gauche et forte au milieu). A droite, décollement bruité des lianes par rapport au décor

2.6.5.3 Deux comportements différents

Au sein de l'algorithme, les particules que nous pouvons définir comme « têtes chercheuses » et qui construisent les branches, présentent deux comportements différents (cf.illustration ci-dessous), en fonction de la distance par rapport au décor:

- si elles sont proches du décor : elles sont alors plus ou moins plaquées au décor et subissent une force vers le haut (héliotropisme),
- si elles sont loin du décor : c'est la gravité qui s'applique plus ou moins fortement.

Les autres forces (aléatoire, évitement d'obstacle et guides) s'appliquent en plus, quel que soit le comportement de la particule.

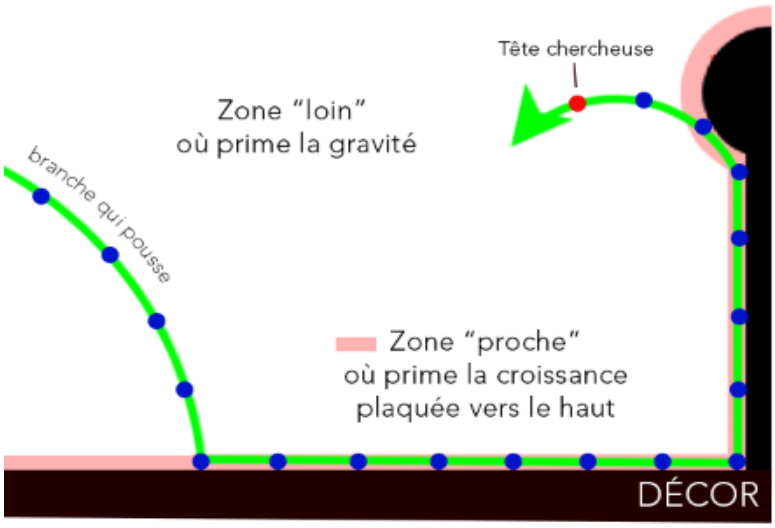


Illustration 65: Schéma explicatif en 2D des différences entre les deux comportements possibles d'une particule

Le paramètre qui donne la distance seuil définissant les zones proches ou non, est évidemment modifiable. Nous l'appellerons « paramètre de suivi du décor » car il va directement déterminer si le lierre va rester collé au décor ou au contraire pendre sur le bord d'un toit par exemple.
Si la valeur de ce paramètre est petite et donc la distance seuil très courte, les différentes forces subies par la particule pourront la faire sortir très facilement de la zone « proche » et elle sera très vite entraînée par la seule gravité, générant des retombées de lierre (cf. illustration ci-dessous).
Si la valeur est forte et donc la distance seuil grande, il y a toutes les chances que la particule reste dans le comportement où elle est collée au décor, épousant la moindre de ses formes

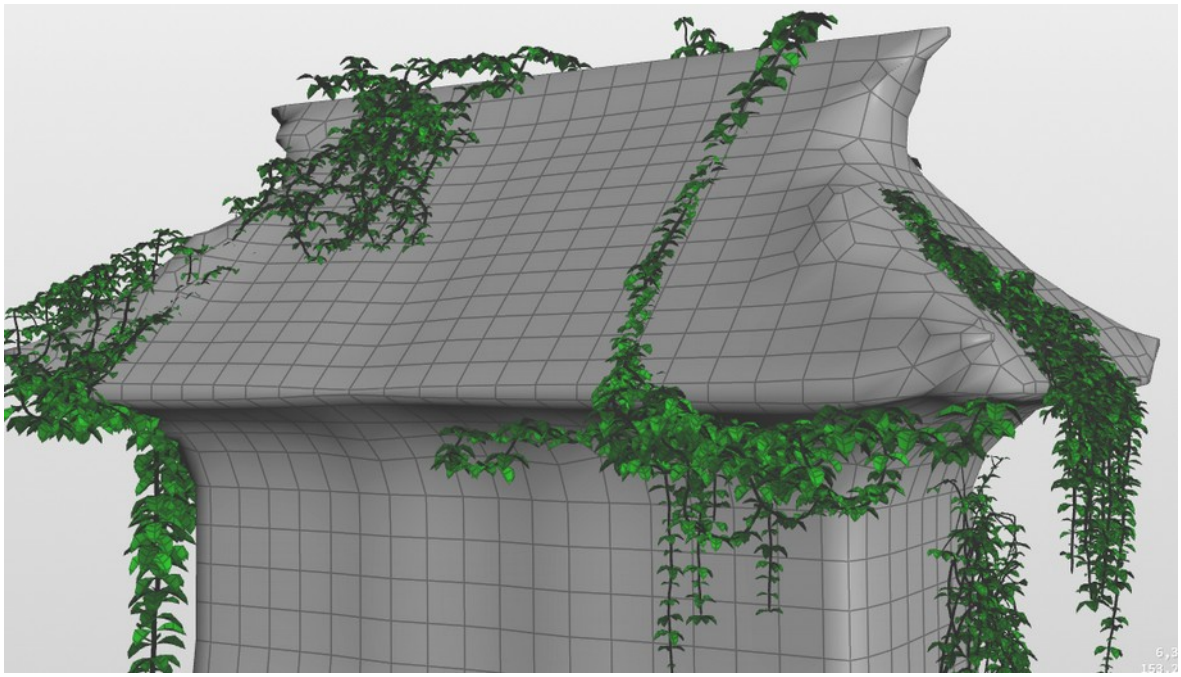


Illustration 66: Plantes grimpantes générées avec différents paramètres sur une même maison

En cas de contact avec le décor, la direction calculée de la particule est très impactée, car elle doit poursuivre dans la même direction tout en épousant la forme du décor. La formule utilisée vient directement de la thèse de Knutzen :

$$d' = d - n\,(n \cdot d)$$

ou bien :

la nouvelle direction d' = la direction précédente d
– la direction décor n * le produit scalaire entre (la direction précédente d et la direction décor n)

Cette formule est illustrée ci-contre.

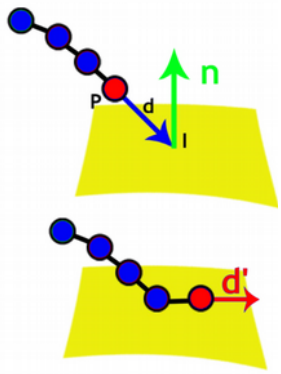


Illustration 67: Schéma explicatif de la formule de projection issue de la thèse de Knutzen

2.6.5.4 Embranchements et rayons

Comment produire un embranchement ?
Au fur et à mesure qu'une particule avance et construit une branche, un compteur s'incrémente. Quand ce compteur dépasse une valeur maximum (modifiable par l'utilisateur et appelée « longueur maximum de branche »), le compteur est remis à zéro et une toute nouvelle particule est émise au même endroit que la première. C'est la première particule d'une nouvelle branche. Sa direction de croissance a un angle particulier (que nous appellerons « angle de dispersion », modifiable par l'utilisateur) par rapport à la direction de la particule parente. L'influence de cet angle est démontrée dans l'illustration ci-dessous :

Illustration 68: Angle de dispersion de 70 (à gauche) à 30 degrés (milieu) à 5 degrés (à droite)

On peut aussi jouer sur un paramètre de probabilité de ramification : à chaque fois que la longueur maximum d'une branche est atteinte, au lieu de produire systématiquement une nouvelle particule, il y a une probabilité qu'il n'y ait pas de nouvelle branche et la branche parente continue son chemin. Si cette probabilité est nulle, il n'y aura jamais aucune nouvelle branche et la plante grimpante sera donc constituée d'une unique liane, quel que soit le nombre d'itérations de l'algorithme.

Bien évidemment, grâce à la technique découverte avec Valhalla, toutes les branches ont leurs UV correctement dépliés.

En botanique, la formule rigoureuse pour calculer les diamètres des branches d'un arbre en fonction de celui de la branche maîtresse est la formule de Leonardo, où $d^n = d_1^n + d_2^n$, avec n compris entre 2 et 3, d le diamètre de la branche sous l'embranchement et d_1 et d_2 le diamètre des branches au dessus.
Dans les faits, le calcul que j'utilise est beaucoup plus simple : à chaque nouvelle particule, son numéro d'identification augmente. Les premières particules correspondent aux premières

branches et auront potentiellement un plus gros rayon que les dernières qui sont apparues. Il s'agit donc simplement de faire correspondre les points avec des petits numéros aux plus gros rayons et inversement les plus grands numéros aux plus petits rayons. L'illusion dans notre cas est tout à fait satisfaisante.

Ci-dessous, l'outil dans sa dernière version a élaboré une plante grimpante envahissant une statue. Les feuilles sont peut-être un peu trop petites ou la statue vraiment très grande :

Illustration 69: Le lierre se propage librement sur la statue

2.6.5.5 L'interface

Des captures d'écran des trois onglets de l'interface de l'outil sont présentées ci-après. La lecture de mes différentes découvertes tout au long de la conception des différentes versions de l'outil devrait permettre la compréhension de tous les paramètres exposés.

Ici, l'utilisateur peut en premier lieu contrôler le nombre d'itérations de l'algorithme. Plus ce nombre sera important et plus les branches seront longues et auront de chances de conquérir une large surface.

Le *step size* peut aussi être modifié grâce à cet interface. Il correspond, ici, à la distance entre deux points d'une même branche. En botanique on parle d'internœud.

La *seed* générale peut être aussi modifiée, i. e. une petite valeur aléatoire présente à plusieurs niveaux de l'algorithme où le hasard a une importance (comme la direction du vecteur aléatoire, la probabilité d'embranchement ou encore l'orientation aléatoire des feuilles) et qui peut changer légèrement l'aspect du lierre tout en gardant les mêmes paramètres génériques.

Illustration 70: 1er onglet de l'interface avec tous les contrôles des branches (ou racines)

L'utilisateur a aussi la possibilité de positionner une feuille de chaque côté de la branche, en alternance ou bien d'en placer deux à chaque fois au même niveau, en cochant *double leaves* (cf. ci-dessous).

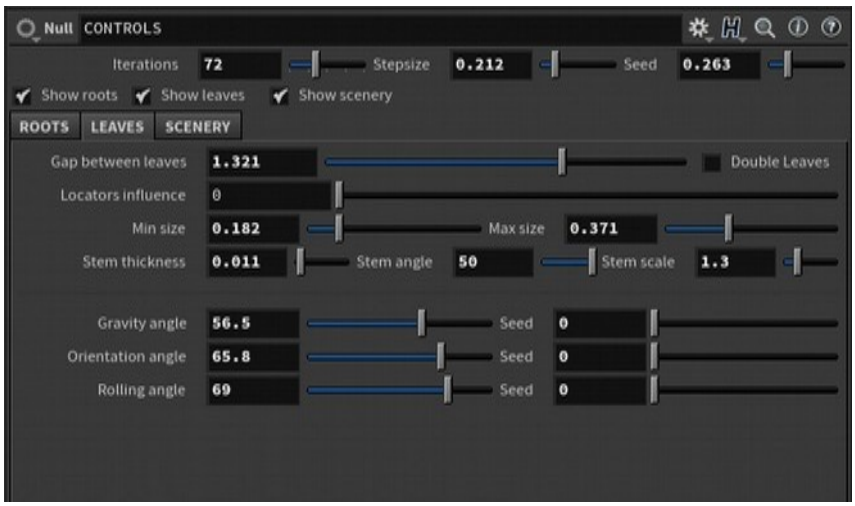


Illustration 71: 2ème onglet de l'interface avec le contrôle de toutes les feuilles

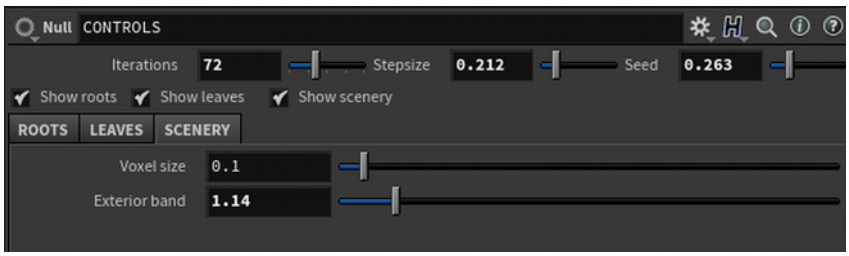


Illustration 72: Dernier onglet de l'interface avec les contrôles de la précision de conversion du décor en voxels (pixels 3D)

Les possibilités de personnalisation de l'interface des outils dans Houdini sont très limitées. Pour avoir de la couleur ou une meilleure disposition des curseurs, il faudrait faire du développement en Python...

2.6.5.6 Remarques

Cet outil semble répondre à toutes les consignes qui m'étaient imposées au départ. Contrairement aux précédentes versions, il peut recouvrir, avec cohérence, une grande surface d'un décor préexistant. Il permet aussi de modéliser un large panel de plantes grimpantes, le lierre certes, mais aussi d'autres espèces comme le rosier grimpant. Une vidéo-exposé de mes travaux sur la végétation procédurale dans Houdini le présente, en 2ème partie [50].

Par ailleurs, les plantes produites par l'outil m'ont souvent émerveillée, notamment dans le cas d'une unique branche-liane. Si une autre liane est proposée comme support, la liane générée peut s'enrouler comme dans la réalité autour d'une autre ! Il est fascinant de découvrir ce comportement « naturel » dans la pousse de la plante. Il est parfaitement logique puisque basé sur des forces qu'elle subit dans la nature. C'est très enthousiasmant de voir que quelque chose de cohérent est ainsi obtenu alors qu'à aucun moment de la conception ce cas de figure n'a été envisagé. C'est un peu la naissance d'une forme de vie autonome...! Mon outil produit donc un résultat alliant automatisation et contrôle, exactement les qualités visées.

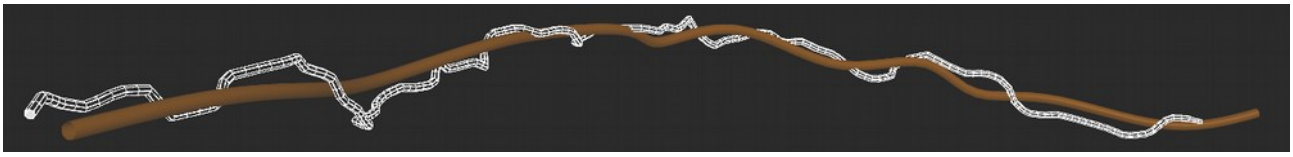


Illustration 73: Une liane qui s'enroule toute seule autour d'une liane support

Bien que l'outil apporte de grandes satisfactions et soit susceptible d'être utilisé sur un long métrage d'animation, il reste certains points à améliorer :

- éviter les collisions entre les branches, bien que cela soit peu visible et souvent masqué par les feuilles,
- supprimer également les collisions entre les feuilles ; un début d'algorithme a été conçu,
- rendre possible l'utilisation complète de l'outil par des modeleurs, que ce soit dans Houdini ou dans Maya via un *digital asset*,
- animer la croissance de la plante grimpante et de ses feuilles, bien que l'utilité dans un contexte de film soit limitée : excepté pour un plan très précis, il est rarement nécessaire dans un décor de voir du lierre pousser en accéléré.
- avoir la possibilité de créer une plante grimpante « en massif », comme sur la photographie ci-après :

Illustration 74: Plante grimpante qui présente des volumes indépendants du décor dessous

Il reste beaucoup de travail avant d'avoir un outil capable de modéliser vraiment tous les styles de plantes grimpantes, tant la nature est riche et diverse.

Finalement, c'est cette dernière version de l'outil qui a ma préférence sur toutes celles conçues précédemment. Dans les faits, les plantes générées restent assez difficiles à contrôler, principalement à cause de l'ajout de la force aléatoire. Elles ont alors un véritable comportement de plante grimpante envahissante. Leur forme générale est aussi très dépendante de celle du décor, si celui-ci vient à changer, même de manière infime, leur aspect initial sera fortement modifié et beaucoup de réglages devront être faits avant de le retrouver.

Conclusion

Cette année de M2 m'a permis d'explorer la végétation procédurale, qui me tenait particulièrement à cœur. C'est en expérimentant que mon intérêt s'est cristallisé sur le cas particulier des plantes grimpantes et rampantes. Mes recherches, avant de considérer le domaine du virtuel, ont donc d'abord porté sur la botanique en général ; la quantité de connaissances et de vocabulaire emmagasinée pourrait servir à la rédaction d'un second mémoire. Il m'a donc été difficile de ne pas rentrer dans tout le détail de ce que j'ai appris.

La réalisation de mes différents projets personnels et de toutes les versions de l'outil de génération de plante grimpante m'ont fait jongler entre différentes méthodes de modélisation. Certaines permettent un contrôle artistique assez poussé (notamment la version 3 de l'outil) tandis que d'autres semblent avoir leur vie propre (le projet Arabesques). Notons que, dans Houdini, puisque les éléments générés sont avant tout des courbes, elles peuvent dans tous les cas, après le processus de génération, être modifiées manuellement pour obtenir l'effet précis souhaité. Et c'est sans doute cette alliance du procédural et du manuel qui permet d'obtenir les meilleurs résultats. Il faut toutefois être conscient que la modélisation procédurale, en général, prend du temps : pour des cas précis, le jeu n'en vaut pas la chandelle et la modélisation manuelle reste la meilleure solution, surtout dans un contexte de production avec un planning serré.

Ma compréhension des mathématiques derrière les questions de projection et de rotation dans l'espace ont grandement servi dans la résolution de problèmes techniques pointus, rencontrés au cours de mes projets.

L'aléatoire généré par l'ordinateur a une place particulière dans mon travail. L'intérêt de la modélisation procédurale est de construire un même « concept » avec des variations. Ces variations sont produites par un hasard très contrôlé : les bornes entre lesquelles il peut s'exprimer doivent être clairement établies, sous peine d'obtenir un résultat brouillon. Une définition toujours plus précise de ces bornes ont permis à mes modèles de gagner en pertinence.

Remarquons que s'inspirer de la botanique, de la réalité de la nature et de notre monde physique donne les meilleurs résultats, d'après mon expérience. En effet, au moment où la pousse a cessé d'être considérée comme processus de base de modélisation, au début de la version 3 de l'outil, ce dernier n'a plus fonctionné convenablement. La plante virtuelle a donc besoin de croître en prenant en compte à chaque instant son support juste sous elle et l'état actuel de son développement. C'est donc son mouvement de croissance qui permet de lui assurer une crédibilité botanique. D'ailleurs, « ramper » et « grimper » sont des verbes d'action qui traduisent le mouvement qu'elles font pour se construire.

Une autre similitude entre mon travail et la nature peut aussi être trouvée dans le processus même de conception : mes projets et surtout mes outils sont une suite d'essais et d'erreurs, et une sélection naturelle s'est opérée pour ne garder que ce qui marche, ce qui peut survivre.

Il n'a pas vraiment été question de rendu lors de mes expérimentations et c'est un peu un regret. Mon temps n'a été dédié qu'à la technique et il n'en restait pas pour créer d'élégantes images de châteaux romantiques envahis de végétation ou d'environnements mystérieux à l'abandon, où la nature aurait repris ses droits.

Perspectives

Travailler sur la végétation virtuelle m'a beaucoup plu et je n'ai fait qu'en effleurer le sujet. La nature est si vaste et peut être vue elle-même comme un gigantesque outil de modélisation procédurale : certaines espèces ne se distinguent que par la taille des tiges ou de certaines parties de la feuille ou bien encore par la présence de petites excroissances à certains endroits et pas à d'autres… La nature s'amuse à modifier tous ces paramètres pour développer de nouvelles espèces, à l'infini ou presque… Il y a tant de végétaux bizarres sur Terre dont nous n'avons même pas idée ! Concevoir un outil un peu complet qui ne générerait que des fleurs procédurales par exemple prendrait beaucoup de temps tant les variations possibles sont immenses. Le travail serait cependant passionnant et le résultat très agréable à regarder.

Un autre sujet d'étude, assez présent dans le domaine de la végétation procédurale, est le biome, c'est-à-dire, dans notre contexte, la répartition des végétaux les uns par rapport aux autres dans un environnement virtuel et les relations qu'ils entretiennent. En effet, certaines espèces s'épanouissent au contact d'autres ou peuvent empêcher le développement de plantes plus fragiles. Le lierre, par exemple, peut potentiellement étouffer et tuer les arbres sur lesquels il s'accroche. Comment générer automatiquement tout un paysage virtuel cohérent sans s'écarter des règles de la botanique?

A l'issue de ce mémoire, puisse le lecteur partager à présent mon enthousiasme face à la beauté algorithmique de la structure des plantes et comprendre cette fascination qui m'a amenée à vouloir les modéliser.

Bibliographie

(1) S. Ulam, 1966, Patterns of growth of figures: Mathematical aspects. In G. Kepes, editor, Module, Proportion, Symmetry, Rhythm, pages 64–74. Braziller, New York,.

(2) P. S. Stevens, 1974, Patterns in nature. Little, Brown and Co., Boston,

(3) J. von Neumann, 1966, Theory of self-reproducing automata. University of Illinois Press, Urbana,. Edited by A. W. Burks.

(4) N. Greene. 1989, Voxel space automata: Modeling with stochastic growth processes in voxel space. Proceedings of SIGGRAPH '89 (Boston, Mass., July 31-August 4, 1989), in Computer Graphics 23,4 (August 1989), pages 175–184, ACM SIGGRAPH, New York,

(5) Anastacio, Fabricio, Mario Costa Sousa, Faramarz Samavati, et Joaquim A. Jorge. 2006. « Modeling Plant Structures Using Concept Sketches ». In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering - NPAR '06*, 105. Annecy, France: ACM Press. https://doi.org/10.1145/1124728.1124746

(6) Feriani, Marta. 2017 « Art-Directing Procedural Vegetation in Houdini Using a Space Colonization Algorithm », Bournemouth University , 66.

(7) Ferrysienanda, Edward. 2014. « A STUDY OF PROCEDURAL GROWTH ALGORITHMS AND ITS IMPLEMENTATION IN HOUDINI », Bournemouth University 26

(8) Guo, Jianwei, Zhanglin Cheng, Shibiao Xu, et Xiaopeng Zhang. 2017. « Realistic Procedural Plant Modeling Guided by 3D Point Cloud ». In *ACM SIGGRAPH 2017 Posters on - SIGGRAPH '17*, 1-2. Los Angeles, California: ACM Press. https://doi.org/10.1145/3102163.3102193.

(9) Keim, Hans, Maryann Simmons, Daniel Teece, Jared Reisweber, et Sara Drakeley. 2016. « Art-Directable Procedural Vegetation in Disney's Zootopia ». In *ACM SIGGRAPH 2016 Talks on - SIGGRAPH '16*, 1-2. Anaheim, California: ACM Press. https://doi.org/10.1145/2897839.2927469.

(10) Knutzen, Johan. 2009. « Generating Climbing Plants Using L-Systems ». Chalmers University of Technology, University of Gothenburg.

(11) Kuruc, Matt, Nancy Tsang, et Tom Nettleship. 2014. « Art-Directable Canopies in Pixar's Vegetation Pipeline ». In *ACM SIGGRAPH 2014 Talks on - SIGGRAPH '14*, 1-1. Vancouver, Canada: ACM Press. https://doi.org/10.1145/2614106.2614162.

(12) Mandelbrot, Benoît B. édition 2010. *Les objets fractals [Texte imprimé] : forme, hasard et dimension*. Champs sciences: 301. Flammarion.

(13) Palubicki, Wojciech, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, et Przemyslaw Prusinkiewicz. 2009. « Self-Organizing Tree Models for Image Synthesis ». *ACM Transactions on Graphics* 28 (3): 1. https://doi.org/10.1145/1531326.1531364.

(14) Prusinkiewicz, Przemyslaw, et Aristid Lindenmayer. 1990. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag. http://algorithmicbotany.org/papers/abop/abop.pdf.

(15) Runions, Adam, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, et Przemyslaw Prusinkiewicz. 2005. « Modeling and visualization of leaf venation patterns ». *ACM Transactions on Graphics* 24 (3): 702-11.

(16) Runions, Adam, Brendan Lane, et Przemyslaw Prusinkiewicz. s. d. « Modeling Trees with a Space Colonization Algorithm », 9.

(17) Shek, Arthur, Dylan Lacewell, Andrew Selle, Daniel Teece, et Tom Thompson. 2010. « Art-Directing Disney's *Tangled* Procedural Trees ». In *ACM SIGGRAPH 2010 Talks on - SIGGRAPH '10*, 1. Los Angeles, California: ACM Press. https://doi.org/10.1145/1837026.1837095.

Webographie

(18) « Artist Highlight: Guilherme Rabello – SpeedTree ». s. d. Consulté le 28 avril 2019. https://store.speedtree.com/artist-highlight-guilherme-rabellos-vine/.

(19) Chevry, Marc. 2019. *IvyTaming 1.0 Teaser Video*. https://vimeo.com/312090275.

(20) D, Glen. 2017. *Procedural Tube with UVs*. https://vimeo.com/210199329.

(21) Dalvi, Rohan. 2015. *Creating animated ivy in Houdini*. https://vimeo.com/128296265.

(22) ———. 2019. « Creating Isolines ». YouTube. 7 mars 2019. https://www.youtube.com/channel/UCfGN9OlJF9KakdyUK04M3cA.

(23) Entagma. 2016a. *Differential Line Growth - The Quick Way*. https://vimeo.com/169438763.

(24) ———. 2016b. *Reaction Diffusion Part I: Theory*. https://vimeo.com/170073061.

(25) ———. 2016c. *Coloring Strands (Point Advection & Hair Shading)*. https://vimeo.com/184145249.

(26) ———. 2016d. *Special Guest - Ben Watts: Shortest Path Growth*. https://vimeo.com/197596529.

(27) ———. 2017a. *Packing The Torus*. https://vimeo.com/210226436.

(28) ———. 2017b. *VEX in Houdini: Space Colonization*. https://vimeo.com/231315378.

(29) ———. 2018. *TD Essentials: Parallel Transport*. https://vimeo.com/251091418.

(30) Estela, Matt. s. d. « HoudiniDops - cgwiki ». Consulté le 12 avril 2019. http://www.tokeru.com/cgwiki/index.php?title=HoudiniDops.

(31) F12. s. d. « 3D Tree Growing Software - The Grove ». Consulté le 11 avril 2019. https://www.thegrove3d.com/.

(32) Fiedler, Simon. 2017. *Building an ivy growth solver in Houdini (Part 1)*. https://vimeo.com/212483416.

(33) Gaal, Stijn van. s. d. « Ivy, shaders and more fun stuff ». ArtStation. Consulté le 11 avril 2019. https://www.artstation.com/stijnvangaal/blog/m2O2/ivy-shaders-and-more-fun-stuff?fbclid=IwAR2uiURBy9pe5BN5R1N818s5HrRRE3WROi70hCi2GiUF9O5LssxCGuUBeyM.

(34) Game, Tutor. s. d. « Procedural Hanging Cables | SideFX ». Consulté le 16 avril 2019. https://www.sidefx.com/tutorials/procedural-hanging-cables/.

(35) Junichiro Horikawa. s. d. *[Houdini Tutorial] 0018 Diffusion-Limited Aggregation (Slow ver.)*. Consulté le 28 avril 2019. https://www.youtube.com/watch?time_continue=116&v=ttvVYqjs9XE.

(36) Lobzhanide, Berika. s. d. « Houdini Particle flow on surface Tutorial ». Vimeo. Consulté le 12 avril 2019. https://vimeo.com/239928598.

(37) « Maya Paint Effects | Maya 2018 | Autodesk Knowledge Network ». s. d. Consulté le 28 avril 2019. https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-CharEffEnvBuild/files/GUID-890E22CA-74F9-4608-AC35-E1E0B1768168-htm.html.

(38) McKenzie, Lauren. 2019. « CGMA Stundet Project: Greenhouse ». *80.lv* (blog). 15 février 2019. https://80.lv/articles/005cg-cgma-stundet-project-greenhouse/.

(39) Luft, Thomas. s. d. « An Ivy Generator ». Consulté le 28 avril 2019. http://graphics.uni-konstanz.de/~luft/ivy_generator/.

(40) Opara, Anastasia. 2017. « Procedural Realistic Tree R&D ». Vimeo. 11 août 2017. https://vimeo.com/229252148.

(41) OverQuantum. s. d. *Panspermia (1990) by Karl Sims - 720p*. Consulté le 28 avril 2019. https://www.youtube.com/watch?v=ZRHH6Y6iggo.

(42) Pepe, Alessandro. 2013. « FX Thinking: stuff growing and crawling - Hyphae - OTL version ». *FX Thinking* (blog). 20 mai 2013. https://pepefx.blogspot.com/2013/05/a-few-days-ago-i-was-reading-this.html.

(43) Pérez, Danièle. 2015. « La flore dans l'art ». Arts Plastiques. 5 juin 2015. https://perezartsplastiques.com/2015/06/05/la-flore-dans-lart/.

(44) « Planetside Software ». s. d. Planetside Software. Consulté le 30 avril 2019. https://planetside.co.uk/.

(45) SideFX. s. d. « For-each copy tutorial ». Consulté le 14 avril 2019. http://www.sidefx.com/docs/houdini/copy/tutorial_stamping.html.

(46) « SpeedTree – 3d Vegetation Modeling and Middleware ». s. d. Consulté le 28 avril 2019. https://store.speedtree.com/.

(47) « Spirale d'or ». s. d. Consulté le 1 mai 2019. https://www.mathcurve.com/courbes2d/logarithmic/spiraledor.shtml.

(48) System, Nervous. 2011. *Hyphae - growth process diagram in 2D*. https://vimeo.com/25604611.

(49) « Xfrog - 3D Trees and Plants for CG Artists ». s. d. Xfrog Inc. Consulté le 30 avril 2019. http://xfrog.com.

(50) Cabé, Fanny. 2019. « Procedural Vegetation Studies with Houdini ». Vimeo. Consulté le 4 mai 2019. https://vimeo.com/326596772.

Index des illustrations

Illustration 1: Premier plan de TheAdventures of Andre and Wally B (1986)....2
Illustration 2: Scène extraite du Jungle Book (2016) où seul Mowgli n'est pas virtuel....2
Illustration 3: Organic Architecture par Greene (1989)....3
Illustration 4: Le chou romanesco présente une structure fractale....3
Illustration 5: Interface de Speedtree....4
Illustration 6: Différents paramétrages pour l'algorithme de nervation d'une feuille....5
Illustration 7: Résultat graphique de L-systèmes de plantes fractales....8
Illustration 8: Exemples d'arbres créés en modifiant un seul paramètre dans le système de Palubicki....9
Illustration 9: Schéma explicatif des 5 étapes de l'algorithme de nervation....10
Illustration 10 : L'algorithme de nervation en 3D....12
Illustration 11: Schéma de la divine proportion ou nombre d'or....12
Illustration 12: l'angle d'or....13
Illustration 13: Cœur d'une fleur de tournesol....13
Illustration 14: Spirale d'or....13
Illustration 15: Image de fin du projet Moss and Flowers (2017)....14
Illustration 16: Le projet Evolve (2018)....15
Illustration 17: Graphe pondéré....16
Illustration 18: Image du début du projet Moss and flowers (2017)....16
Illustration 19: De gauche à droite : le buste de base, le buste avec réduction aléatoire de polygones et sélection du point de départ, et enfin le même buste avec sélection des points d'arrivée....17
Illustration 21: De gauche à droite : les courbes en sortie du shortest path, puis les mêmes lissées et légèrement décollées et enfin les mêmes recouvertes d'un deuxième système de courbes....18
Illustration 20: le résultat rendu (et animé)....18
Illustration 22: Toute première version du buisson de ronces....19
Illustration 23: Photo d'un buisson de ronces....20
Illustration 24: A gauche : les volumes imbriqués qui servent de support à l'algorithme de nervation. Au milieu : Le plus grand volume recouvert de ronces produites par l'algorithme. A droite : tous les volumes de ronces imbriqués....21
Illustration 25: Répartition aléatoire d'arcs de cercles sur les derniers volumes de ronces....21
Illustration 26: Modélisation procédurale quasi finale du buisson de ronces....22
Illustration 27: Couloir d'une caverne générée procéduralement....23
Illustration 28: Lignes créées suite à la simulation de particules....24
Illustration 29: Lianes-tubes générées avec des UV correctement dépliés, puisque chaque carré de la texture d'exemple reste bien carré....25
Illustration 30: Rendu final des lianes sur les parois de la caverne dans Unreal Engine....25
Illustration 31: Les particules (et donc les lignes-lianes en blanc) peinent à recouvrir toutes les surfaces de cavernes (en grisé)....26
Illustration 32: Calcul du système de particules sur 40 images (gauche), 140 images (milieu) et de 240 images (droite)....28
Illustration 33: Dernière image du rendu animé du projet Arabesques....28
Illustration 34: Arabesques sur support plat....29
Illustration 35: Photographies de feuilles de lierre....31
Illustration 36: Photographies de structure de lierre....31
Illustration 37: Application de l'algorithme de nervation comme structure de lierre....32
Illustration 38: Visualisation du champ de force orienté....33
Illustration 39: Les normales des points sur une modélisation de tête de cochon....33
Illustration 40: Comparaison des méthodes de projection....34
Illustration 41: Feuilles réparties dans n'importe quel sens le long des branches....34
Illustration 42: Orientation souhaitée des feuilles le long du décor....34
Illustration 43: Repère orthonormé de la feuille....34
Illustration 44: Interface dans Houdini de la 1ère version d'outil de plante grimpante....35
Illustration 45:A gauche, direction de points calculée par proximité avec les courbes utilisateur. A droite, direction des points calculée par interpolation linéaire....36

Illustration 46: Tronc et branches secondaires sur mur orienté....37
Illustration 47: à gauche les branches suivent la bitangente, à droite elles suivant la courbe principale....37
Illustration 48: Branches bruitées et plaquées sur le mur....38
Illustration 49: Forme de la rampe qui détermine comment les branches vont se coller au mur....38
Illustration 50: Un tronc, des branches et des brindilles....38
Illustration 51: A gauche, modifications manuelles dans les directions de pousse, à droite l'impact sur la structure globale....39
Illustration 52: Différentes variations de courbes-branches esthétiques....40
Illustration 53: Outil simplifié....41
Illustration 54: Outil avec les feuilles par défaut....41
Illustration 55: Variation de densité de feuilles grâce aux metaballs....42
Illustration 56: Capture d'écran de l'outil présenté au modeleur....43
Illustration 57: Problèmes de projection des branches sur un décor biscornu....43
Illustration 58: Rendu basique de la structure seule (à gauche) puis avec feuilles (à droite) du lierre sur un décor....44
Illustration 59: Captures d'écran issues de la vidéo de présentation de l'outil Ivy Taming....45
Illustration 60: Prototypage de l'outil qui se présente comme un L-système....46
Illustration 61: Influence de la gravité : d'aucune (gauche), à moyenne (milieu) à forte (à droite)....47
Illustration 62: Influence très forte de deux guides tracés par l'utilisateur (courbes rouges à peine visible)...47
Illustration 63: influence du vecteur aléatoire : de nul (à gauche) à très fort (à droite)....47
Illustration 64: influence de l'attraction du décor (faible à gauche et forte au milieu). A droite, décollement bruité des lianes par rapport au décor....48
Illustration 65: Schéma explicatif en 2D des différences entre les deux comportements possibles d'une particule....48
Illustration 66: Plantes grimpantes générées avec différents paramètres sur une même maison....49
Illustration 67: Schéma explicatif de la formule de projection issue de la thèse de Knutzen....49
Illustration 68: Angle de dispersion de 70 (à gauche) à 30 degrés (milieu) à 5 degrés (à droite)....50
Illustration 69: Le lierre se propage librement sur la statue....51
Illustration 70: 1er onglet de l'interface avec tous les contrôles des branches (ou racines)....52
Illustration 71: 2ème onglet de l'interface avec le contrôle de toutes les feuilles....52
Illustration 72: Dernier onglet de l'interface avec les contrôles de la précision de conversion du décor en voxels (pixels 3D)....53
Illustration 73: Une liane qui s'enroule toute seule autour d'une liane support....53
Illustration 74: Plante grimpante qui présente des volumes indépendants du décor dessous....54