

ÉCOLE DE TECHNOLOGIE SUPERIEURE
UNIVERSITE DU QUEBEC

RAPPORT DE PROJET PRESENTE A
L'ÉCOLE DE TECHNOLOGIE SUPERIEURE

COMME EXIGENCE PARTIELLE
A L'OBTENTION DE LA
MAITRISE EN GENIE CONCENTRATION TECHNOLOGIE DE L'INFORMATION
M. ING

PAR
Yakhoub DRAME

CONCEPTION ET RÉALISATION D'UNE APPROCHE DE DÉGRADATION
INTERACTIVE ET ARTISTIQUE D'UN ENVIRONNEMENT VIRTUEL

MONTREAL, LE 9 JUILLET 2013



Yakhoub DRAME, 2013



Cette licence [Creative Commons](#) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRESENTATION DU JURY

CE RAPPORT DE PROJET A ETE EVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Eric Paquette, directeur de projet

Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. David Labb  , jury

Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

REMERCIEMENTS

Je tiens à remercier M. Paquette pour la supervision qu'il m'a donnée tout au long du projet. Je souhaiterais également remercier l'équipe du projet du Métalab de la Société des arts technologiques (SAT) qui m'a aidé lors de la réalisation du projet, notamment Luc Courchesne, Zack Settel, Emmanuel Durand et Alexandre Quessy. Enfin je voudrais remercier l'École de technologie supérieure sans laquelle je n'aurais pu décrocher l'opportunité de mener ce projet et suivre le cursus de ma maîtrise.

CONCEPTION ET RÉALISATION D'UNE APPROCHE DE DÉGRADATION INTERACTIVE ET ARTISTIQUE D'UN ENVIRONNEMENT VIRTUEL

Yakhoub DRAME

RÉSUMÉ

L'éphémère et la présence humaine sont des notions dont la combinaison est encore peu explorée dans l'univers des environnements virtuels immersifs et collaboratifs. Leur combinaison garantit un renforcement de la notion d'existence de l'homme et de celle de l'autre dans de tels environnements. Le Métalab de la Société des arts technologiques s'est intéressé à cette problématique sous la forme du projet de recherche création « environnements susceptibles ». Son but est donc d'étudier ces notions d'éphémère et de présence de l'homme dans de tels environnements en permettant aux utilisateurs de dégrader l'environnement pour y laisser des traces éphémères et d'interagir entre eux. Ce rapport fait état d'un prototype s'inscrivant de ce projet. Ce prototype prend la forme d'une application graphique où un utilisateur peut déposer en temps réel des empreintes de pas dans une scène 3D à l'aide de la souris. Cette application est constituée d'une interface utilisateur offrant les possibilités d'afficher des empreintes de façon ponctuelle ou de dessiner des trajets représentant des déplacements en traçant des chemins sur le sol. L'application tire profit de techniques d'infographie, tels que le *normal mapping* et de l'utilisation de nuanceurs, à titre de performance et de qualité visuelle.

DESIGN AND IMPLEMENTATION OF AN INTERACTIVE AND ARTISTIC APPROACH FOR VIRTUAL ENVIRONMENT DEGRADATION

Yakhoub DRAME

ABSTRACT

Ephemeral effects and human presence are notions which, brought together, are not fully explored in the virtual immersive and collaborative environments. Their alliance is a solid foundation for the notion of human existence and that of the others in those environments as well. The Métalab of the Society of technologic arts tries to address this issue with their research-creation project called « *environnements susceptibles* ». Its main goal is to study ephemeral and human presence in such environments by enabling the user to affect the environment, leave ephemeral traces and interact with other users. This report accounts for a prototype implemented in the scope of this project. This prototype is a real time graphics application allowing the user to put footsteps in a 3D scene. It is made up of an interface enabling to draw footsteps one by one or to draw along paths on the ground. The application takes advantage of computer graphics techniques such as *normal mapping* and shaders to reach efficiency and visual quality.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 État de l'art	7
1.1 Détérioration d'environnement dans la recherche	7
1.1.1 Dégradation par synthèses de texture	7
1.1.2 Vieillissement de scène	8
1.1.3 Saleté et poussière	9
1.1.4 Corrosion	9
1.1.5 Craquelure et écaillement de la peinture	10
1.1.6 Phénomènes d'écoulement d'eau	10
1.1.7 Égratignures, rayures, impact	11
1.2 Outils de développement.....	12
1.2.1 Librairie d'affichage graphique.....	12
1.2.2 Librairie de graphe de scène.....	13
1.2.3 Librairie d'environnement collaboratif	13
1.2.4 Librairies d'interfaces graphiques	14
1.3 Problématiques.....	14
1.3.1 Application dans le contexte du réalisme.....	15
1.3.2 Emploi en tant que symbole de « l'éphémère »	15
1.3.3 Notions de présence et d'interaction	15
1.3.4 Développement de l'application	16
1.3.5 Collaboration entre les intervenants	16
CHAPITRE 2 OBJECTIFS	17
2.1 Processus technologique	17
2.1.1 Modélisation de l'éphémère visuellement sous forme de traces.....	17
2.1.2 Moyens d'interaction avec l'utilisateur	17
2.1.3 Gestion des contraintes d'affichage, de stockage et de temps réel	17

2.2 Processus d'ingénierie	18
2.2.1 Travail en équipe	18
2.2.2 Intégration du projet de maîtrise	18
2.2.3 Adaptation à un cahier des charges changeant.....	19
CHAPITRE 3 MÉTHODOLOGIE	21
3.1 Étapes de la réalisation.....	21
3.1.1 Développements techniques	21
3.1.2 Exploration artistique	22
3.2 Équipe de la Société des arts technologiques	23
3.2.1 Directeurs de projets.....	23
3.2.2 Développeurs.....	24
3.3 Processus technologiques.....	24
3.3.1 Études de travaux de recherche sur la dégradation	24
3.3.2 Mise au point de méthodes d'interaction intuitives	24
3.3.3 Utilisation de nuanceurs	25
3.4 Processus d'ingénierie	25
3.4.1 Analyse des exigences et définition des livrables	25
3.4.2 Mises au point lors de réunions avec l'équipe	25
3.4.3 Gestion de configuration	26
3.4.4 Arrimage avec d'autres sous-projets.....	26
CHAPITRE 4 CONCEPTION.....	27
4.1 Mise en perspective des travaux de recherche	27
4.1.1 Dégradation d'un environnement par un des utilisateurs	28
4.1.2 Dégradation par marquage d'empreintes de pas	29
4.1.3 Rapport entre les dégradations et l'interaction.....	31
4.1.4 Techniques à utiliser dans le contexte du temps réel	32
4.1.5 Utilisation dans le contexte d'application distribuée	34
4.1.6 Conclusion de la mise en perspective.....	35
4.2 Premier prototype.....	39
4.2.1 Analyse	39
4.2.2 Conception.....	40

4.2.3	Modélisation des dégradations	40
4.2.4	Type d'interaction	44
4.2.5	Point négatifs et positifs	45
4.3	Deuxième prototype.....	46
4.3.1	Analyse	46
4.3.2	Conception.....	49
4.3.3	Technologies utilisées et remplacées	51
4.3.5	Amélioration du stockage de l'information	55
4.3.7	Techniques d'infographie mises de l'avant	64
4.3.8	Point négatifs et positifs	66
	CHAPITRE 5 DISCUSSION	69
5.1	Difficultés surmontées	69
5.2	Valeur ajoutée de l'application.....	70
5.2.1	Travail sur le stockage d'information par les textures	70
5.2.2	Précision des empreintes	72
5.2.3	Plusieurs méthodes d'interaction	74
5.2.4	<i>Normal mapping</i>	74
5.3	Travail plus accentué sur les fonctionnalités que sur le réalisme graphique	75
5.4	Discontinuités légères dans le déplacement de la caméra	75
5.5	Lecture de la texture d'informations d'empreintes de pas.....	76
5.6	Économie de mémoire dans la texture d'informations	76
5.7	Aspect modulable des listes d'empreintes par partitions de la scène	77
5.8	Flux de données entre l'interface et l'environnement.....	78
5.9	Changement du cahier des charges	79
5.9.1	Mise de côté de certains aspects du projet	79
5.9.2	Simplification d'autres aspects	79
5.10	Relation avec l'équipe du projet « environnements susceptibles ».....	80
5.10.1	Côté collaborateur	80
5.10.2	Côté client.....	81
	CONCLUSION 83	
	RECOMMANDATIONS	85

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES	90
--	----

LISTE DES TABLEAUX

	Page
Table 4-1 Tableau récapitulatif des articles de recherche.....	27
Table 4-2 Dégradations provoquées par des utilisateurs	29
Table 4-3 Rapprochement entre les dégradations des travaux et les empreintes de pas.....	30
Table 4-4 L'interaction dans les articles de recherche	32
Table 4-5 Le temps réel dans les articles de recherche.....	33
Table 4-6 Les articles de recherche et les environnements distribués	34
Table 4-7 Tableau de mise en perspective des travaux.....	35
Table 4-8 Sélection des articles répondant le mieux aux objectifs du projet de maîtrise	37

LISTE DES FIGURES

	Page
Figure 4-1 Aperçu de l'environnement du premier prototype	40
Figure 4-2 Diagramme UML (premier prototype).....	41
Figure 4-3 Images de dégradations	42
Figure 4-4 Environnement couvert de dégradations	42
Figure 4-5 Diagrammes de séquence de l'ajout (a) et de l'affichage (b) d'une dégradation....	43
Figure 4-6 Aliassage et débordement de dégradation.....	46
Figure 4-7 Interfaces des deux prototypes	47
Figure 4-8 Diagramme UML (deuxième prototype)	48
Figure 4-9 Diagramme de séquence d'ajout d'empreintes (deuxième prototype).....	49
Figure 4-10 Diagramme de séquence de l'affichage des empreintes (deuxième prototype)...	50
Figure 4-11 Comparaison de fragments dans une empreinte de pas	52
Figure 4-12 Variation des paramètres d'une empreinte	53
Figure 4-13 Application d'empreintes de formes, orientations, tailles, et types différents.....	54
Figure 4-14 Représentation d'une empreinte dans un texel.....	55
Figure 4-15 Représentation d'un tableau pour 4 partitions et 11 empreintes.....	57
Figure 4-16 Texture associée au tableau de la Figure 4-15	58
Figure 4-17 Interface sous le mode d'interaction par clic	61
Figure 4-18 Orientation d'empreintes vers l'extérieur et l'intérieur du chemin	62
Figure 4-19 Représentation de chemins d'empreintes	63
Figure 4-20 Vecteurs de trajectoire.....	63
Figure 4-21 Vecteurs de position, direction et orientation	63

Figure 4-22 Série de pas et de vecteurs	64
Figure 4-23 Scène sans et avec <i>normal mapping</i>	65
Figure 5-1 Évolution du nombre d'images par secondes par le nombre de partitions	70
Figure 5-2 Évolution du nombre moyen d'empreinte par partition par le nombre de partitions	71
Figure 5-3 Évolution du nombre d'images par secondes en fonction du rayon des empreintes	72
Figure 5-4 Zoom sur des empreintes de pas	73
Figure 5-5 Empreintes dont la taille est supérieure à celle d'une partition	77

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	<i>Application Programming Interface.</i> Interface de programmation
BRDF	<i>Bidirectional Reflectance Distribution Function.</i> Fonction de distribution de réflectance bidirectionnelle
FPS	<i>Frame Per Second.</i> Nombre d'images par seconde
FPS	<i>First-Person Shooter.</i> Jeu de tir à la première personne
Git	Logiciel de gestion de version décentralisé
GLSL	Langage de programmation des nuanceurs
libQGLViewer	Librairie C++ basée sur QT facilitant la création de visualisateur 3D en OpenGL
Métalab	Laboratoire de recherche et de développement en art technologique de la Société des arts technologiques
OpenGL	Environnement de développement d'application graphique 2D et 3D
OSG	<i>OpenSceneGraph.</i> Interface de développement d'application 3D en code source ouvert
SAT	Société des arts technologiques
SPIN	Suite d'outils et librairies pour le développement d'environnements virtuels distribués et de visualisation 3D
Redmine	Application web en code source ouvert de gestion complète de projet
VBO	<i>Vertex Buffer Object.</i> Objet tampon de vertex

INTRODUCTION

« Les ronds dans l'eau produits par un jet de pierre, les pas dans le sable, les feuilles d'automne qui volent à notre passage, les post-it qu'on attache ici et là, les objets perdus et les déchets, les marques sur les planchers, la fonte des neiges, l'érosion et l'usure des pierres, sont autant de manifestations des forces naturelles et humaines qui transforment le monde physique. Ces traces que nous laissons, à travers un large spectre spatial et temporel, contribuent à la construction de notre identité et témoignent de notre présence au monde. Elles peuvent être éphémères comme un parfum, un bruit, ou durables comme une inscription dans la pierre, l'usure d'un plancher de bois, une photo... »

Cette définition de l'éphémère décrite par Luc Courschesne s'inscrit dans une volonté d'étudier l'interaction entre l'homme et son environnement. Les traces, égratignures ou usures que ce dernier laisse derrière lui sont des phénomènes observé au quotidien, renforçant le sens de présence et de celle de l'autre.

Dans l'univers de l'infographie, cette notion contribue à améliorer l'expérience d'immersion.

La réalité virtuelle immersive en particulier a pour objectif de plonger un individu dans un environnement virtuel artificiel où il se sent immergé autant que dans la réalité. Elle consiste en la simulation interactive immersive visuelle, sonore, voire haptique d'environnements réels ou imaginaires. Cette technologie se manifeste aujourd'hui surtout sous formes de projets artistiques et de recherche. Elle est mise en place grâce à des interfaces innovantes telles que des salles immersives sphériques ou cubiques ou d'écrans stéréoscopiques. Les domaines d'applications sont vastes : la formation par simulateur, les jeux vidéos, la narration visuelle et surtout les arts numériques.

Le projet de maîtrise s'inscrit aussi dans le domaine des environnements virtuels collaboratifs. Ces environnements mettent en place des outils permettant à ses utilisateurs d'interagir entre eux par le biais de communication homme – machine – homme. Dans de tels espaces numériques, les utilisateurs peuvent être représentés par des images 2D voire 3D dotées d'expressions, ou même d'avatars à leur image. Les environnements virtuels

collaboratifs ont beaucoup d'application dans les domaines de la médecine, de l'architecture, ou du divertissement.

Cependant un manque à pallier dans ces univers immersif collaboratif est le sentiment d'être là, et le sentiment de sentir la présence de l'autre et sa trace. Les environnement virtuels collaboratifs reposant sur une communication entre ses utilisateurs, vient alors le besoin d'approfondir les notions de la présence de ceux-ci et de celle des autres.

Le projet « environnements susceptibles » s'inscrit directement dans cette lignée. C'est un projet de recherche création dirigé par le Métalab de la Société des arts technologiques (SAT) visant à mettre en place une matière virtuelle évolutive et altérée par la présence humaine. Il implique le développement d'une plateforme de télé-présence immersive rendant possible la création de mondes virtuels distribués et partagés, ainsi que l'observation des règles de socialisation qu'ils induisent.

Le Métalab de la Société des arts technologiques

Le Métalab est un espace de recherche mettant en relation utilisateurs, créateurs, chercheurs, designers et développeurs. S'appuyant sur les relations entre les technologies, l'art et la société, ce laboratoire vise à explorer et transférer des connaissances dans des contenus numériques riches en relation avec l'audiovisuel et les données. Il est composé d'une équipe de recherche d'une trentaine de personnes dont des chercheurs associés, des étudiants-chercheurs, et des développeurs-chercheurs. Cette équipe remplit la mission de recherche de la SAT et réalise des projets en collaboration avec des artistes ou avec des organismes. Son programme se divise entre la scénographie immersive, l'interaction intelligente et l'expérience réseau.

Les objectifs du projet « environnements susceptibles »

Les objectifs principaux du projet « environnements susceptibles » sont d'enrichir l'expérience de la réalité partagée, qu'elle soit physique, augmentée ou virtuelle, par l'ajout de caractéristiques évolutives, de phénomènes réactifs et de moyens de laisser des traces tant

éphémères que durables de la présence humaine. L'environnement devient ainsi un système dynamique d'échange et de partage en direct ou en différé, offrant de nouvelles modalités aux relations interpersonnelles et à l'affirmation de soi. Cet environnement mettrait également en lumière les mécanismes complexes qui régissent le monde physique, les phénomènes propres aux écosystèmes complexes et les effets de nos comportements et attitudes sur l'environnement naturel et humain.

Originalité du contexte de la SAT

Peu d'équipes dans le monde combinent à la fois les compétences et l'expérience en développement de composantes immersives et interactives, en programmation générative d'environnements dynamiques, et en développement d'interfaces inédites. La SAT est une de ces équipes et ses atouts seront mis à profit dans le projet « environnements susceptibles ».

Impact sur les pratiques en art numérique

Le projet « environnements susceptibles » élargit la notion de présence dans les œuvres immersives et interactives en réseau qui, jusqu'ici, s'exprime exclusivement par les contributions directes de contenus et de présence en ligne. Les avancées en « espaces intelligents » qui en seront issues profiteront à la communauté élargie d'artisans de la création numérique, qu'ils soient du monde de l'art, de l'industrie des jeux vidéo ou de la simulation.

Développement d'une application graphique dans le cadre de ce projet

L'application graphique développée à l'égard de ce projet de maîtrise est un prototype servant de fer de lance au projet « environnements susceptibles ». Elle tente de répondre au problème de modélisation de la présence et l'éphémère, remédiant aux contraintes que peut poser le temps réel et proposant différents moyens d'interaction avec l'environnement. Elle puise ses forces dans des librairies de développement d'application graphique de bas et haut niveau telles que OpenGL, libQGLViewer et Qt. De plus elle profite des nuanceurs pour

atteindre une qualité de rendu précise et laisser un grand nombre d'empreintes simultanément dans la scène.

L'emploi de la plateforme SPIN par l'équipe du MétaLab a désigné plusieurs outils de développement qui lui sont compatibles : OpenGL et OpenScenegraph. Cependant des lacunes critiques d'OpenScenegraph ont mené à son remplacement par d'autres libraires alliant interaction et développement graphique : Qt et libQGLViewer. L'utilisation des nuanceurs reste compatibles avec toutes ces librairies et permet d'implémenter les empreintes de pas et de répondre aux contraintes d'affichage, temps réel et de stockage d'information des empreintes.

La réalisation de ce projet comporte plusieurs défis et problèmes. Les travaux de recherche dans le domaine de la dégradation d'environnement se dirigent davantage vers une amélioration du réalisme des scènes graphiques. S'inspirer de tels travaux n'est pas évident pour trouver des solutions s'identifiant au sentiment de présence des autres utilisateurs. La nature des dégradations doit pouvoir communiquer le sentiment de présence des autres utilisateurs. De plus, les méthodes d'interaction introduisent une autre dimension de réflexion sur le sujet. Grâce à ces méthodes, l'utilisateur doit pouvoir laisser des empreintes dans l'environnement en respectant les contraintes du temps réel. Ces dernières soulèvent également des points importants quant au choix des marques de l'éphémère pour un affichage prompt et une transmission optimale d'un site à un autre dans un environnement virtuel distribué. En outre les dimensions de l'environnement sont à définir. Les utilisateurs seront plusieurs à se déplacer dans l'environnement simultanément dans certains cas ou seul. La taille de l'environnement devra permettre une exploration satisfaisante tout en étant assez petite pour que les utilisateurs trouvent leurs traces. Enfin les requis du projet de maîtrise sont restés longtemps imprécis ce qui a rendu plus difficile de déterminer les fonctionnalités qui feraient de la collaboration avec la SAT une réussite.

Les objectifs majeurs sont d'ordre technique et d'ingénierie. D'un coté l'éphémère doit être représenté par des traces de pas sur le sol. L'utilisateur doit se voir offrir des méthodes d'interaction lui permettant de simuler des déplacements dans l'environnement. L'application doit répondre à des contraintes d'affichage, de stockage et de temps réel. Elle doit permettre

le traitement en mémoire de milliers d'empreintes, tout en les stockant en mémoire sans affecter l'affichage et en respectant les limites du matériel supportant l'application. D'un autre côté, le travail d'équipe emmène des objectifs de communication entre collaborateurs et d'identification des besoins. Le projet de maîtrise doit pouvoir persister dans le temps pour pouvoir être réutilisé par l'équipe du projet « environnement susceptibles ». Finalement l'objectif de s'adapter au cahier de charges imprécis s'imposait.

Pour répondre à ces objectifs, la méthodologie abordée fut de commencer par situer le projet de maîtrise dans le projet « environnements susceptibles ». Ce dernier se divise en une étape de développement technique et une étape d'exploration artistique et l'application graphique se situe aux prémisses du développement technique. Cette étape de développement technique renseignait sur les technologies à utiliser pour que le développement de l'application puisse s'imbriquer facilement avec d'autres travaux du projet « environnements susceptibles ». Une phase de recherche de travaux effectués dans le domaine de la dégradation de l'environnement a permis d'identifier des dégradations puis de les mettre en relation avec le contexte du projet de maîtrise. Les empreintes de pas ont finalement été implémentées puis les méthodes d'interactions utilisées devaient pouvoir reposer sur l'éventail de fonctionnalités proposé par les outils de développement tels que Qt ou libQGLViewer. L'utilisation des nuanceurs fut prévue pour un travail sur l'éclairage et sur la modélisation des traces de pas. Au niveau du processus d'ingénierie, les livrables restaient à être définis puis les exigences à être analysées.

Lors de sa conception, un premier prototype permettait d'afficher un grand nombre de dégradations autant sur le sol que certains murs. Cependant, le processus de modélisation des dégradations offrait un réalisme trop faible. Dans le deuxième prototype, les dégradations se sont réduites à une seule, les empreintes de pas. Les points saillants de ce deuxième prototype sont l'implémentation du *normal mapping* pour l'éclairage, le stockage d'information des empreintes dans des textures, la division de la scène en zone ayant chacune une liste d'empreinte puis l'introduction d'une interface graphique permettant d'alterner entre un mode de déplacement par clic et un autre par chemin tracé à la souris. Dans ce prototype, plusieurs milliers d'empreintes peuvent coexister sans ralentissement de l'application.

Dans ce rapport nous nous intéresserons en premier lieu à l'état de l'art dans le contexte du projet maîtrise. Les objectifs ainsi que la méthodologie employés seront présentés. Une section sur la discussion donnera suite à une partie sur la conception. Enfin une conclusion clôturera le rapport suivi de recommandations.

CHAPITRE 1

État de l'art

1.1 Détérioration d'environnement dans la recherche

La détérioration d'environnement apparaît souvent sous le thème de la simulation de phénomènes naturels. Au cours des dernières années, différentes tendances et techniques sont apparues, pour certaines globales, pour d'autres concentrées sur des phénomènes plus spécifiques.

1.1.1 Dégradation par synthèses de texture

Beaucoup de travaux de recherche s'intéressent à des méthodes copiant de multiples effets de détériorations à partir d'images de l'environnement en tirant profit de la synthèse de texture. Gu et al. (2006) capturent des BRDF qui varient dans le temps et l'espace en sept dimensions à l'aide d'un dôme composé de seize caméras et cent-cinquante sources de lumières. Ils peuvent alors obtenir la texture, la réflectance et les aspects des matériaux avec leur variation dans le temps. Par factorisation, l'apparence et les aspects évoluant dans le temps sont séparés pour que de nouveaux modèles soient générés par synthèse de texture.

Paquette et al. (2007) ont conçu une méthode visant à éviter le besoin d'apprendre les méthodes d'érosion avec des paramètres. Cette méthode s'oriente davantage vers une reproduction d'effets par synthèse de textures. L'artiste peut lui-même sélectionner les zones de dégradation dans des images, à partir desquelles le système produit un « masque de dégradation ». Le masque peut être édité et les régions où la texture devrait être vieillie peuvent y être spécifiées.

Lu et al. (2007) proposent de capturer des textures sur de vrais objets, dont les effets vieillissants varient dans le temps, ainsi que leurs paramètres géométriques contextuels pour donner plus de contrôle aux artistes. Les liens entre textures et paramètres contextuels permettent de transférer les textures à des objets dont les géométries sont différentes. En effet

la variation d'une texture dépend du processus de dégradation qu'elle subie, mais également de la géométrie affectée. Les facteurs systématiques qui influent sur le processus de dégradation sont intuitifs, physiques, mesurables et extraits de travaux antérieurs inspirés par la physique : l'exposition de la surface, l'orientation de la surface contre la position de l'agent vieillissant, la courbure de la surface, l'épaisseur de la surface. Un algorithme sélectionne ensuite les paramètres contextuels répondant le mieux aux variations de textures mesurées. L'approche ici se base sur une analyse de données et des caractéristiques physiques pour capturer des processus chimiques, mécaniques voire biologiques évoluant dans le temps.

1.1.2 Vieillissement de scène

Chen et al. (2005) présentent une approche de transport de dégradation. Leur technique se base sur un transport de particules qui entraînent une dégradation sous forme de vieillissement. Par un processus itératif prenant quelques minutes, des « gamma-ton » sont émis depuis des sources de « gamma-ton ». Finalement les effets d'érosion sont générés suivant la « carte de gamma-ton » produite précédemment. Les effets d'érosion sont conséquences des modifications des propriétés des matériaux et de la géométrie de la scène, de multi-texturing, synthèse de texture et displacement mapping. De plus les sources de vieillissement, lors de futures itérations, deviennent des sources de « gamma-ton » donnant lieu à un phénomène de « gamma-transport ». Cette technique peut donc s'adapter à la géométrie de la scène, couvrir des scènes urbaines. De plus par ses itérations elle prend compte des effets de transport global et d'effets de multi-vieillissement.

Dans le thème du vieillissement des bâtiments d'une scène urbaine, une méthode basée sur l'efflorescence a été publiée par Mérillou et al. (2012). Ils s'inspirent du principe de solubilité du sel contenu dans les matériaux de bâtiment et de son comportement face aux conditions externes, à leur localisation et aux caractéristiques architecturales pour simuler la dégradation. L'utilisateur peut localiser des zones à dégrader, puis le modèle, en fonction de la géométrie et des caractéristiques des matériaux, localise les attaques, génère la profondeur et concentration de sel et crée les textures de vieillissement.

1.1.3 Saleté et poussière

Certains sujets plus spécifiques comme l'accumulation de poussière dans un environnement ou sur des objets ont été explorés. Sun et al. (2007) ont proposé un travail qui explore trois types de phénomènes : l'accumulation de poussière, le séchage de la peinture et le séchage de surfaces rugueuses mouillées. Ils se concentrent sur le changement temporel dans le modèle diffus de texture. Pour cela ils font varier les BRDF de matériaux dans le temps. Un outil de mesure de BRDF enregistre 41 échantillons pour des cas de séchage de peinture, surfaces rugueuse et accumulation de poussière. Pour chaque instance, des modèles de BRDF variant dans le temps sont dérivés, prêts à être intégrés dans des logiciels de rendu et contrôlés par des facteurs environnementaux sur de nouveaux matériaux.

Gu et al. (2007) se sont intéressés à une approche créant des apparences de vieillissement telles que l'accumulation de poussières, de saleté ou de lipides sur des surfaces transparentes d'objets. En effet ce type d'accumulation de poussière donne lieu à des phénomènes plus complexes en ce qui touche à la propagation de la lumière. Ils modélisent le contaminant comme étant une couche très fine d'éléments éparpillés sur la surface transparente. Ils construisent par la suite un modèle analytique de BRDF et s'inspirent de la théorie de la radiosité pour étudier le transport de la lumière entre la surface de contaminant et la surface transparente et capturer les effets visuels sur les couches transparentes. Ils peuvent aussi bien utiliser leur technique sur des scènes en 3D que des photos en 2D.

1.1.4 Corrosion

Merillou et al. (2001a) se sont intéressés au phénomène de corrosion. Ils ont proposé un modèle de corrosion du métal affectant la géométrie, la couleur et la réflectance d'objets 3D. Dans une première étape le type de corrosion peut être choisi (uniforme pour toute la surface du métal, galvanique quand deux métaux entrent en contact, corrosion créant des crevasses et d'autres). Une carte de diffusion permet de contrôler le point de départ de la corrosion, la diffusion sur la surface et le volume de diffusion. Des normes physiques représentant des

phénomènes environnementaux réels de la corrosion permettent de contrôler le taux de corrosion. L'étape du rendu de la scène inclus également des modifications empiriques du paramètre de rugosité et un post-traitement par BRDF apporte de la porosité.

1.1.5 **Craquelure et écaillage de la peinture**

La peinture est souvent sujette à une dégradation au cours du temps, phénomène étudié par Paquette et al. (2002). Leur approche est basée sur un modèle où la couche de peinture suit une représentation 2D des propriétés de sa surface avec un ensemble de segment représentant les craquelures. Les craquelures sont propagées dans la surface par un cycle de simulation dont les étapes sont la génération des contraintes de traction et adhésion et la comparaison de ces contraintes avec la force maximale de cassage et craquèlement que la couche peut supporter avant de craquer ou s'écailler. Leur approche fournit également le rendu de l'effet recourbant de l'écaillage de la peinture en se basant sur des données empiriques.

1.1.6 **Phénomènes d'écoulement d'eau**

Ce type de dégradation correspond à l'apparition de taches liées aux écoulements d'eau sur des surfaces. En 2010, une méthode fut développée par Endo et al. (2010). Ils ont considéré l'application du système de flux de particule de Dorsey et al. (1996) comme des méthodes pour ajouter des taches directement sur les façades de bâtiments. Leur variante permet à l'utilisateur d'ajuster des paramètres de façon interactive pour obtenir l'apparence tachée souhaitée. Les gouttes d'eau sont représentées par des particules et leur dissolution, leur transport et leur sédimentation par dépôt sont spécifiés par l'utilisateur. Varier et ajuster les paramètres est ici plus rapide qu'obtenir les taches depuis un système d'édition de photos. Cependant leur système prend plusieurs minutes par taches et ne peut être pas appliqué à des scènes d'échelle urbaine.

Bosch et al. (2011) ont aussi développé une approche appliquée aux phénomènes écoulements. Les caractéristiques principales de leur approche se concentrent sur un modèle

d'écoulement de l'eau et sur l'apparition des taches qui en résultent. Les effets de phénomènes d'érosion y sont séparés des matériaux depuis des photographies par une extraction de données de changement d'apparence. Le système permet ensuite d'appliquer les effets sur le matériau de l'objet ou de différente couleur.

1.1.7 Égratignures, rayures, impact

Merillou et al. (2001b) proposent une approche où les rayures sont affichées par des BRDF. De vraies rayures sont mesurées par un dispositif mesurant la forme d'une surface. Les valeurs ainsi obtenues définissent les BRDF pour chaque matériau. Pour définir la location des égratignures en elle-même, une technique de texture est utilisée où une couleur indique la zone égratignée et l'autre une zone non égratignée.

Paquette et al. (2001) se sont intéressés à des dégradations mécaniques sous forme d'impact. Leur technique est semi-automatique et simule la déformation d'un objet par impact répétitif sur de longues périodes. La simulation de l'impact repose sur le choc d'un autre objet sur la surface et le raffinement du maillage polygonal de la surface. L'utilisateur sélectionne un outil qui simule l'objet frappant la surface. Il peut déterminer plusieurs chemins que suivra l'objet dans sa trajectoire ainsi que d'autres paramètres tels que sa forme, sa taille et son volume compacté. Chaque chemin correspond à une trajectoire et un choc d'objet. Enfin l'outil raffine le maillage en projetant les points caractéristiques des objets sur la surface, et en modifiant les polygones.

Dans cette présentation de l'état de l'art en terme de dégradation de l'environnement, plusieurs travaux apparaissent en lien direct avec le projet « environnement susceptibles » et le projet de maîtrise. Les éléments les plus intéressants à analyser sont à titre d'exemple la capacité de certaines dégradations à être directement administrées par des utilisateurs, l'emploi de certaines techniques dans le cadre d'une application en temps réel et la possibilité d'interaction utilisateur offerte avec les dégradations. Une analyse plus détaillée de ces éléments sera présentée dans la section 4.1 du document.

1.2 Outils de développement

Différentes librairies graphiques et outils sont nécessaires dans le cadre du développement d'une application graphique pouvant dégrader un environnement.

1.2.1 Librairie d'affichage graphique

OpenGL est une librairie graphique 2D et 3D. Conçu en 1992, OpenGL est devenu l'API d'application graphique 2D et 3D le plus utilisé dans l'industrie. OpenGL promeut l'innovation et le développement d'applications performantes grâce à l'incorporation d'ensemble de rendus, de mappage de textures, d'effets spéciaux et de fonctions de visualisation. C'est également un outil multi-plateforme devenu véritable standard dans le monde du développement d'applications graphiques. OpenGL permettra dans le projet de maîtrise d'assurer le rendu de la scène, des lumières, des textures et de la visualisation de la scène.

Il est possible de personnaliser certaines opérations du pipeline d'OpenGL donnant plus de contrôle sur le résultat souhaité, et ce, grâce au langage de nuanceur. Un langage de nuanceur permet la programmation graphique s'exécutant sur la carte graphique en temps réel. Les nuanceurs permettent d'obtenir une qualité d'image maximale en intervenant directement aux étapes de traitement de sommets et de fragments du pipeline d'OpenGL. Un fragment correspond à un pixel interpolé entre plusieurs sommets d'une primitive.

Le GLSL (Rost et al., 2009) est le principal langage de nuanceur d'OpenGL. Inspiré du langage C, il partage les caractéristiques des langages de nuanceur classiques à savoir : le développement de nuanceur de sommet et de nuanceur de fragment, le support des vecteur comme type fondamental, la génération de fragments interpolés pour les programmes de fragment et l'accès aux textures et autres tampons de données. Les nuanceurs permettront dans le projet de maîtrise d'avoir une plus grande emprise sur l'éclairage de la scène et le dessin des empreintes de pas.

1.2.2 Librairie de graphe de scène

OpenSceneGraph (Wang et Qian, 2010) est un cadriel d'outils de rendu graphique 3D de haute performance disponible en code source ouvert. Implémenté en C++ et OpenGL, il est multiplateforme et fonctionne sous divers systèmes d'exploitation : Windows, MacOS, GNU/Linux, IRIX, Solaris, HP-Ux, AIX et FreeBSD. La caractéristique principale d'OpenSceneGraph est qu'il s'organise autour de la construction de graphes de scène et qu'il encapsule des fonctions OpenGL dans un cadre orienté objet. Quatre librairies cœur, trois librairies d'interfaces graphiques et plusieurs librairies auxiliaires le structurent. L'implémentation en OpenGL du projet de maîtrise ouvre la possibilité de l'importer sous OpenSceneGraph. Cela produirait une application plus compatible avec l'environnement de développement du projet « environnements susceptibles ».

1.2.3 Librairie d'environnement collaboratif

La plateforme SPIN (*Spatial Interaction Framework*) correspond à une suite d'outils et d'applications qui permettent la gestion des flux de contrôle dans les environnements virtuels distribués et les systèmes de visualisation 3D. Cette plateforme permet de créer des environnements immersifs et de réalité virtuelle, de créer de la 3D collaborative, d'incorporer de la téléprésence, de la visualisation ou de la performance distribuée et de créer des jeux en ligne. Il est possible d'importer des éléments tels que des images, des vidéos, des objets 3D ou des sons et de leur donner des propriétés spatiales, de tailles, d'orientation, d'amplitude et autres dans le but de construire des scènes animées en temps réel.

SPIN est développé à partir d'OpenSceneGraph et d'OpenSoundControl, protocole de synthèse audio et de communication en réseaux. Les caractéristiques les plus essentielles à retenir sont :

- L'architecture de la plateforme SPIN, basée sur OpenSceneGraph
- Le modèle de communication client-serveur.
- Le support d'environnements immersifs distribués en réseaux.
- Le support d'interaction collaborative dans la réalité virtuelle partagée.

SPIN offre de ce fait un éventail d'outil pour transformer l'environnement virtuel du projet de maîtrise en environnement distribué.

1.2.4 **Librairies d'interfaces graphiques**

Le projet de maîtrise nécessite des outils de développement pour interagir avec l'utilisateur. Ces outils sont Qt (Thelin, 2007) et libQGLViewer. Qt est un cadriel de développement multiplateforme pour des systèmes bureautiques, embarqués ou mobiles. Il utilise le C++ et s'emploie, entre autres, pour développer des interfaces utilisateurs graphiques supportant la gestion de fichiers, l'accès aux bases de données, l'interaction avec l'utilisateur, la gestion de fils d'exécution et le support de réseau.

La librairie libQGLViewer est basée sur Qt et facilite la liaison entre les interfaces utilisateurs de Qt et les applications graphiques OpenGL. Elle fournit des fonctionnalités d'interfaces utilisateur 3D, telles que la possibilité de déplacer la caméra virtuelle à l'aide de la souris ou de sélectionner des objets dans une scène. Elle facilite la création d'application OpenGL interactive tout en étant multiplateforme.

Le C++ est un langage de programmation orienté objet issu du langage C et mis au point dans le but d'améliorer ce langage. Véritable standard du monde du développement informatique, il supporte à l'instar d'autres langages de programmation orienté objet : l'encapsulation d'objets, la composition de classes, l'héritage entre classe, la surcharge d'opérateur. Le C++ est le langage qui implémente le projet pour une question de pratique. Il implémente déjà Qt et libQglViewer. L'employer en tant que langage de développement du projet facilite la compatibilité entre les interfaces graphiques de Qt, libQglviewer et de ce fait OpenGL.

1.3 **Problématiques**

Les sections qui suivent présentent les différents problèmes en lien avec la réalisation de ce projet.

1.3.1 Application dans le contexte du réalisme

Les travaux de recherche précédents font état de dégradations appliquées à des scènes 3D dans le but d'en améliorer le réalisme. Ce réalisme est indispensable au sentiment d'immersion mais ces travaux ne fournissent pas de solutions concrètes au sentiment de présence des autres utilisateurs.

1.3.2 Emploi en tant que symbole de « l'éphémère »

Certains types de dégradations peuvent être qualifiés d'éphémères, mais elles ne remplissent pas le but que l'éphémère devrait remplir : donner un sentiment de présence et être administré par des utilisateurs en guise de leur signature dans l'environnement. Le choix du type de dégradation qui traduit avec justesse la notion artistique de présence et d'éphémère recherchée n'est pas évident et demandera probablement des essais et erreurs.

1.3.3 Notions de présence et d'interaction

Dans le cadre du projet « environnements susceptibles », l'utilisateur est celui qui crée les dégradations et laisse des traces. Il se doit donc d'interagir avec l'environnement et ce dernier doit lui fournir des méthodes d'interaction intuitives. Ce principe d'interaction en temps réel est aussi le facteur qui renforce le sentiment de présence, principe souvent absent dans les travaux de recherche sur la dégradation. De plus, les méthodes d'interaction existantes pour la détérioration ont davantage le but de faciliter l'application des dégradations dans l'environnement pour des artistes plutôt que de permettre à des utilisateurs arbitraires de dégrader la scène. Choisir un paradigme d'interaction qui conviendra à des utilisateurs qui ne sont pas des experts posera certains défis.

1.3.4 Développement de l'application

Dimensions de l'environnement

Plusieurs utilisateurs navigueront dans l'environnement, parfois de façon concurrente alors qu'à d'autres moments ce sera individuellement sur de plus longues périodes de temps. Déterminer la taille de l'environnement devient alors difficile. L'environnement se doit d'être assez grand pour être intéressant à explorer, et assez petit pour que plusieurs utilisateurs puissent passer au même endroit et ainsi constater la présence des autres.

Une application en temps réel

L'application soulève le problème de fonctionner en temps réel. Elle doit permettre à ses utilisateurs de pouvoir interagir avec l'environnement et entre eux sans qu'il y ait de délais ou de retard d'exécution, tout en garantissant une qualité visuelle suffisante. Le problème d'affichage temps réel amènera des contraintes quant au choix des marques de l'éphémère afin qu'elles puissent être affichées rapidement et transmises rapidement d'un site à un autre dans l'environnement virtuel distribué.

1.3.5 Collaboration entre les intervenants

L'application s'inscrit dans le cadre du projet « environnements susceptibles ». Cela implique de travailler directement avec l'équipe du projet, communiquer avec elle, lui présenter des résultats et autres. Certains aspects de cette application doivent être complémentaire avec le projet « environnements susceptibles » et y être aisément incorporés. Les requis initiaux et tout au long du projet n'étaient pas très précis. Déterminer les fonctionnalités qui proposeront une valeur ajoutée intéressante pour l'équipe artistique devient alors difficile.

CHAPITRE 2

OBJECTIFS

2.1 Processus technologique

Les objectifs principaux en termes de processus technologiques sont de trouver une modélisation de l'éphémère, trouver des moyens d'interaction avec l'utilisateur et gérer les contraintes liées à l'affichage et au stockage d'information de l'application.

2.1.1 Modélisation de l'éphémère visuellement sous forme de traces

L'éphémère doit être représenté par des traces. Des dégradations laissées par l'utilisateur doivent être modélisées sous une forme intuitive et simple à reconnaître. Dans le cadre de l'application, des empreintes de pas font office de traces dans la scène. Les empreintes de pas sont une trace évidente que laisse un homme dans son sillon dans son déplacement.

2.1.2 Moyens d'interaction avec l'utilisateur

L'application doit permettre à l'utilisateur de laisser des traces dans l'environnement. En se référant au type de traces indiqué, notamment les empreintes de pas, le moyen d'interaction qui en ressort est le déplacement dans l'environnement. L'utilisateur devrait être en mesure de produire un déplacement qui laisserait des traces de pas sur le sol.

2.1.3 Gestion des contraintes d'affichage, de stockage et de temps réel

L'application doit être en mesure d'afficher, de transmettre, de stocker en mémoire ainsi que sur le disque dur, des milliers d'empreintes sans ralentissement de l'affichage et sans nécessiter plus de mémoire que ce qu'une carte graphique contemporaine peut offrir, soit

entre 1.0 et 1.5 Go. Elle doit aussi proposer un éclairage qui mette en relief les empreintes de pas.

2.2 **Processus d'ingénierie**

Les processus d'ingénierie reposent sur le travail d'équipe, la persistance et le partage du projet de maîtrise dans l'équipe ainsi que l'adaptation au cahier des charges.

2.2.1 **Travail en équipe**

Le développement de l'application s'inscrit directement dans le projet « environnements susceptibles », lui même géré par une équipe au sein du Métalab de la SAT. L'objectif de travailler en équipe devient logique. Il s'agit également de comprendre les besoins des collaborateurs et d'y apporter des solutions efficaces.

2.2.2 **Intégration du projet de maîtrise**

L'application sera incorporée plus tard dans le projet « environnements susceptibles ». Elle doit être facile à réutiliser par l'équipe et devrait s'arrimer avec les technologies employées dans le projet « environnements susceptibles ». Par conséquent elle doit être d'emblée accessible à l'équipe au complet pour pouvoir être consultée. Certaines technologies permettent de partager des contenus entre différents contributeurs. Le Métalab de la SAT utilise dans ce but un serveur muni de Redmine (Lesyuk, 2013) sur lequel sont hébergés ses différents projets. Redmine est une application de gestion de projet en code source ouvert. Elle propose aussi des services d'hébergement et de versionnage de projet. Elle supporte automatiquement la persistance de données pour palier au risque de perte de données et permet le partage des fichiers notamment au sein du projet « environnements susceptibles ».

2.2.3 Adaptation à un cahier des charges changeant

Les chefs d'équipe du projet « environnements susceptibles » sont également ceux qui en déterminent le cahier des charges. Dans le cadre du développement de l'application, le cahier des charges était informel et se caractérisait par des directions globales que l'application devait prendre. L'objectif d'avancer pas à pas s'imposait donc, en gardant à l'esprit les directions données par l'équipe.

CHAPITRE 3

MÉTHODOLOGIE

La méthodologie concernant le projet « environnement susceptibles » sera traitée dans la section 3.1 et 3.2. Elle s'articule autour des étapes de réalisation du projet « environnements susceptibles » et de la description de l'équipe du projet. Les sections 3.3 et 3.4 s'orientent davantage vers la méthodologie du projet de maîtrise, notamment les processus technologiques et d'ingénierie mis en oeuvre.

3.1 Étapes de la réalisation

Les étapes du projet « environnement susceptibles » se répartissent en deux catégories : son développement technique et ses explorations artistiques. Le projet « environnement susceptibles » comporte une série d'expérimentations techniques et artistique conjointes la première année, un projet exploratoire par chacun des deux chercheurs la deuxième année, et un projet conjoint et intégrateur la troisième année.

3.1.1 Développements techniques

Les développements techniques sont matérialisés par l'élaboration de techniques de mémorisation des actions humaines dans les mondes virtuels sous forme de traces structurales environnementales - ou encore un système pouvant faire apparaître des formes visuelles et sonores d'usure liées à l'intensité de manipulations et à la fréquence des passages. Ces techniques sont :

- Des ajouts à l'environnement SPIN permettant la simulation de systèmes dynamiques. Ils incluent la modélisation physique du mouvement pour le développement de programmes simulant le comportement de corps en mouvement comme des balles de billard, des particules et des liquides.

- Le développement d'interfaces pour l'interaction physique (excitation, déplacement, perturbation, etc.) avec des simulateurs, des systèmes dynamiques et des systèmes évolutifs.
- Le développement de matières virtuellement malléable qui seront ajoutées à l'environnement de développement de SPIN et plus spécifiquement, le développement de nouveaux types de « nœuds » pour la déformation par pliage, cintrage et pinçage de structures polygonales manipulables en temps réel. Le but est de simuler des terrains, des tissus et des membranes animés.
- L'adaptation et l'intégration des librairies de modélisation physique sonore (simulation de cloches, cordes et instruments à vent) dans SPIN. Ces librairies qui existent dans le domaine public seront adaptées et intégrées.
- Le développement de capacités d'enregistrement localisées dans l'environnement virtuel et physique pour créer des étiquettes visuelles et sonores.
- Le développement d'interfaces de navigation et de manipulation au sein de ces « environnements susceptibles ».

Force est de constater que le projet « environnements susceptibles » couvre un très large éventail de fonctionnalités. Le projet de maîtrise couvre un sous-ensemble en s'attaquant au développement d'interface physique, de navigation et de manipulation.

3.1.2 Exploration artistique

L'exploration artistique se répartit sur 3 années. La première année, une série de sketches préliminaires seront réalisés. Ils auront comme objet l'exploration artistique et le développement technique, ainsi que leurs validations. La deuxième année deux projets seront réalisés par Luc Courchesne et Zack Settel. Ils permettront à chacun de valider les technologies développées dans la première année dans leurs pratiques respectives en image et en son. Le premier projet compte répertorier des concepts de traces, d'usure et d'érosion de nouvelles catégories de matières virtuelles, les décrire et les mettre à l'essai dans un espace d'expérimentation où des visiteurs-testeurs pourront se familiariser avec leurs

caractéristiques formelles et plastiques. Le deuxième projet portera sur l’interaction utilisateur basée sur des simulations de systèmes dynamiques telles que les particules dans des fluides. Finalement, un projet intégrateur mettra à contribution les développements techniques et les explorations artistiques des années un et deux en proposant une expérience ancrée à la fois dans les espaces physique, augmentés et virtuels. Ce travail collaboratif et évolutif proposera une expérience unifiée des espaces physique, augmenté et virtuel. Les visiteurs seront en mesure de circuler librement au sein de ces espaces, contribuant chacun à sa façon à leurs constructions ou à leurs modifications. Les artistes souhaitent ainsi proposer un monde en évolution constante où se pose la question du permanent et de l’éphémère, du mouvement et de la transformation constante, qu’elle soit perceptible ou non, de nos réalités.

Le projet de maîtrise s’inscrit dans le cadre de la première année de ce programme. Il représente un sketch parmi plusieurs autres et se concentre sur la dégradation d’un environnement par des empreintes de pas.

3.2 Équipe de la Société des arts technologiques

L’équipe du projet « environnement susceptibles » est composée de Luc Courchesne (image) et Zack Settel (son) ainsi que d’une équipe de développeurs, de techniciens et d’étudiants.

3.2.1 Directeurs de projets

Luc Courchesne et Zack Settel sont conjointement directeurs du projet « environnement susceptibles ». M. Courchesne apporte son expérience en conception et en réalisation d’installations immersive interactives ainsi que les outils de sa plate-forme de télé-présence immersive résultant d’une démarche initié au milieu des années 1990 avec les projets Paysage no. 1 (1997) et Rendez-vous (1999). M. Settel apporte son expérience en conception et en réalisation de performance et d’installations immersive interactives musicales et sonores. Ses travaux s’appuient sur son projet de recherche, initié en 2004, en audio 3D dans le contexte de la réalité augmentée et la réalité virtuelle.

3.2.2 Développeurs

Mike Wozniewski est un développeur collaborateur de M. Courchesne et de M. Settel. Son travail s'étend sur la plateforme de prototypage de réalités virtuelles partagées (SPIN) adoptée par les deux chercheurs dans de nombreux travaux. Auteur principal de la plateforme SPIN, il est en mesure de l'enrichir avec des algorithmes qui permettront à la fois à l'environnement virtuel d'évoluer de façon autonome tout en prenant en compte les « perturbations » amenées par les utilisateurs.

L'équipe se compose aussi d'étudiants de 2^e cycle, de 3^e cycle et au post-doctorat de l'Université de Montréal, de l'Université Concordia et de l'ÉTS qui seront amenés à contribuer conceptuellement et techniquement au développement des projets. C'est ici que la majeure partie de la contribution de ce projet de maîtrise entre en jeu.

3.3 Processus technologiques

3.3.1 Études de travaux de recherche sur la dégradation

Une étude de travaux de recherche a été réalisée avant le développement de l'application graphique. Les travaux de recherche peuvent être retrouvés dans le chapitre 1. Ils ont servis à cerner une première approche en terme d'application de dégradation dans un environnement virtuel.

3.3.2 Mise au point de méthodes d'interaction intuitives

Pour mettre en place une méthode d'interaction intuitive, les bibliothèques Qt et libQGLViewer ont été utilisées. En effet elles fournissent des outils de sélection d'objet, de gestion de périphériques tels que la souris ou le clavier. Le déplacement pourra être simulé à l'aide du déplacement de la souris, et certains paramètres pourront être modifiés par l'utilisateur pour altérer l'apparence des traces de pas.

3.3.3 Utilisation de nuanceurs

Les nuanceurs donnent un contrôle total sur le rendu des sommets et des fragments en remplaçant le pipeline fixe d'OpenGL, notamment dans la gestion de l'éclairage et de la modélisation de traces. Il sera possible de mettre au point un éclairage précis et détaillé de la scène. Le pipeline fixe d'OpenGL ne permet pas de varier le calcul de couleur et d'éclairage, ce qui est trop limitatif pour l'affichage d'empreintes de pas. Les nuanceurs remplacent le pipeline fixe d'OpenGL et permettent de calculer la luminosité de chaque fragment et donc de chaque pixel de la scène. L'utilisation des nuanceurs dans la modélisation des traces permettra d'appliquer les empreintes sur les fragments précis désirés.

3.4 Processus d'ingénierie

Le processus d'ingénierie se résumera à une analyse des exigences, une définition des livrables, une communication au sein de l'équipe par des réunions, la mise en place d'une gestion de configuration de l'application ainsi que l'étude de l'arrimage avec d'autres sous-projets « d'environnements susceptibles ».

3.4.1 Analyse des exigences et définition des livrables

L'analyse des exigences et la définition des livrables font partie de la méthodologie du projet de maîtrise. Les livrables se trouvent à être l'application en elle même. Les exigences seront évoquées plus en détail dans les sections 4.2.1 et 4.3.1.

3.4.2 Mises au point lors de réunions avec l'équipe

Les réunions avec l'équipe ont été très importantes. Elles faisaient partie de la méthodologie car elles ont permis de recadrer le projet de maîtrise et d'en identifier les exigences.

3.4.3 Gestion de configuration

La gestion de configuration du projet est celle utilisée dans le cadre des projets divers de la SAT. Le Métalab utilise git (Loeliger, 2009) et possède un serveur Redmine (Lesyuk, 2013). Ce système de versionnage assure la persistance du prototype.

3.4.4 Arrimage avec d'autres sous-projets

Pour l'étude des pistes d'arrimages avec d'autres sous-projets « d'environnements susceptibles », il s'agit d'identifier les technologies présentées dans certaines des exigences et de s'assurer que l'application les utilise. Ces technologies seront par conséquent partagées dans tous les sous-projets « d'environnements susceptibles ».

CHAPITRE 4

CONCEPTION

4.1 Mise en perspective des travaux de recherche

Les articles de recherche dans le domaine de la dégradation sont intimement liés au contexte du projet « environnement susceptibles ». Dans le cadre du projet « environnement susceptibles », certains types de dégradation ne possèdent aucun lien direct avec les notions de présence des autres et d'existence. Les travaux sur le vieillissement de scènes globales,

Table 4-1 Tableau récapitulatif des articles de recherche

Auteurs	Dégradation	Technique
Gu et al. (2006)	Multiples	BRDF variant dans le temps, génération de textures
Paquette et al. (2007)	Multiples	Sélection de zones dégradées, édition d'un masque de dégradation, génération de texture
Lu et al. (2007)	Multiples	Capture de textures sur de vrais objets avec des paramètres géométriques
Chen et al. (2005)	Vieillissement de scènes 3D	Transport de particules, modification de matériaux, géométrie, texture
Mérillou et al. (2012)	Vieillissement de bâtiments urbains par efflorescence	Simulation de la dégradation du sel, synthèse de textures
Sun et al. (2007)	Accumulation de poussière, séchage de peinture et de surfaces rugueuses	BRDF variant dans le temps, modèles de BRDF par phénomènes
Gu et al. (2007)	Accumulation de poussière, saleté ou lipides sur surfaces transparentes	BRDF et radiosité
Merillou et al. (2001a)	Corrosion	Carte de diffusion de corrosion inspirée de phénomènes physiques
Paquette et al. (2002)	Craquelure et écaillement de la peinture	Modèle de propagation de craquelures et recourbements sur une surface
Endo et al. (2010)	Taches liées à l'écoulement d'eau	Système de flux de particule avec dissolution, transport et sédimentation
Bosch et al. (2011)	Taches liées à l'écoulement d'eau	Extraction de données de photographies et application sur des matériaux
Merillou et al. (2001b)	Égratignures, rayures	Modèle de BRDF, édition d'une texture de localisation
Paquette et al. (2001)	Impacts	Modification de maillage polygonal par simulation d'impact répétitifs

notamment celui sur l'émission de particules de dégradation et celui sur le vieillissement de bâtiments manifestent plus une variation en apparence d'éléments dans le temps (Chen et al., 2005; Mérillou et al., 2012). La corrosion est pareillement un phénomène qui se manifeste à l'insu de l'intervention l'homme.

L'analyse présentée dans les sections qui suivent consiste à étudier si certaines de ces dégradations peuvent être causées par un utilisateur jusqu'à quel point elles admettent une interaction avec celui-ci, si les techniques employées pour les reproduire sont compatibles avec du temps réel et si elles concordent avec une application dans des environnements distribués. La **Error! Reference source not found.** récapitule les dégradations trouvées dans les articles de recherche ainsi que les techniques employées pour les modéliser.

4.1.1 Dégradation d'un environnement par un des utilisateurs

Les dégradations suscitant une intervention directement d'un utilisateur sont en premier lieu de type mécanique. Ce type correspond aux dégradations par égratignures, rayures ou impacts (Merillou et al., 2001b; Paquette et al., 2001). Un utilisateur pourrait aisément modifier l'apparence d'objets par des mouvements ou avec l'aide d'outils. La saleté ainsi que les taches pourraient être provoquées de façons analogues. Ces deux types de dégradation se retrouveraient sur des objets ou les vêtements et des utilisateurs pourraient facilement les plaquer dans un environnement ou sur eux-mêmes.

D'autres dégradations (Bosch et al., 2011; Endo et al., 2010; Merillou et al., 2001a) symbolisent plus des phénomènes biochimiques observés dans la nature et seraient difficilement engendrées par l'homme. La corrosion est une de ces dégradations provoquées par un contact avec de l'eau, de l'air ou d'autres éléments. L'écoulement de l'eau est un cas particulier d'apparition de corrosion : il résulte en général de fuite d'eau ou de la circulation de l'eau. Un utilisateur pourrait asperger un objet avec un liquide corrosif. Le utilisateur pourrait également utiliser un liquide qui tacherait l'objet, ce qui pourrait être une alternative plus simple à réaliser.

Le craquèlement de la peinture apparaît avec le temps et ne nécessite aucune intervention de l'homme (Paquette et al., 2002). Quant au vieillissement de bâtiments par efflorescence, la seule action de l'homme est de bâtir les bâtiments. Tel que présenté dans l'article de Mérillou et al. (2012), la présence de sel dans les matériaux entraîne une réaction au contact de l'eau. L'application de ces dégradations par un utilisateur n'est donc pas innée. La Table 4-2 montre les relations que chaque article entretien entre ses dégradations et l'intervention d'un utilisateur.

Table 4-2 Dégradations provoquées par des utilisateurs

Auteurs	Utilisateur	Pertinence
Gu et al. (2006)	Discutable	Partiellement
Paquette et al. (2007)	Discutable	Partiellement
Lu et al. (2007)	Discutable	Partiellement
Chen et al. (2005)	Non, processus causé par l'environnement	Non
Mérillou et al. (2012)	Non, processus causé par l'environnement	Non
Sun et al. (2007)	Modification de l'apparence d'objets	Oui
Gu et al. (2007)	Traces de doigts sur des objets poussiéreux	Oui
Merillou et al. (2001a)	Non, corrosion non causée par l'homme	Non
Paquette et al. (2002)	Craquèlement de la peinture conséquence d'un choc	Oui
Endo et al. (2010)	Écoulement d'eau provoqué par un utilisateur	Oui
Bosch et al. (2011)	Écoulement d'eau provoqué par un utilisateur	Oui
Merillou et al. (2001b)	Égratignures et rayures causées par des utilisateurs	Oui
Paquette et al. (2001)	Impacts causés par un utilisateur	Oui

4.1.2 Dégradation par marquage d'empreintes de pas

Les empreintes de pas sont des dégradations qui ne sont pas directement abordé dans les travaux de recherche. Rechercher des points communs avec les travaux de recherche peut révéler des pistes d'implémentation et méthodologie. Parmi les dégradations qui s'en rapprochent se trouvent les égratignures, rayures et impacts (Merillou et al., 2001b; Paquette et al., 2001). Ces dégradations révèlent l'aspect mécanique dans l'action de produire

d'empreinte de pas. Une empreinte de pas s'assimile facilement à un impact effectué sur le sol. Par ailleurs, laisser des empreintes de pas derrière soi peut laisser des saletés dans le sol, le tacher voire laisser des empreintes dans de la poussière. Ces dernières dégradations peuvent s'ajouter aux dégradations mécaniques pour former des empreintes plus créatives.

Parmi les techniques passées en revues dans le corpus d'articles, plusieurs conviennent à la création d'empreinte de pas. Modifier les polygones d'une surface pour former une empreinte pourrait s'inspirer de la méthode de modification de polygones employée dans la dégradation par impacts (Paquette et al., 2001). Le changement de BRDF (Gu et al., 2007;

Table 4-3 Rapprochement entre les dégradations des travaux et les empreintes de pas

Auteurs	Empreintes de pas		Pertinence
	Lien	Technique	
Gu et al. (2006)	Empreinte dont l'apparence varie dans le temps	Capture de TSV-BRDF d'empreintes de pas sur plusieurs sols	Oui
Paquette et al. (2007)	Usage d'une texture d'empreinte entrée	Sélection de la zone de l'empreinte de pas	Oui
Lu et al. (2007)	Discutable (apparence de l'empreinte, pas la forme)	Capture de paramètres géométriques et de textures	Partiellement
Chen et al. (2005)	Vieillissement global incompatible avec empreinte	Non	Non
Mérialou et al. (2012)	Efflorescence incompatible	Non	Non
Sun et al. (2007)	Empreintes de pas dans la poussière, mouillant le sol	Capture de BRDF appliquées à des matériaux	Oui
Gu et al. (2007)	Empreintes de pas dans la poussière	Ajout d'une « forme d'empreinte » dans la répartition de la poussière	Oui
Merillou et al. (2001a)	Incompatibilité avec le phénomène de corrosion	Carte de corrosion incompatible	Non
Paquette et al. (2002)	Conséquence d'une empreinte dans un sol peint	Incorporation de la forme d'une empreinte à la carte	Partiellement
Endo et al. (2010)	Écoulement d'eau en lien avec une empreinte	Incorporation de la forme au modèle	Partiellement
Bosch et al. (2011)	Écoulement d'eau en lien avec une empreinte	Incorporation de la forme d'une empreinte à l'application sur matériaux	Partiellement
Merillou et al. (2001b)	Incompatibilité des rayures avec les empreintes	Non	Non
Paquette et al. (2001)	Impact en forme d'empreinte	Utilisation d'une forme d'empreinte	Oui

Gu et al., 2006; Merillou et al., 2001b; Sun et al., 2007) conviendrait aussi à cette emprise par la modification de paramètres de matériaux en fonction des localisation d'empreintes de pas.

Une autre technique dont le modèle de dépôt d'empreinte peut s'inspirer est la génération de textures. Un telle œuvre requerrait de capturer des photos d'empreintes de pas à partir de photographies ou de vraies empreintes de pas. Coupler l'utilisation de masques de dégradation (Merillou et al., 2001a; 2001b; Paquette et al., 2007) à cette méthode n'est pas adéquat car ces masques se limitent à aider l'artiste dans la localisation des empreintes sur une texture, ce qui est difficilement adaptable à la création rapide de traces de pas. La Table 4-3 évalue comment le type de dégradation abordé dans les articles s'applique à une empreinte de pas et si les techniques employées dans les articles peuvent générer une empreinte de pas.

4.1.3 **Rapport entre les dégradations et l'interaction**

Les phénomènes pouvant correspondre directement à une intervention humaine sont des dégradations mécaniques telles que les égratignures, rayures et impacts. La Table 4-4 met les travaux en perspective par rapport à l'interaction et informe sur les possibilités d'interaction qu'ont les dégradations traitées. Un utilisateur peut facilement causer un enfoncement dans un mur par un impact (Paquette et al., 2001), ou rayer du métal (Merillou et al., 2001b) avec un objet. Le séchage d'objet (Sun et al., 2007) peut également être liée à l'interaction avec un utilisateur. Une situation où un utilisateur aspergerait un objet d'eau qui sècherait progressivement est envisageable. L'accumulation de la poussière (Gu et al., 2007; Sun et al., 2007) peut également être synonyme d'interaction – un utilisateur pourrait laisser des empreintes de pas dans la poussière ou laisser des traces de doigts sur des objets poussiéreux. Des phénomènes vus dans les articles tels que les écoulements d'eau ont un lien discutable avec l'interaction. L'écoulement d'eau (Bosch et al., 2011; Endo et al., 2010) pourrait être provoqué par un utilisateur mais la corrosion à laquelle il laisse place est un phénomène qui prend beaucoup de temps à ce mettre en place dans le monde réel.

Les techniques décrites dans les articles de recherches sont souvent la modification d'apparence de matériaux par BRDF, la génération de textures ou la modification polygonale. Une interaction avec un utilisateur serait envisageable si elle devait engendrer de telles opérations pour modifier l'apparence d'objets ou de la scène. S'assurer que les délais d'exécution de ces techniques est raisonnable conditionne cette notion d'interaction et définit la validité de ces techniques.

Table 4-4 L'interaction dans les articles de recherche

Auteurs	Interaction	Pertinence
Gu et al. (2006)	Discutable	Partiellement
Paquette et al. (2007)	Application de masques de dégradation	Partiellement
Lu et al. (2007)	Discutable	Partiellement
Chen et al. (2005)	Non, processus causé par l'environnement	Non
Mérialou et al. (2012)	Non, processus causé par l'environnement	Non
Sun et al. (2007)	Interaction avec des objets et modification de leur apparence	Oui
Gu et al. (2007)	Traces de doigts sur des objets poussiéreux	Oui
Merillou et al. (2001a)	Corrosion non causée par l'homme	Partiellement
Paquette et al. (2002)	Craquèlement de la peinture conséquence d'un choc	Partiellement
Endo et al. (2010)	Écoulement d'eau provoqué par un utilisateur	Partiellement
Bosch et al. (2011)	Écoulement d'eau provoqué par un utilisateur	Partiellement
Merillou et al. (2001b)	Égratignures et rayures causées par des utilisateurs	Oui
Paquette et al. (2001)	Impacts causés par un utilisateur	Oui

4.1.4 Techniques à utiliser dans le contexte du temps réel

Le temps réel impose aux techniques de s'exécuter dans des délais brefs, assez pour ne pas ralentir l'exécution de l'application. Les aspects à considérer ici concernent les étapes post-traitements, suivant la capture de textures de photographies, objets réels ou l'édition de masques ou cartes de dégradation. La Table 4-5 met en perspective les travaux par rapport au temps réel et informe sur le temps de calcul des méthodes employées.

Dans les techniques et méthodes passées en revue dans les articles de recherche, les BRDF apparaissent souvent. La modification de propriétés de matériaux ou textures est souvent

Table 4-5 Le temps réel dans les articles de recherche

Auteurs	Temps réel	Pertinence
Gu et al. (2006)	Temps de calcul de 60 à 11 500 secondes	Non
Paquette et al. (2007)	Temps de calcul de 4 à 65 secondes	Oui
Lu et al. (2007)	Temps de calcul de 1 à 10s	Oui
Chen et al. (2005)	Temps de calcul de 60 à 180 secondes	Non
Mérillou et al. (2012)	Temps de calcul de 155 à 240 secondes	Non
Sun et al. (2007)	Temps de calcul de 15 à 900 secondes	Non
Gu et al. (2007)	Temps de calcul de 3 à 60 secondes	Oui
Merillou et al. (2001a)	N/A	Non
Paquette et al. (2002)	Temps de calcul de 180 secondes	Non
Endo et al. (2010)	Temps de calcul de 180 à 600 secondes	Non
Bosch et al. (2011)	Temps de calcul de 60 à 120 secondes	Non
Merillou et al. (2001b)	N/A	Non
Paquette et al. (2001)	Temps de calcul de 4 à 8 secondes	Oui

coûteuse en durée et ne pourrait, sauf dans certains cas, s'appliquer au temps réel. Les BRDF inspirés d'objets réel conduisent à des modèles coûteux en temps de calcul pour le temps réel (Pacanowski et al.). Les approches progressives où des apparences d'objet varient dans le temps génèrent beaucoup de données et sont plus coûteuses en mémoire. Elles demandent un calcul récurrent d'apparences à différents états comme un sol mouillé puis de plus en plus sec (Gu et al., 2006; Sun et al., 2007).

La génération de texture abordée dans plusieurs des articles semble moins coûteuse en temps de calcul. Dans l'article de Paquette et al. (2007) cette génération ne prend que quelques secondes. La technique de génération d'impact (Paquette et al., 2001) semble adaptée au contexte du temps réel, ne comptant que quelques secondes en temps de calcul. Finalement dans l'article de Merillou et al. (2001b), les techniques de *bump mapping* ou *displacement mapping* peuvent rajouter des ralentissements d'exécution supplémentaires.

4.1.5 Utilisation dans le contexte d'application distribuée

Une application distribuée implique un flux de données entre plusieurs sites distants. La Table 4-6 met en perspective les travaux par rapport aux environnements distribués. Les dégradations vieillissant la scène de façon globale (Chen et al., 2005; Mérillou et al., 2012) produiraient trop d'informations à échanger. Pour cette raison, les dégradations plus localisées telles qu'égratignures, impacts ou corrosion à titre d'exemple faciliterait les

Table 4-6 Les articles de recherche et les environnements distribués

Auteurs	Environnements Distribués	Pertinence
Gu et al. (2006)	Discutable (paramètres supplémentaires)	Partiellement
Paquette et al. (2007)	Discutable (paramètres supplémentaires)	Partiellement
Lu et al. (2007)	Discutable (paramètres supplémentaires)	Partiellement
Chen et al. (2005)	Non	Non
Mérillou et al. (2012)	Non	Non
Sun et al. (2007)	Discutable (paramètres supplémentaires)	Partiellement
Gu et al. (2007)	Discutable (paramètres supplémentaires)	Partiellement
Merillou et al. (2001a)	Discutable (paramètres supplémentaires)	Partiellement
Paquette et al. (2002)	Discutable (paramètres supplémentaires)	Partiellement
Endo et al. (2010)	Discutable (paramètres supplémentaires)	Partiellement
Bosch et al. (2011)	Discutable (paramètres supplémentaires)	Partiellement
Merillou et al. (2001b)	Discutable (paramètres supplémentaires)	Partiellement
Paquette et al. (2001)	Discutable (paramètres supplémentaires)	Partiellement

échanges de données (Merillou et al., 2001a; 2001b; Paquette et al., 2001). Les travaux de recherche ne fournissent aucun paramètre relatif à la localisation d'une dégradation excepté pour le travail de Paquette et al. (2007) où un masque de localisation de dégradation est introduit. Aucun article ne propose un modèle sous lequel les dégradations seraient stockées et partagées entre différents sites. Rappelons que leur objectif premier est d'améliorer le réalisme dans des scènes 3D.

4.1.6 Conclusion de la mise en perspective

Pour déterminer si les articles de recherches répondent aux besoins du projet de maîtrise, il faut tout d'abord déterminer si les dégradations qui y sont traitées peuvent être considérées comme des empreintes de pas et si les techniques et méthodes de modélisation peuvent

Table 4-7 Tableau de mise en perspective des travaux

Auteurs	Empreintes de pas		Interaction	Temps réel	Environnements Distribués
	Lien	Technique			
Gu et al. (2006)	Oui	Discutable	Discutable	Non	Discutable (paramètres supplémentaires)
Paquette et al. (2007)	Oui	Discutable	Discutable	Oui	Discutable (paramètres supplémentaires)
Lu et al. (2007)	Discutable	Discutable	Discutable	Oui	Discutable (paramètres supplémentaires)
Chen et al. (2005)	Non	Non	Non	Non	Non
Mérialou et al. (2012)	Non	Non	Non	Non	Non
Sun et al. (2007)	Oui	Discutable	Discutable	Non	Discutable (paramètres supplémentaires)
Gu et al. (2007)	Oui	Discutable	Oui	Oui	Discutable (paramètres supplémentaires)
Merillou et al. (2001a)	Non	Non	Discutable	Non	Discutable (paramètres supplémentaires)
Paquette et al. (2002)	Discutable	Discutable	Discutable	Non	Discutable (paramètres supplémentaires)
Endo et al. (2010)	Discutable	Discutable	Discutable	Non	Discutable (paramètres supplémentaires)
Bosch et al. (2011)	Discutable	Discutable	Discutable	Non	Discutable (paramètres supplémentaires)
Merillou et al. (2001b)	Non	Non	Oui	Non	Discutable (paramètres supplémentaires)
Paquette et al. (2001)	Oui	Discutable	Oui	Oui	Discutable (paramètres supplémentaires)

modéliser des empreintes de pas. L'interactivité, les contraintes de temps réels et la possibilité d'utilisation en environnements distribués sont des facteurs supplémentaires qui entrent dans le processus d'analyse. La Table 4-7 synthétise la pertinence que chaque article possède en fonction de ces facteurs.

En ce qui concerne les informations de la colonne sur les empreintes de pas, les seuls articles répondant aux besoins du projet sont ceux qui portent sur les impacts, et d'une façon plus discutable l'accumulation de poussière et les dégradations de types multiples. En effet, un impact en forme d'empreinte pourrait être une solution pour le projet et laisser une empreinte dans de la poussière également. Dans le cas des articles à dégradations multiples, les techniques utilisées permettraient de reproduire l'apparence d'empreintes de pas à partir d'échantillons d'empreintes tels que des photos ou par l'acquisition de textures à partir de vraies empreintes. Les articles portant sur la corrosion et les rayures et égratignures sont trop incompatibles avec un principe d'empreinte de pas pour être mentionnés. Des empreintes de pas ne peuvent causer de la corrosion, puis des égratignures sont des phénomènes d'une échelle différentes que les empreintes de pas.

Le temps réel permet d'écartier facilement les articles dont les temps de calcul se comptent en plus de quelques secondes. Une empreinte de pas se doit d'apparaître presque de façon instantanée pour simuler ce processus le mieux possible. La colonne temps réel rappelle que seul les articles sur les impacts, l'accumulation de poussière sur les surfaces transparentes et les dégradations multiples s'effectuent en quelques secondes.

Pour ce qui est du lien entre les travaux de recherche et les environnements distribués, la capacité des méthodes employées à faciliter la distribution d'informations entre des sites distants n'est pas abordée. En effet, l'objectif des travaux n'est pas de dégrader des scènes en temps réel et de communiquer des informations sur les dégradations entre des sites distants mais d'augmenter le réalisme des scènes. Comme la dernière colonne de la Table 4-7 le montre, tous les articles nécessiteraient des paramètres supplémentaires pour permettre de supporter les environnements distribués. Une empreinte de pas, en particulier, nécessite des paramètres supplémentaires de localisation qui doivent être transmis efficacement. Dans les travaux sur les impacts (Paquette et al., 2001) et sur les craquèlement de la peinture (Paquette et al., 2002), les masques pourraient être modifiés pour introduire la localisation des empreintes. Envoyer ces masques entre différents sites permettrait aux sites de les appliquer aux emplacements précis des empreintes. Un même ajout conviendrait aux approches de génération de texture. Appliquer l'emplacement des empreintes à ces dernières constituerait le travail supplémentaire de localisation. Dans les travaux sur l'accumulation de poussière, la

poussière pourrait ne pas être disposée à l'emplacement des empreintes. Cependant cela rendrait l'échange d'information plus complexe. Enfin dans les travaux sur l'écoulement d'eau et l'apparition de taches, lier l'écoulement d'eau à la localisation et la forme d'empreinte n'est pas très évident.

Table 4-8 Sélection des articles répondant le mieux aux objectifs du projet de maîtrise

Auteurs	Empreintes de pas		Temps réel	Environnements Distribués
	Lien	Technique		
Gu et al. (2006)	Empreinte dont l'apparence varie dans le temps	Capture de TSV-BRDF d'empreintes de pas	Temps de calcul de 3 à 11500 s	Discutable (paramètres supplémentaires)
Paquette et al. (2007)	Texture d'empreinte en entrée	Sélection de la zone de l'empreinte de pas	Temps de calcul de 4 à 65 s	Discutable (paramètres supplémentaires)
Lu et al. (2007)	Apparence de l'empreinte, mais pas la forme	Capture de paramètres géométriques et de textures	Temps de calcul de 1 à 10s	Discutable (paramètres supplémentaires)
Sun et al. (2007)	Empreintes de pas dans la poussière, mouillant le sol	Capture de BRDF appliquées à des matériaux	Temps de calcul de 15 à 900 s	Discutable (paramètres supplémentaires)
Gu et al. (2007)	Empreintes de pas dans la poussière	Ajout de « formes » à la répartition de poussière	Temps de calcul de 3 à 60 s	Discutable (paramètres supplémentaires)
Paquette et al. (2002)	Conséquence d'une empreinte dans un sol peint	Incorporation de la forme d'une empreinte à la carte	Temps de calcul de 180 s	Discutable (paramètres supplémentaires)
Endo et al. (2010)	Écoulement d'eau en lien avec une empreinte	Incorporation de la forme au modèle	Temps de calcul de 180 à 600 s	Discutable (paramètres supplémentaires)
Bosch et al. (2011)	Écoulement d'eau en lien avec une empreinte	Incorporation de la forme d'une empreinte	Temps de calcul de 60 à 120 s	Discutable (paramètres supplémentaires)
Paquette et al. (2001)	Impact en forme d'empreinte	Utilisation d'une forme d'empreinte	Temps de calcul de 4 à 8 s	Discutable (paramètres supplémentaires)

Tel que le montre la colonne sur l'interaction de la Table 4-7, l'interactivité peut intervenir le plus facilement dans les articles traitant de dégradations mécaniques telles que les rayures et égratignures ou les impacts. Les dégradations liées aux taches liées à l'écoulement d'eau, l'accumulation de poussières et le craquèlement de la peinture présentent des moyens

d'interactions plus indirects et discutables. L'écoulement d'eau serait provoqué par la pose d'empreintes d'utilisateur, puis des taches apparaîtrait à cet emplacement. Le craquèlement de peinture interviendrait lorsqu'un utilisateur marcherait sur un sol peinturé. La peinture craquèlerait donc à l'emplacement des empreintes. Concernant l'accumulation de poussière, un utilisateur pourrait laisser des empreintes derrière lui en marchant sur un sol poussiéreux.

Les articles présentant plus d'affinités au projet de maîtrise sont réunis dans la Table 4-8. Chaque article de la Table 4-8 peut être appliqué dans le contexte du projet de maîtrise d'une façon ou d'une autre. Cependant le projet imposerait des modifications majeures aux techniques ou aux dégradations employées. Les articles sur l'accumulation de poussière devraient prendre en compte la forme d'empreintes de pas. La modification de polygones suivant l'impact d'une empreinte de pas semble pouvoir s'appliquer au projet de maîtrise. La technique devrait être modifiée pour tenir compte de la forme d'empreintes de pas. Cependant donner des moyens d'interaction directs aux utilisateurs pour les provoquer serait nécessaire. De plus, le raffinement du maillage de polygones amène un problème majeur : le nombre élevé de polygones à traiter et afficher. Des pas mouillant le sol pourraient faire apparaître des tâches sur celui-ci, mais les temps de calcul sont trop long pour du temps réel. Le temps de calcul le plus raisonnable dans un contexte de temps réel ne doit pas dépasser quelques secondes. Les temps de calcul éliminent aussi la plupart des travaux inspirés de modèle de BRDF. Les articles sur les impacts, l'accumulation de poussière sur surface transparente et les générations de texture sont donc les seuls qui vont plus dans ce sens. Cependant ils ne sont pas adaptés à une utilisation en environnements distribués.

En conclusion, la Table 4-8 montre qu'adapter les dégradations et techniques de certains articles pourrait permettre de modéliser différents types d'empreintes de pas. L'incompatibilité de chacun de ces articles avec le projet de maîtrise pour les facteurs du temps réel et des environnements distribués conjoints impose de trouver une nouvelle méthode pour appliquer des empreintes de pas dans un environnement.

4.2 Premier prototype

Le premier prototype est une application graphique où des images de dégradation peuvent être collées dans la scène par de simples clics de la souris.

4.2.1 Analyse

4.2.1.1 Exigences fonctionnelles

Les exigences fonctionnelles sont d'afficher une scène, détériorer l'environnement et se déplacer dans l'environnement. La première exigence est d'afficher une scène avec des surfaces dégradables. Cette scène peut être constituée d'un sol et de murs. Détériorer l'environnement est également une exigence majeure. Il doit être possible de modifier l'apparence de l'environnement et de faire apparaître des dégradations sur celui-ci. Finalement le déplacement dans l'environnement doit simuler une exploration de celui-ci. En effet dans le projet « environnements susceptibles », les utilisateurs évoluent dans un environnement en laissant des traces derrière-eux.

4.2.1.2 Exigences non-fonctionnelles

Les exigences non-fonctionnelles sont l'utilisation d'OpenGL, de Qt et de la librairie libQGLViewer. OpenGL est l'interface de programmation choisie dans le cadre de beaucoup d'application graphiques 3D. Ici le projet « environnements susceptibles » induisait que l'application devait être implémentée en OpenGL pour faciliter l'intégration éventuelle avec OpenSceneGraph. L'utilisation de Qt est une exigence qui s'est présentée en réponse aux problématiques d'interaction pour détériorer l'environnement et se déplacer dans celui-ci. Le cadriel Qt propose beaucoup de possibilités d'interaction par souris et clavier qui peuvent être aisément intégrées à l'API OpenGL et ce par l'utilisation de la librairie libQGLViewer. Le prototype sera développé et testé sur un ordinateur portable muni d'un processeur Intel

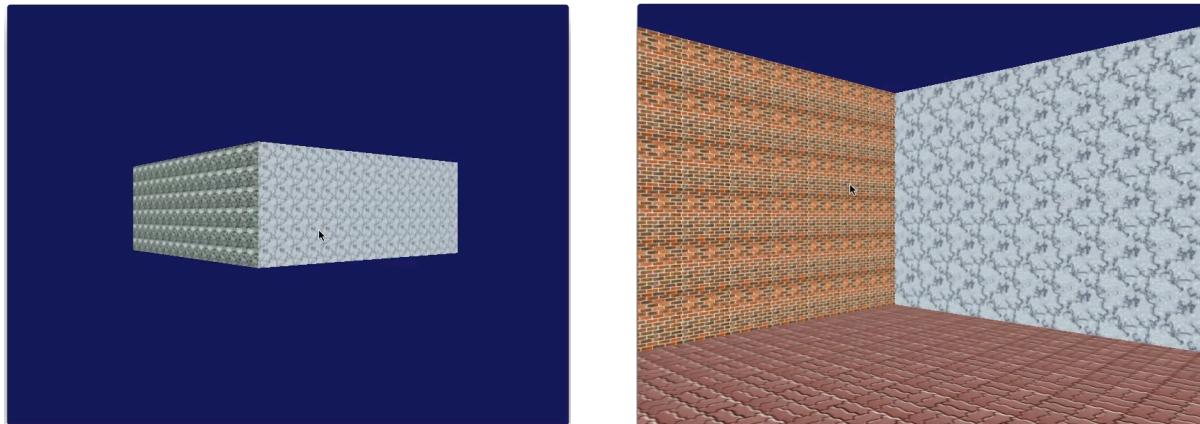


Figure 4–1 Aperçu de l'environnement du premier prototype

Core i5 à 2.3GHz de 4Go de mémoire vive et d'une carte graphique Intel HD Graphics 3000 de 384 Mo. La SAT dispose d'équipements plus puissants et variés. Elle possède des équipements plus performants que ceux-ci et l'application pourrait être déployée par exemple dans un Panoscope, dôme immersif muni de caméras. Travailler avec un ordinateur moins puissant tout en assurant un fonctionnement correct du prototype garantit donc une marge de manœuvre à la SAT qui combine souvent plusieurs projets dans un même système, ce qui requiert une plus grande puissance de calcul.

4.2.2 Conception

Le design de l'application repose sur une architecture simple. Une fenêtre de visualisation se charge de montrer la scène, ses objets, et les dégradations. Une interface où est intégrée la fenêtre de visualisation se charge d'intercepter les actions du utilisateur. La Figure 4–1 est un aperçu de l'environnement du premier prototype. La Figure 4-2 montre un diagramme UML du premier prototype.

4.2.3 Modélisation des dégradations

La modélisation des empreintes s'est faite par une application directe d'images de dégradation dans la scène. Les images sont stockées sous formes de textures qui sont plaquées sur des polygones.

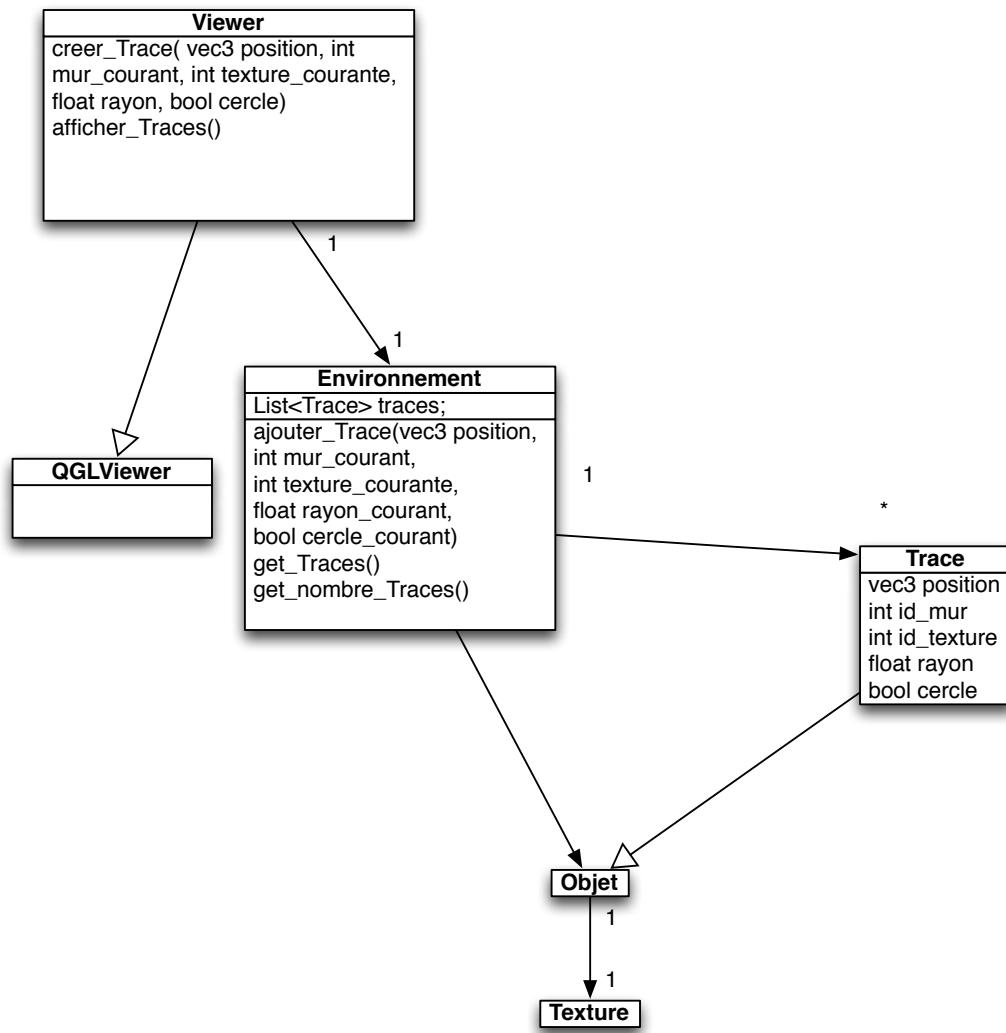


Figure 4-2 Diagramme UML (premier prototype)

Les images d'usures affichées dans la Figure 4-3 représentent des dégénérescences, de la rouille et des fissures.

Une dégradation a pour caractéristiques :

- Sa location dans la scène (deux nombres flottants)
- L'identifiant de la texture utilisée (un entier)
- Son rayon (un nombre flottant)
- Sa forme : un cercle ou un carré (un booléen)
- L'identifiant du mur sur lequel la dégradation est collée (un entier)

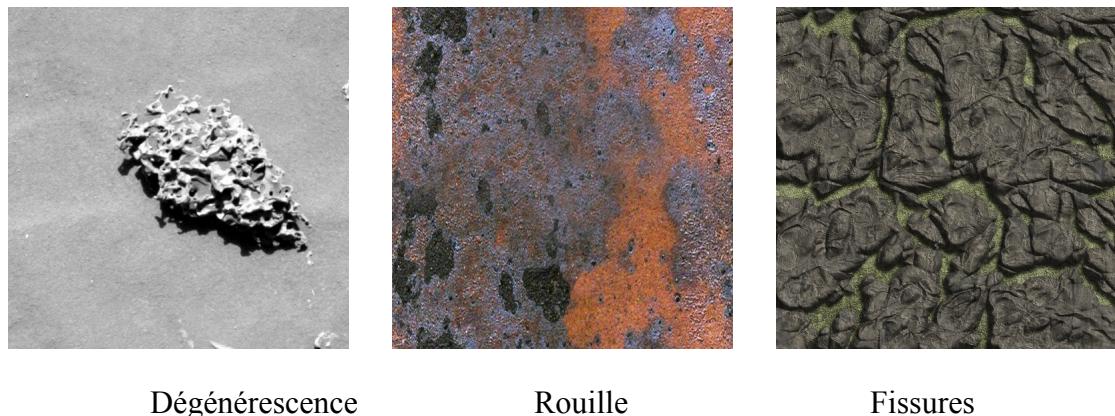


Figure 4-3 Images de dégradations

La Figure 4-4 montre l'environnement couvert de différents types de dégradations sur plusieurs murs. Les dégradations sont stockées dans la classe *Environnement* sous forme d'une liste d'objets *Trace*. La classe *Environnement* agit par conséquent en tant que modèle, et la classe *Viewer* en tant que visualisateur des données. La classe *Trace* contient un attribut par caractéristique.

Lors du réaffichage de l'environnement et pour chaque trace, le visualisateur retrouve la texture grâce à l'identifiant de texture associé et, en fonction de la valeur de forme, crée un polygone en forme de carré ou forme de cercle et y applique la dite texture.

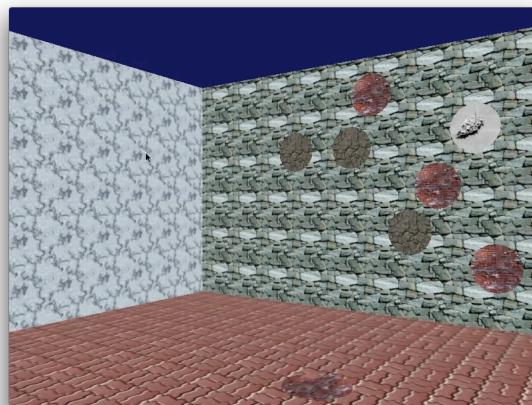


Figure 4-4 Environnement couvert de dégradations

4.2.3.1 La juxtaposition des dégradations dans l'environnement

Dans ce prototype, les dégradations sont « juxtaposées » dans l'environnement. L'application crée un polygone par dégradation qui est posé sur les murs de l'environnement. Une fois que l'application intercepte un clic de l'utilisateur sur une des surfaces de l'environnement, la position du point cliqué et des valeurs courantes sont envoyées à la fonction *créer_Trace()* du

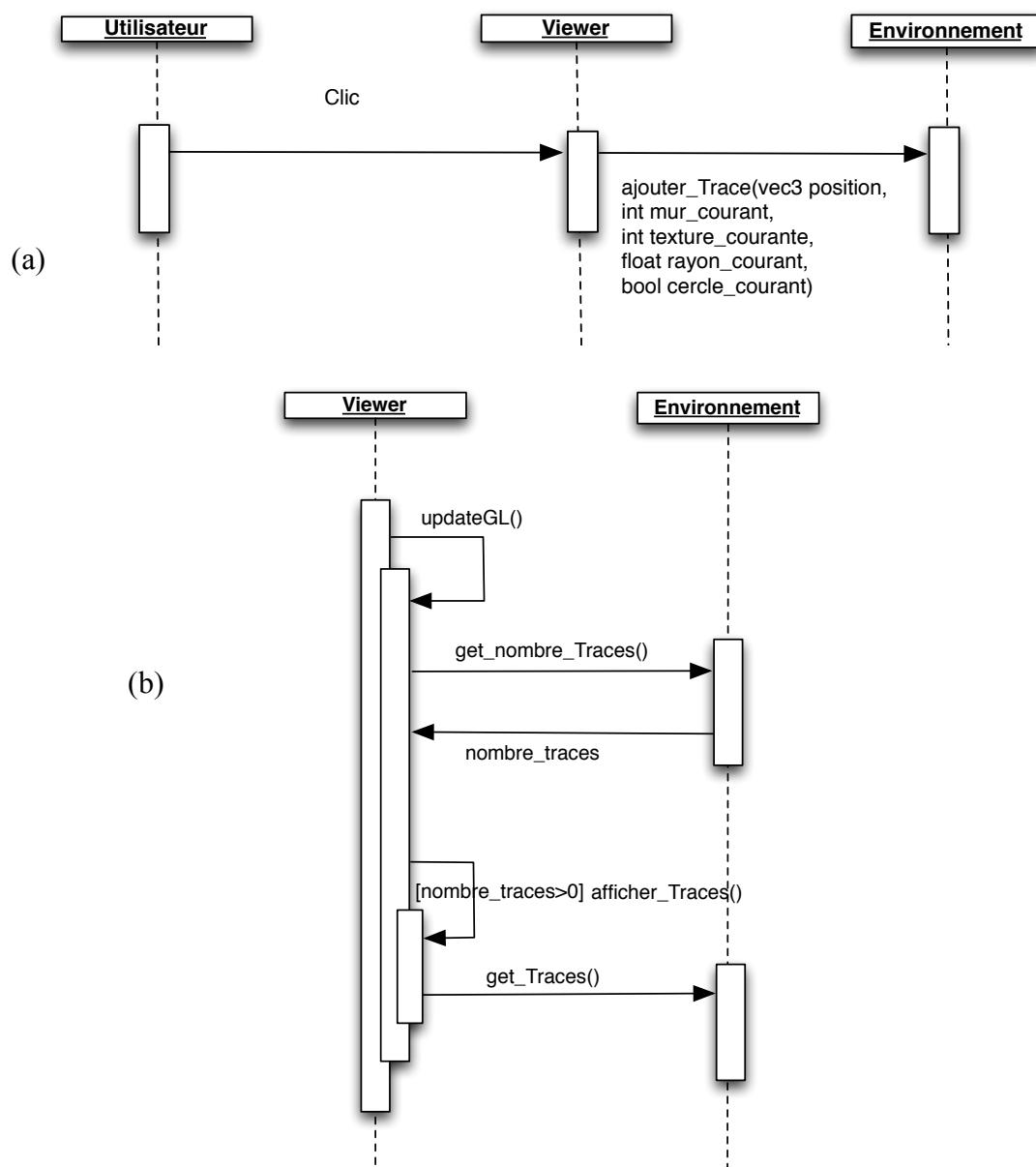


Figure 4-5 Diagrammes de séquence de l'ajout (a) et de l'affichage (b) d'une dégradation

Viewer. Dans le diagramme de séquence (a) de la Figure 4-5, l'ajout des dégradations à l'environnement est montré et dans le diagramme de séquence (b) de la Figure 4-5 l'affichage des dégradations est montré. Dans la Figure 4-5 (a), le visualisateur reçoit des valeurs dites courantes car elles stockent toutes les caractéristiques que l'utilisateur a choisi avant de générer une dégradation par clic. Ces valeurs sont la forme de la dégradation (un booléen égal à *false* dans le cas d'un carré et *true* dans le cas d'un cercle), son rayon et sa texture de dégradation. L'identifiant du mur de la dégradation est retrouvé à partir du point cliqué par l'utilisateur. Avec certaines touches du clavier, l'utilisateur peut changer la valeur courante de ces caractéristiques excepté pour l'identifiant du mur de la dégradation. La fonction *afficher_Traces()* du *Viewer* récupère le nombre de traces dans l'instance d'*Environnement*. Dans la Figure 4-5 (b), la méthode *updateGL()* entraîne le rafraîchissement de la scène. Si le nombre de traces est strictement positif, le visualisateur récupère les traces de l'environnement. Il génère ensuite un polygone par dégradation très légèrement en avant de la surface d'un mur, donnant l'impression que la dégradation se fond dans la surface. L'orientation est calculée lors de l'affichage de la dégradation en fonction de l'identifiant de mur stocké dans l'instance d'objet. Le calcul de l'affichage de la forme intervient aussi à ce moment. Les sommets des polygones générés autour de la position de la dégradation suivent une forme circulaire dans le cas où le booléen de forme circulaire est à *true*. La texture qui est plaquée sur le polygone est retrouvée grâce à l'identifiant de texture de l'objet de *Trace*, identifiant correspondant à celui de la texture en mémoire.

4.2.4 Type d'interaction

4.2.4.1 Déplacement dans l'environnement

La librairie libQGLViewer fournit un mode déplacement natif associé au déplacement de la caméra de sa classe *Viewer*. Tout *Viewer* offre un déplacement par défaut grâce à la souris. Ce mode de déplacement se divise en trois modes. Le premier, enclenché par la molette de la souris correspond à un zoom. Il est très fluide, mais ne permet pas d'avancer ou reculer dans la scène. Les deux autres modes de déplacement, nécessitant le maintien du clic de la souris,

permettent de tourner autour de l'origine ou de se déplacement latéralement à un endroit donné. Pour limiter les rotations à des rotations vers la gauche et la droite, des contraintes de rotation ont été ajoutées sur la caméra pour qu'elle ne puisse se mouvoir que selon son plan X et Y. Le mouvement de zoom utilisé pour donner l'impression d'avancer et reculer dans l'environnement est parfois trop rapide. Un second mode a été implémenté pour résoudre ce problème. Il utilise les flèches du clavier pour réaliser un zoom. La fluidité trouvée dans le premier mode disparaît ici et le déplacement est davantage incrémentiel. En effet il dépend de la fréquence d'interception des événements du clavier qui fonctionne à 60Hz.

4.2.5 Point négatifs et positifs

4.2.5.1 Fonctionnel

Ce prototype était une bonne façon d'entrer dans le sujet. Il est très basique et permet d'apposer des dégradations dans le décor de façon simple. La forme des images de dégradation prend la forme d'un carré ou d'un cercle, l'orientation change en fonction du mur qui est affecté, et le rayon également peut être changé.

4.2.5.2 Discontinuité entre les empreintes et l'environnement

Un gros désavantage est que ces dégradations ne sont pas reproduites avec réalisme dans l'environnement. La technique utilisée pour la modification de l'environnement n'est pas intéressante car les dégradations sont représentées par des images collées sur les surfaces de la scène. La juxtaposition des dégradations les unes sur les autres entraîne souvent un problème d'aliassage (*Z-fighting*). De plus, le prototype détecte les murs où les dégradations s'appliquent, mais il ne détecte pas les bords des murs et les dégradations sont susceptibles de déborder dans le vide. La Figure 4-6 (a) expose un problème d'aliassage des différentes dégradations superposées, la Figure 4-6 (b) et la Figure 4-6 (c) désignent le débordement d'une dégradation à la jonction de deux murs.

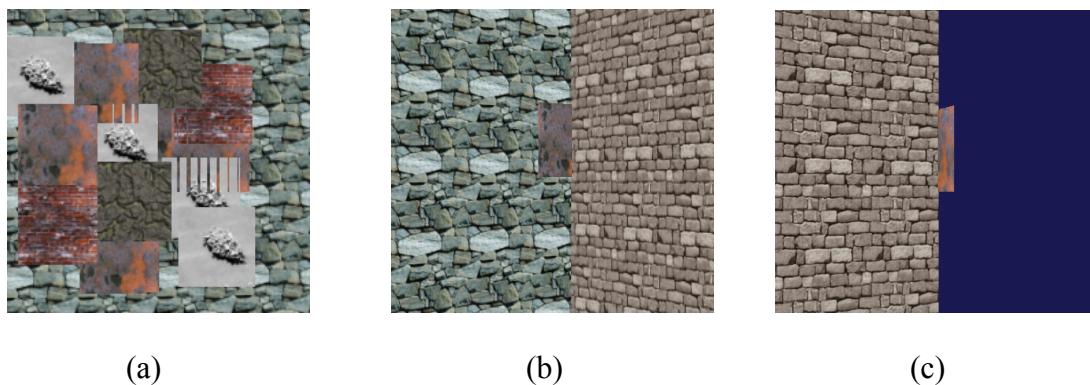


Figure 4-6 Aliassage et débordement de dégradation

4.3 Deuxième prototype

Le premier prototype était une première ébauche de dégradation de l'environnement interactif et en temps réel. Les lacunes de ce premier prototype et les rencontres avec la Société des arts technologiques ont entraîné une redéfinition des exigences du prototype. Les problématiques concernant les types de dégradations, leur application, l'interaction avec l'utilisateur et les contraintes de temps réel sont approchées autrement.

4.3.1 Analyse

4.3.1.1 Exigences Fonctionnelles

Les exigences fonctionnelles pour le second prototype sont d'afficher une scène, d'afficher des empreintes de pas, de se déplacer dans la scène. L'exigence d'afficher une scène est plus influencée ici par les dégradations sous formes d'empreintes de pas. Les empreintes peuvent s'appliquer sur un sol courbé, mais les empreintes ne sont que sur le sol. Les empreintes de pas ont été la dégradation choisie lors d'une réunion d'équipe avec la SAT. Elles représentent la trace immédiate que laisse un être humain dans son sillon et pourraient être facilement reconnaissable par une tierce personne dans l'environnement.

Le déplacement dans l'environnement avec le zoom de la souris était fluide mais il ne correspondait pas exactement à un déplacement. Le déplacement avec les touches multidirectionnelles du clavier, qui était saccadé dans le premier prototype, doit devenir plus fluide.

4.3.1.2 Exigences non-fonctionnelles

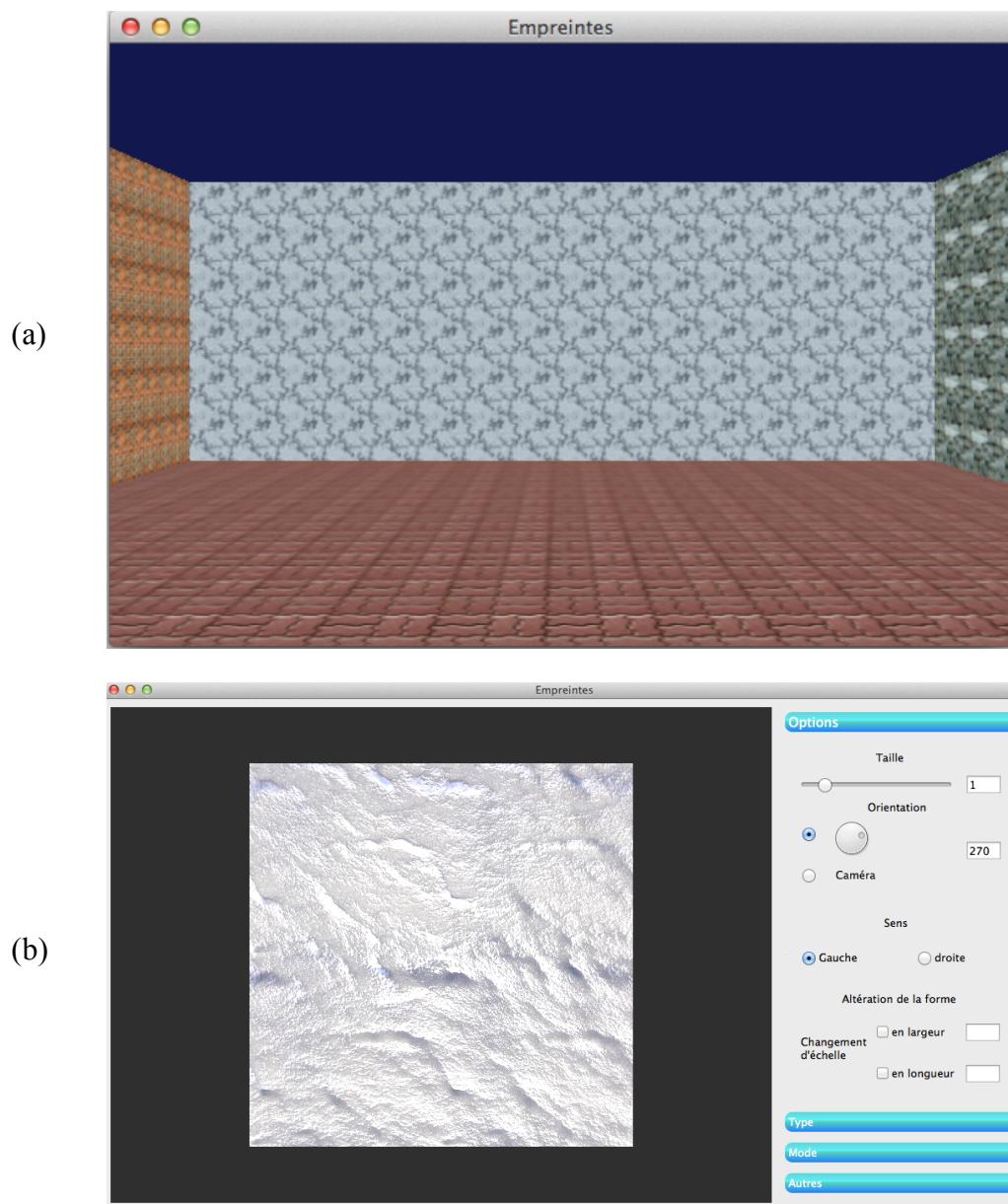


Figure 4-7 Interfaces des deux prototypes

Les nuanceurs sont une exigence qui permet d'approfondir le travail sur l'application de la dégradation dans l'environnement. Ils permettent de travailler fragment par fragment, pixel par pixel et donnent accès à l'usage de techniques plus complexes que la simple utilisation des textures dans le premier prototype. Le prototype sera développé et testé sur le même

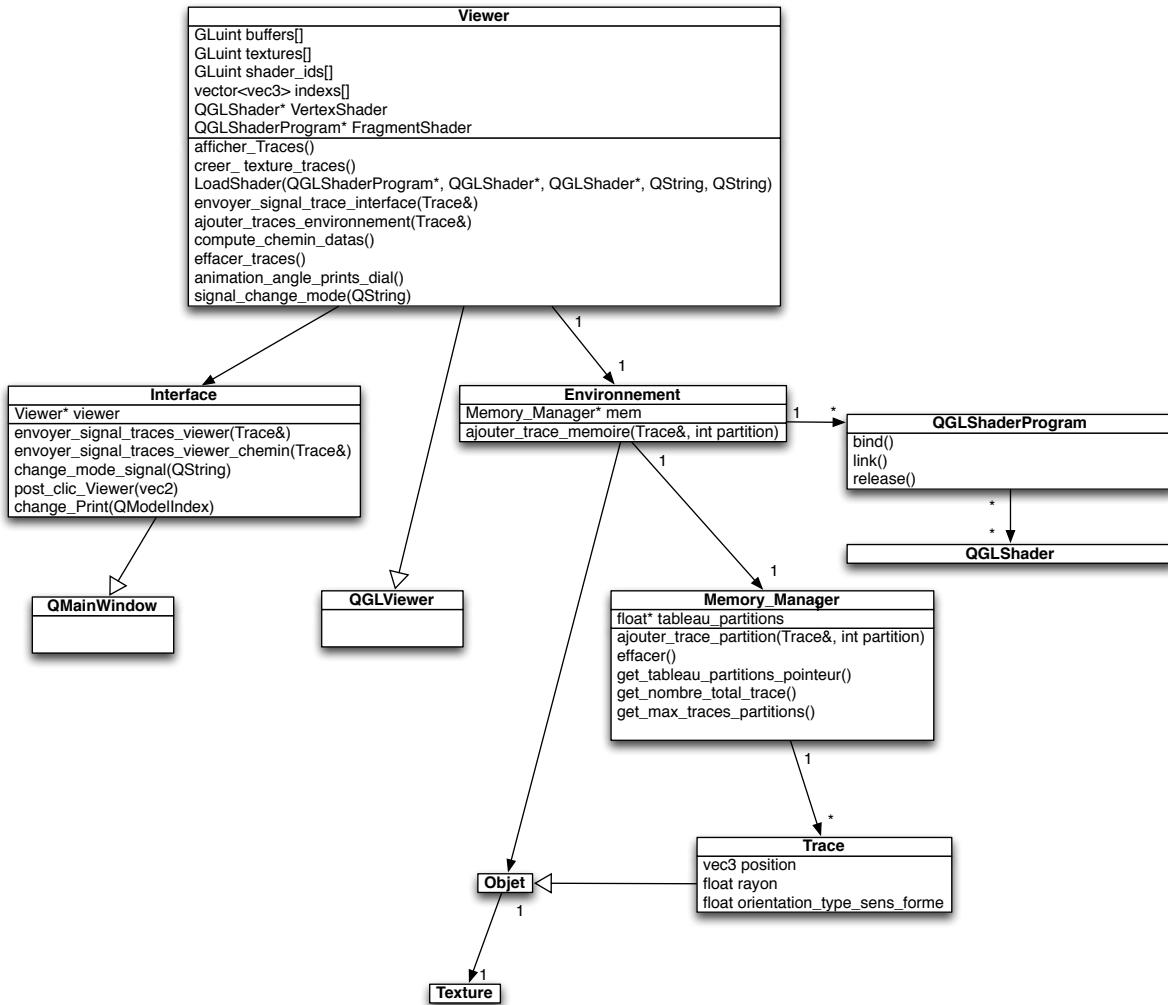


Figure 4-8 Diagramme UML (deuxième prototype)

équipement que pour le premier prototype. En calibrant le prototype sur un ordinateur moins puissant, le projet sera plus facilement intégrable dans le système final à la SAT, où la charge de calcul sera plus élevée à cause des traitements additionnels. Ces traitements additionnels proviendront par exemple de calculs sonores et d'un environnement plus riche.

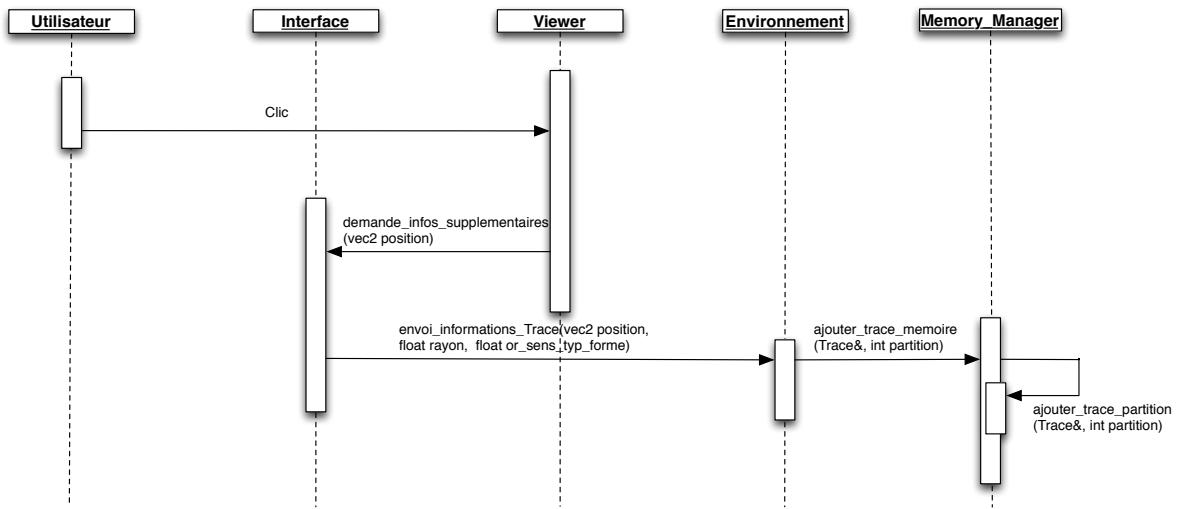


Figure 4-9 Diagramme de séquence d'ajout d'empreintes (deuxième prototype)

4.3.2 Conception

4.3.2.1 Approche du design

L'approche est restée la même mais ici, l'interface utilisateur étant plus complexe, il a fallu la séparer de la fenêtre de visualisation. La Figure 4-7 présente l'interface du premier prototype en (a) face à celle du deuxième prototype en (b).

La Figure 4-8 représente le diagramme UML du deuxième prototype. Le visualisateur devient associé à l'interface où l'utilisateur peut modifier des paramètres d'empreintes et l'accès aux données d'empreintes se fait par le biais de la classe *Memory_Manager*.

La Figure 4-9 montre le diagramme de séquence menant au stockage d'empreinte de pas dans la mémoire. Le visualisateur intercepte le clic de l'utilisateur et retrouve la position du clic dans la scène. L'interface utilisateur contient des informations supplémentaires sur les empreintes. Pour cette raison, le visualisateur communique la position du clic à l'interface. L'interface peut alors envoyer à l'environnement toutes les informations nécessaires pour ajouter une trace en mémoire. L'environnement retrouve la partition où le clic est localisé en comparant la position du clic aux coordonnées des partitions.

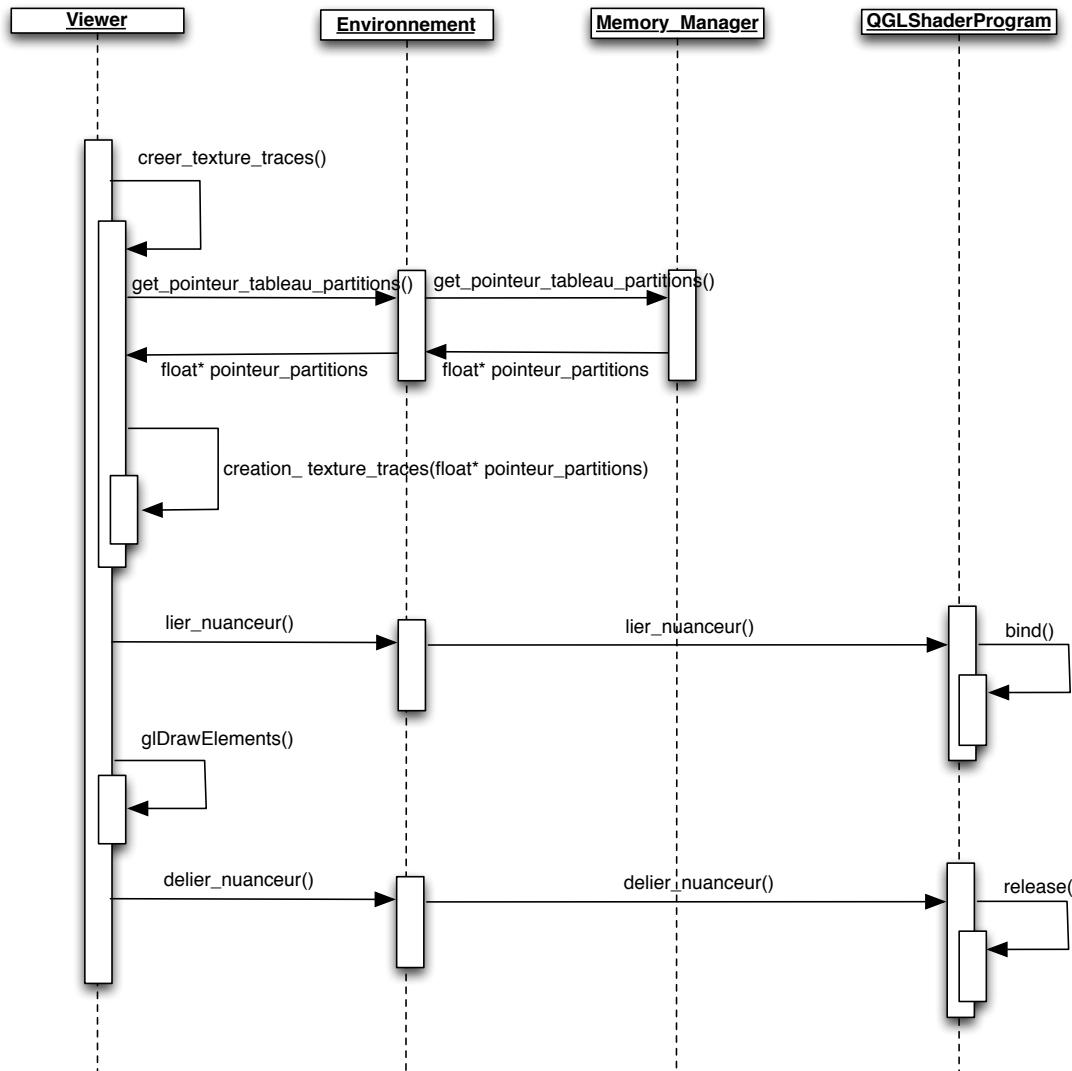


Figure 4-10 Diagramme de séquence de l'affichage des empreintes (deuxième prototype)

Dans la Figure 4-10, un diagramme de séquence montre le processus de modélisation des empreintes. Le visualisateur charge un tableau de nombres à points flottants contenant les informations des empreintes de pas par le biais du *Memory_Manager* et crée à partir du tableau une texture d'informations. Le nuanceur est lié à la suite de la création de cette texture pour pouvoir en lire le contenu et afficher les empreintes et la scène.

4.3.3 Technologies utilisées et remplacées

4.3.3.1 Continuité avec l'environnement de développement

OpenGL est resté l'API de rendu graphique, mais des fonctions plus complexes ont été sollicitées. Qt est resté l'outil privilégié pour l'interaction utilisateur. Qt a, dans ce prototype, été plus sollicité de par ses possibilités de facilement ajuster certains paramètres dans l'application ou changer de modes d'interaction. La librairie libQGLViewer continue d'établir un pont entre OpenGL et Qt. Dans ce deuxième prototype, l'outil de gestion de caméra de libQGLviewer est plus approfondi.

4.3.3.2 Remplacement de fonctionnalités OpenGL par l'utilisation de nuanceurs

Dans le premier prototype, les transformations sur les sommets et l'éclairage étaient calculés automatiquement par OpenGL. Dans le deuxième prototype, l'utilisation des nuanceurs impose de réaliser les calculs de sommets et d'éclairage dans le programme de nuanceur de l'application, ce qui complexifie l'application.

Sous OpenGL, l'éclairage est calculé seulement aux sommets et chaque fragment reçoit une valeur erronée surtout en présence de réflexion spéculaire. Cela peut donner naissance à des phénomènes d'aliassage. Les nuanceurs permettent d'utiliser des techniques d'éclairage plus précises et complexes car ils effectuent des calculs de luminosité pour chaque fragment. La technique du *normal mapping* a été choisie pour rehausser l'allure de la surface. Cette technique calcule l'éclairage de chaque fragment d'un polygone en fonction de la normale fournie pour le polygone et de la normale correspondante issue d'une texture de normales. Cette texture de normales permet de perturber localement la normale de la surface pour imiter des détails de surface à une échelle plus fine que celle des polygones. Le *normal mapping* permet ainsi d'accentuer le relief des éléments d'une surface et ce sans rajouter de polygones supplémentaires.

De plus l'utilisation de nuanceurs élimine le besoin de « juxtaposer » des polygones texturés sur l'environnement comme c'était le cas dans le premier prototype, et permet de modifier directement les polygones de la scène. En effet chaque fragment de chaque polygone de la scène suit les calculs des nuanceurs de fragments. Changer la couleur d'un fragment en fonction d'une proximité arbitraire à une position d'une empreinte devient possible. De plus la texture d'empreinte de pas est utilisée dans le nuanceur. Un premier test vérifie la proximité du fragment traité à la position d'une empreinte. La position de l'empreinte est entourée d'un quadrilatère dont la taille varie en fonction du rayon fixé par l'utilisateur lors de l'ajout de l'empreinte. Dans le cas où le fragment se trouve dans le quadrilatère contenant

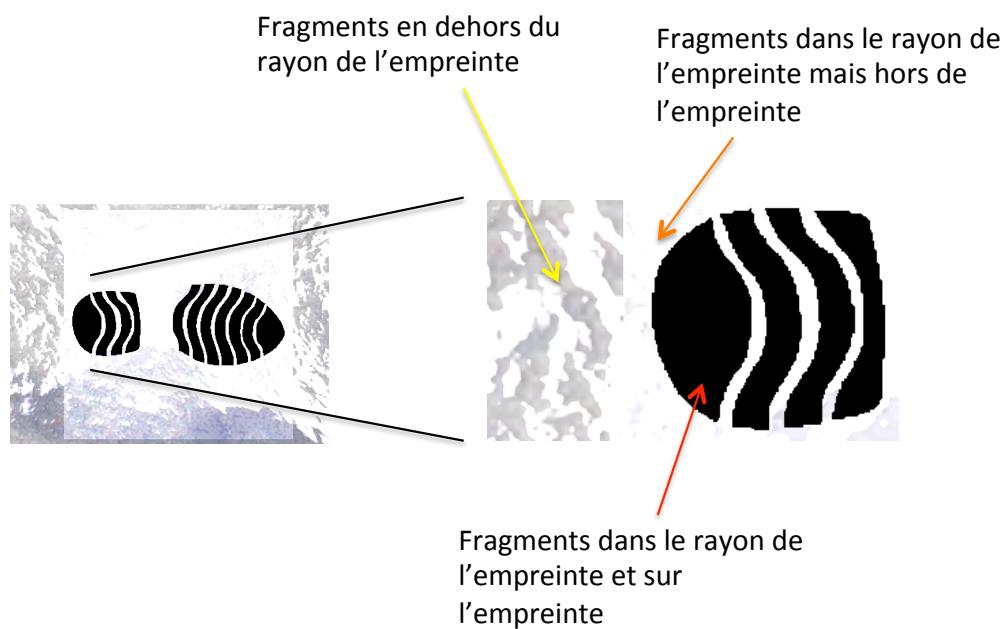


Figure 4-11 Comparaison de fragments dans une empreinte de pas

l'empreinte, un second test consiste à ne changer la couleur du fragment que si le texel lui correspondant dans la texture d'empreinte de pas fait bien partie de l'empreinte. Pour faciliter ce calcul, chaque texture d'empreinte de pas représente une empreinte noir sur un fond blanc. Ce test se réduit à ne copier la couleur d'un texel que si ses valeurs RGB se rapprochent du noir. La figure Figure 4-11 met en évidence les différents cas que les fragments peuvent remplir. Les fragments se trouvant en dehors du quadrilatère blanc de l'empreinte sont

pointés par la flèche jaune et ils sont ignorés par le test sur la proximité par rapport à la position de l'empreinte. Les fragments se trouvant dans la zone en surbrillance sont pointés par la flèche orange et vérifient le test de proximité à la position de l'empreinte, mais ils ne vérifient pas le test consistant à savoir s'ils font partie de l'empreinte ou pas. Les fragments pointés par la flèche rouge prennent la couleur du texel correspondant dans la texture d'empreinte car ils vérifient les deux tests.

4.3.4 La modélisation des traces

Les empreintes de pas ont été choisie comme type de dégradation de l'environnement par l'équipe de la SAT. Dans le deuxième prototype, les empreintes de pas ont davantage de caractéristiques que dans le premier prototype :

- Position suivant deux dimensions (sur le sol)
- Identifiant pour la texture d'empreinte de pas utilisée
- Rayon
- Sens (empreinte du pied gauche ou droit)
- Orientation
- Modification de la forme verticale ou horizontale



Figure 4-12 Variation des paramètres d'une empreinte

La **Figure 4-12** montre une empreinte où chaque paramètre est modifié. Dans l'ordre sont montrés une empreinte de référence (a), une empreinte dont le rayon est réduit (b), une empreinte dont le sens est changé (c), une autre où l'orientation est modifiée (d) et enfin une empreinte deux fois plus large et deux fois moins haute (e). La **Figure 4-13** montre une combinaison de ces paramètres avec plusieurs types d'empreintes.

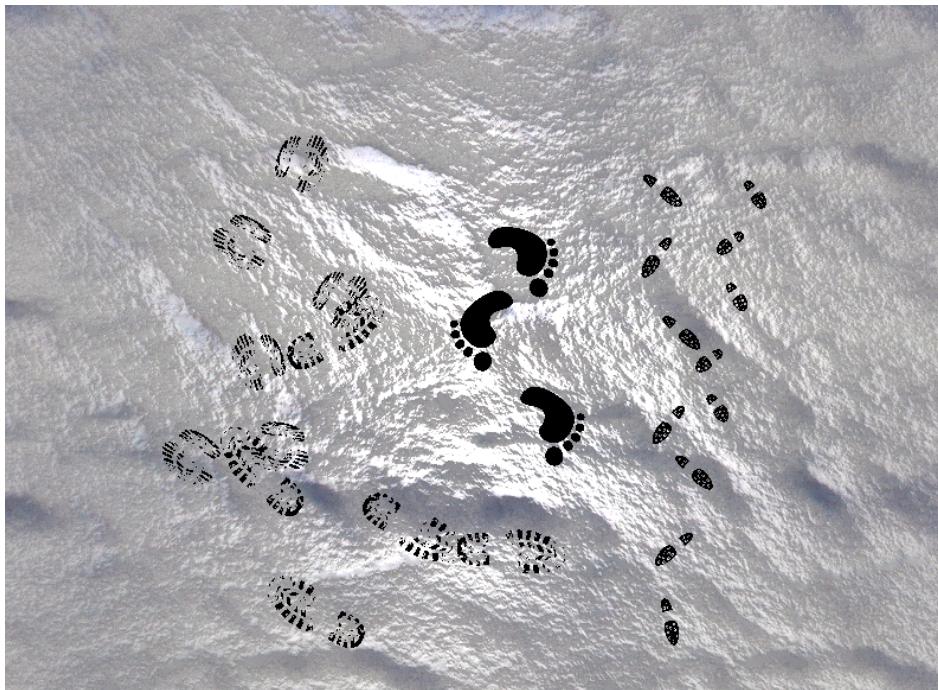


Figure 4-13 Application d'empreintes de formes, orientations, tailles, et types différents

4.3.4.1 Utilisation des nuanceurs dans la modélisation des traces

La modélisation des traces par nuanceurs est plus efficace que celle sous mode immédiat dans le premier prototype. Toute l'information est copiée une seule fois sur le GPU, ce qui permet un réaffichage plus rapide car l'information n'a pas à être retransmise à chaque réaffichage. Dans le cas du mode immédiat d'OpenGL, les informations sont retransmises du CPU au GPU à chaque réaffichage.

Une texture est créée contenant toutes les empreintes et leurs informations. Au lieu de contenir des valeurs sur 8 bits comprises entre 0 et 255 par texel, et par canal de couleur R, G, B et A,

chacun de ces canaux peut contenir un nombre en virgule flottante représenté sur 32 bits. Cette texture n'est pas une texture habituelle. Les textures habituelles sont des éléments qui reçoivent en données des images dont les dimensions en largeur et en hauteur sont les même et de préférence des puissances de deux. La texture utilisée ici est une texture rectangulaire. Ce type de texture peut prendre n'importe quelle dimension. L'aspect rectangulaire est clé car il permet d'ajuster la taille de la texture au nombre d'empreintes générées dans l'environnement. Chaque canal contient un ou plusieurs paramètres d'empreinte et une empreinte occupe l'espace d'un texel. La Figure 4-14 illustre ce principe en décortiquant un texel représentant une empreinte.

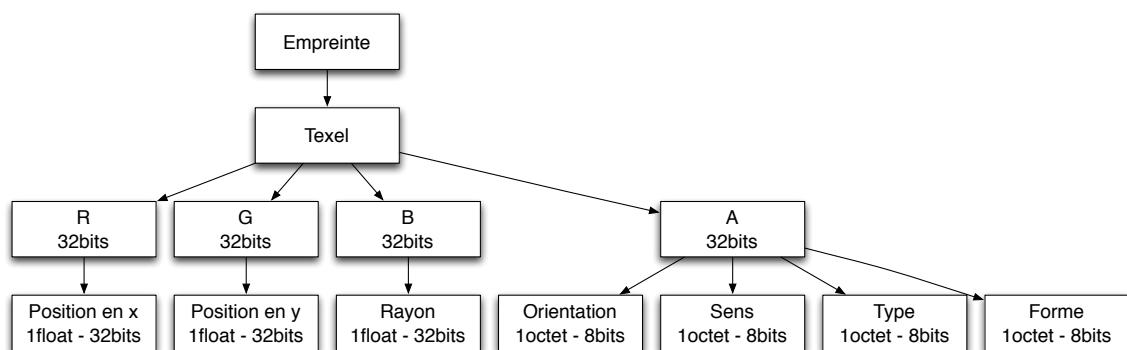


Figure 4-14 Représentation d'une empreinte dans un texel

La texture est envoyée au nuanceur qui la lit colonne par colonne comme un tableau. Pour chaque fragment, la proximité à une empreinte est testée. Si un fragment se trouve dans le rayon d'une empreinte, le nuanceur lui affecte la couleur correspondante de l'empreinte. Le rayon peut changer avec la modification de la forme de l'empreinte, son orientation, sa forme verticale et horizontale.

4.3.5 Amélioration du stockage de l'information

Le stockage de l'information sur les empreintes a été optimisé dans ce deuxième prototype et a permis de faire cohabiter un grand nombre d'empreintes simultanément sans ralentir l'application. Dans une première phase, il existait une liste qui comprenait toutes les

empreintes de l'application. Cela signifiait que dans le nuanceur, chaque fragment devait tester chaque empreinte comprise dans la liste. Avec une partition de l'environnement, l'objectif était de réduire ce test en ne testant que les empreintes proches du fragment.

4.3.5.1 Partition de l'environnement

La scène est composée essentiellement d'un sol texturé ayant la forme d'un carré. Ce carré est divisé suivant un carré parfait que l'utilisateur peut indiquer : 1x1, 2x2, 3x3, 4x4, etc. Chaque carré qui en résulte possède la même surface.

4.3.5.2 Liste d'empreintes par partition

Avant de pouvoir lire les texels de la texture d'empreintes, le programme doit transférer les informations d'empreintes du CPU au GPU. Pour chaque partition une « liste d'empreintes » existe en mémoire CPU et y est représentée par une colonne dans un tableau. La première ligne du tableau comprend pour chaque colonne d'empreintes des informations géométriques sur la position de la partition et le nombre d'empreintes qu'elle comprend. Le nombre de colonnes du tableau correspond à la taille de l'information d'une empreinte (*4 floats*) par le nombre de partitions de la scène (le carré parfait entré au lancement du programme). La hauteur est égale au plus grand nombre d'empreintes pouvant être trouvé dans une des partitions plus un pour la première ligne du tableau. Le tableau comprend donc des colonnes où certains emplacements sont vides. Dans la Figure 4-15, les quatre colonnes sont équivalentes aux quatre partitions de l'environnement. La hauteur est de six nombres en point flottant. En effet la partition trois contient cinq empreintes, et ce chiffre est le plus grand nombre d'empreintes dans une partition. La création, l'accès au tableau et sa mise à jour sont gérés par la classe *Memory_Manager*.

Un problème intimement lié à l'utilisation de la STL imposait d'avoir un tableau de nombre à points flottants plutôt que des listes ou vecteurs pour stocker les informations d'empreintes. OpenGL requiert un tableau contigu pour former une texture, ce que peut garantir un tableau

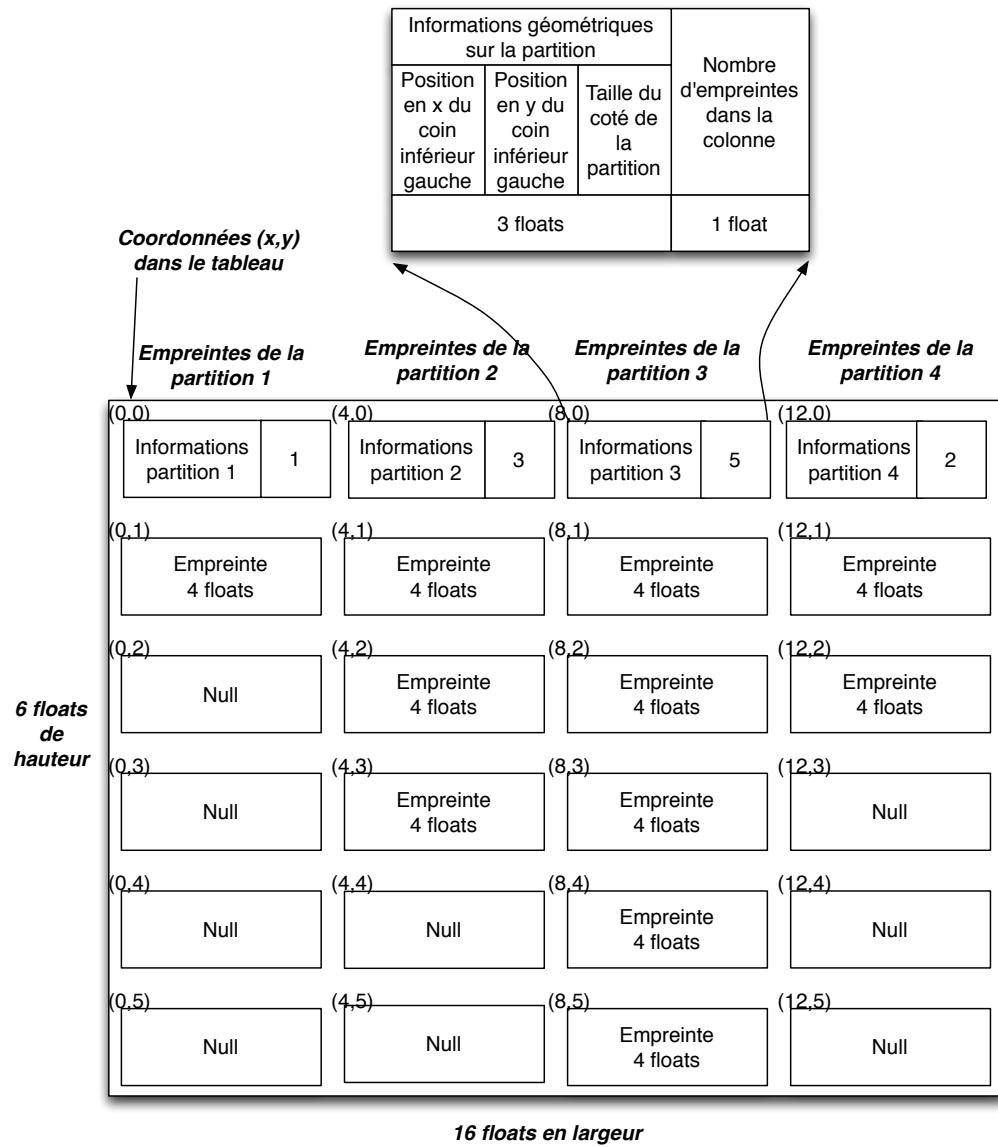


Figure 4-15 Représentation d'un tableau pour 4 partitions et 11 empreintes

de nombre à points flottants. Les listes ou vecteurs proposés par la STL auraient pu servir à stocker les informations d'empreintes. Cependant ils ne sont pas contigus en mémoire.

La classe *Memory_Manager* se charge de remplir le tableau d'empreintes en fonction de la partition où l'utilisateur a cliqué l'utilisateur. Lors de son initialisation elle remplit la première ligne du tableau avec, pour chaque partition, les coordonnées du coin inférieur gauche, la taille du côté de la partition et le nombre d'empreintes dans cette partition. Les informations géométriques de cette première ligne permettent au nuanceur de localiser les

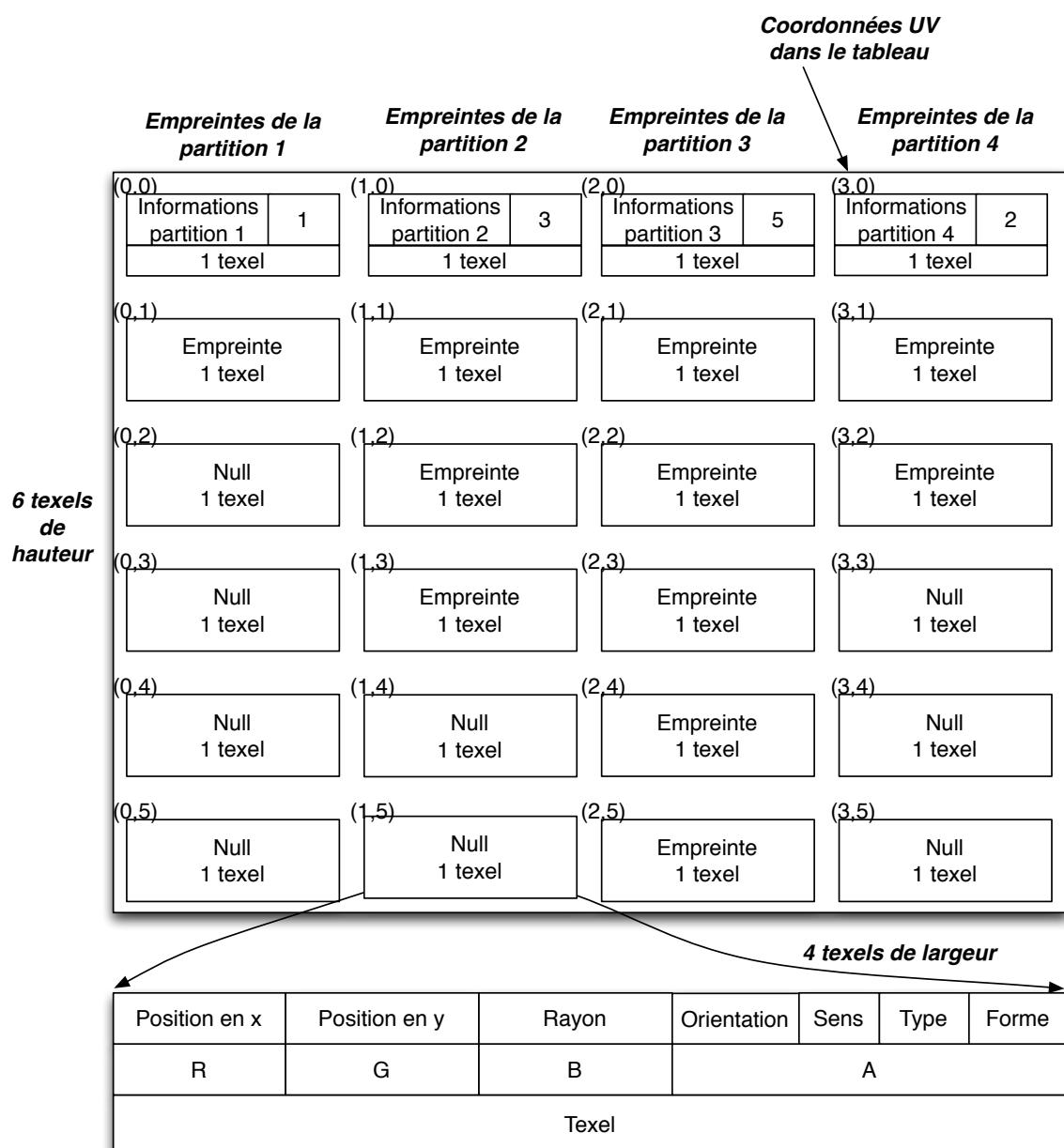


Figure 4-16 Texture associée au tableau de la Figure 4-15

fragments dans les partitions. Les lignes suivantes contiennent les informations sur les empreintes. Si l'utilisateur veut poser une empreinte sur la scène, la partition où l'utilisateur a cliqué est retrouvée à partir des coordonnées du point cliqué, le *Memory_Manager* enregistre les informations de la nouvelle empreinte dans la colonne correspondante et met à jour le nombre d'empreintes de la colonne. Trouver la colonne à mettre à jour, et de ce fait la zone du tableau qu'il faut remplir, correspond sensiblement au même test utilisé pour subdiviser la scène. En effet si la scène est divisée en 4 régions, un tableau d'une largeur de 4 colonnes d'empreintes sera créé. Si l'utilisateur clique dans la partition en bas à gauche de la scène, partition qui correspond à la première partition de la scène, le *Memory_Manager* mettra les informations d'empreinte dans la première colonne du tableau.

Le tableau a une taille fixe qui est limitée à la dimension maximale d'éléments d'une texture en hauteur. D'autre part la texture d'empreintes est réinitialisée à chaque fois que l'utilisateur crée une empreinte, et quand sa hauteur change. La hauteur correspond au plus grand nombre d'empreintes qui se trouvent dans une des listes d'empreintes plus un pour un texel d'information sur les coordonnées de la partition et son nombre. Par exemple, la texture affichée dans la Figure 4-16 représente le tableau de nombres à points flottant dans la Figure 4-15. Cette texture a quatre texels de largeur et six texels de hauteur. La troisième partition contient cinq empreintes, ce qui correspond à cinq texels dans la texture, puis les informations de coordonnées de la partition et de nombre d'empreinte dans cette partition ajoutent un texel en hauteur dans chaque colonne. Le contenu du texel d'une empreinte est identique à celui décrit dans la Figure 4-14.

Le test de localisation dans le nuanceur

Le nuanceur a accès au nombre de partitions dans l'environnement et à la texture d'empreinte. Grâce aux coordonnées de chaque fragment, il peut retrouver dans quelle partition de la scène il se trouve et par conséquent quelle colonne de la texture il faut explorer. Pour ce faire un test de localisation est effectué pour tous les texels de la première ligne de la texture, ces texels contenant des informations géométriques sur les partitions de

l'environnement. Si le fragment réussit le test pour un texel, le fragment sera alors testé avec toute les empreintes de cette colonne de texels.

4.3.6 Type d'interaction

Plusieurs types d'interaction ont été développés dans l'application. Ils permettent surtout de se déplacer dans l'environnement et d'y apposer des empreintes.

4.3.6.1 Déplacement par un contrôle de la caméra

Le déplacement dans l'environnement est simulé par le déplacement de la caméra. Le programme intercepte des évènements utilisateurs tels que l'appui des flèches directionnelles pour avancer reculer se tourner à gauche et se tourner à droite et le mouvement de la souris avec le clic gauche enfoncé pour se tourner vers le haut et se tourner vers le bas.

Chaque déplacement correspond soit à une translation, soit à une rotation. Dans le premier prototype, garder une touche enfoncée donnait un mouvement un peu saccadé, car le taux d'interceptions des évènements repose sur la fréquence à laquelle l'application les intercepte. La librairie libQGLViewer propose un outil qui permet d'interpoler le mouvement de la caméra entre deux positions, technique finale utilisée pour un déplacement plus fluide.

4.3.6.2 Mode d'interaction par clic

Ce deuxième mode d'interaction est très utile pour le débogage de l'application et pour pouvoir étudier une empreinte de près. Dans ce mode, chaque clic que l'utilisateur fait dans la scène génère une empreinte et ce, suivant plusieurs paramètres qu'il peut spécifier dans l'interface graphique. La Figure 4-17 montre cette interface et les différentes options proposées. Ces options sont la taille, l'orientation, le sens du pied (gauche ou droite), un changement d'échelle en largeur ou hauteur de l'empreinte, et la texture de l'empreinte en

elle-même. Ce mode d'interaction était le meilleur moyen pour tester le bon fonctionnement de l'application option par option.

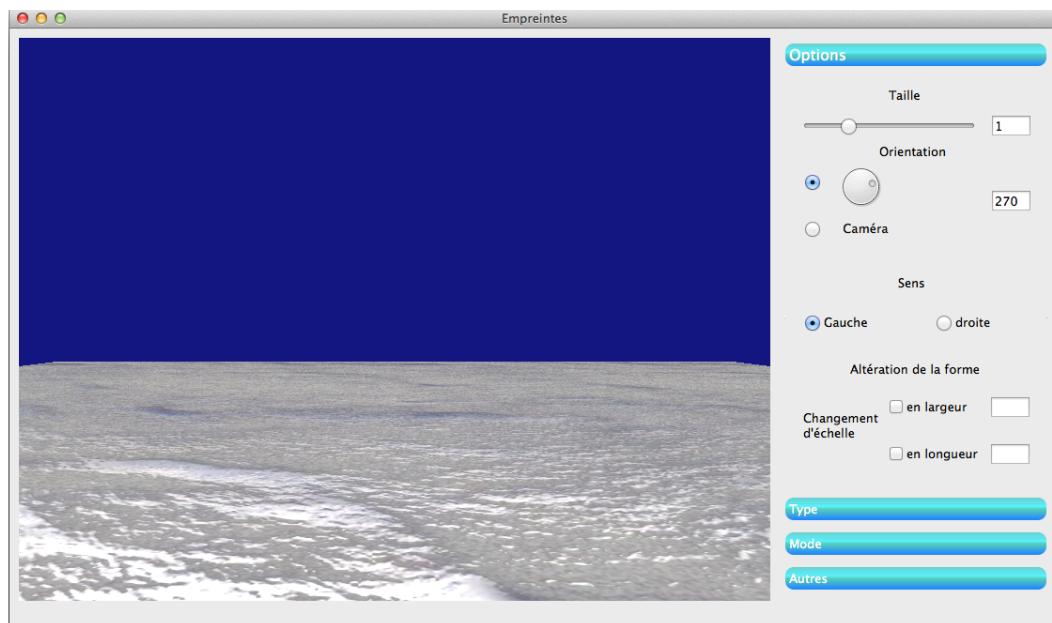


Figure 4-17 Interface sous le mode d'interaction par clic

Dans le programme, l'évènement de cliquer sur la scène est intercepté par des fonctions de la librairie Qt. Les mécanismes de signaux/slots conduisent à l'appel d'une fonction par la fenêtre de visualisation qui envoie toutes les données d'empreintes récupérées au gestionnaire d'empreinte.

4.3.6.3 Mode d'interaction par le suivi du chemin de la souris

Ce mode d'interaction est plus intuitif et est plus proche de l'approche du déplacement dans l'environnement. En maintenant le clic gauche enfoncé et en déplaçant le curseur dans la scène, il est possible de faire apparaître un sillon de pas qui suit la trajectoire de ce déplacement. Ici les options que l'utilisateur peut spécifier sont plus limitées : l'orientation et le sens du pied sont directement liés à la trajectoire que suit le curseur. Il est toutefois possible d'indiquer la taille souhaitée pour les empreintes, un changement d'échelle, l'image

de l'empreinte et aussi un léger changement d'orientation – comme montré dans la Figure 4-18. Les empreintes de pas peuvent être davantage orientées vers l'extérieur du sillon ou l'intérieur. Les possibilités de combinaison de ces paramètres se multiplient et différents types de chemins peuvent être dessinés, ce que la Figure 4-19 montre.



Figure 4-18 Orientation d'empreintes vers l'extérieur et l'intérieur du chemin

L'approche de conception est de trouver la trajectoire du mouvement de la souris et de dessiner de part et d'autre des empreintes. La première étape de cette opération est de récupérer les coordonnées de points que le curseur laisse derrière lui chaque fois que la souris parcours une distance égale à la taille du rayon de l'empreinte. Ces coordonnées sont les points verts sur la Figure 4-20. Lors de cette opération, chaque événement de la souris est traité. Les distances entre chacune des positions de ces événements s'additionnent jusqu'à atteindre la valeur de du rayon de l'empreinte. Souvent le mouvement de la souris n'est pas régulier, raison pour laquelle certains points verts sont plus proches que d'autres. La génération de vecteurs verts consécutifs entre ces coordonnées crée une trajectoire. Tous les deux points verts, un vecteur, correspondant au vecteur bleu sur la Figure 4-21, est généré. Le vecteur bleu est perpendiculaire au vecteur qui est la somme des deux vecteurs vert



Figure 4-19 Représentation de chemins d’empreintes

consécutifs partant du point vert. Il donne la position de l’empreinte représentée par le point jaune sur la Figure 4-20, la Figure 4-21 et la Figure 4-22. Le vecteur rouge donne la direction de l’empreinte suivant l’orientation de la trajectoire. Il est dans la direction de la somme des deux vecteurs verts.

Le vecteur somme de deux vecteurs verts, non montré sur les figures, joindrait chaque point jaune au premier point rouge se trouvant dans la direction du vecteur rouge. Le vecteur beige représente l’orientation supplémentaire pouvant être ajoutée dans le calcul de la rotation de l’empreinte pour accentuer l’effet d’empreintes de pas tournées vers l’extérieur. Cette

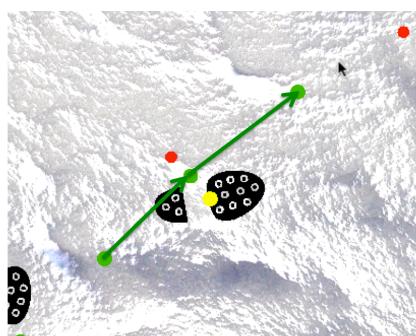


Figure 4-20 Vecteurs de trajectoire

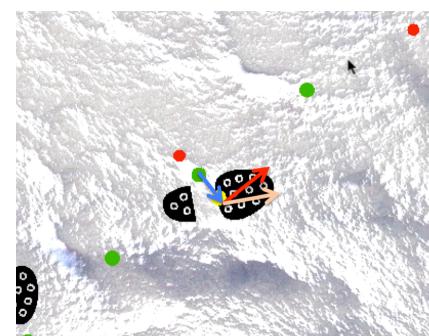


Figure 4-21 Vecteurs de position, direction et orientation

orientation prend compte du sens du pied : gauche ou droite. L'utilisateur peut spécifier l'orientation supplémentaire dans l'interface ainsi qu'un espace plus grand entre les empreintes le long de la trajectoire. Les vecteurs rouge et beige sont de longueur unitaire. Leur longueur n'importe donc pas dans les calculs.

La Figure 4-22 montre une succession d'empreintes avec chacun des vecteurs et points générés dans le calcul de leur orientation et position.



Figure 4-22 Série de pas et de vecteurs

4.3.7 Techniques d'infographie mises de l'avant

Les techniques d'infographie mises de l'avant dans l'application ont servi à améliorer la qualité du rendu et améliorer sa performance.

4.3.7.1 Le *normal mapping* pour un éclairage plus réaliste

Le *nomal mapping* est une technique d'éclairage qui rajoute du relief aux textures. Cette technique repose sur le calcul de la couleur de chaque fragment lu par un nuanceur en fonction de la carte de normales fournie avec la texture. Il faut tout d'abord calculer l'espace tangent de chacun des polygones de la scène et l'envoyer au nuanceur avec les coordonnées de texture des polygones. Chaque texel de la texture de normale fournit les composantes X, Y et Z de la normale en espace tangent. Ces texels agissent en tant que facteur de variation de la normale. Dans le nuanceur et pour chaque fragment, le calcul d'éclairage est effectué en espace tangent à l'aide de la normale fournie par le texel de la *normal map*. En rajoutant la couleur de la texture du fragment, un résultat plus réaliste est obtenu. La Figure 4-23 montre le sol de l'environnement avec et sans l'utilisation du *normal mapping*.

4.3.7.2 Utilisation de la mémoire graphique

Toutes les informations de la scène, à savoir les sommets, les coordonnées de textures et l'espace tangent sont stockés dans des VBO. Les VBO sont des éléments où sont stockées les données listées précédemment, mais encore, ce sont des éléments qui sont copiés dans la mémoire de la carte graphique. L'application indique alors à la carte graphique d'utiliser les données quelle à en mémoire pour rafraîchir la scène et ce, sans avoir besoin d'accéder à la

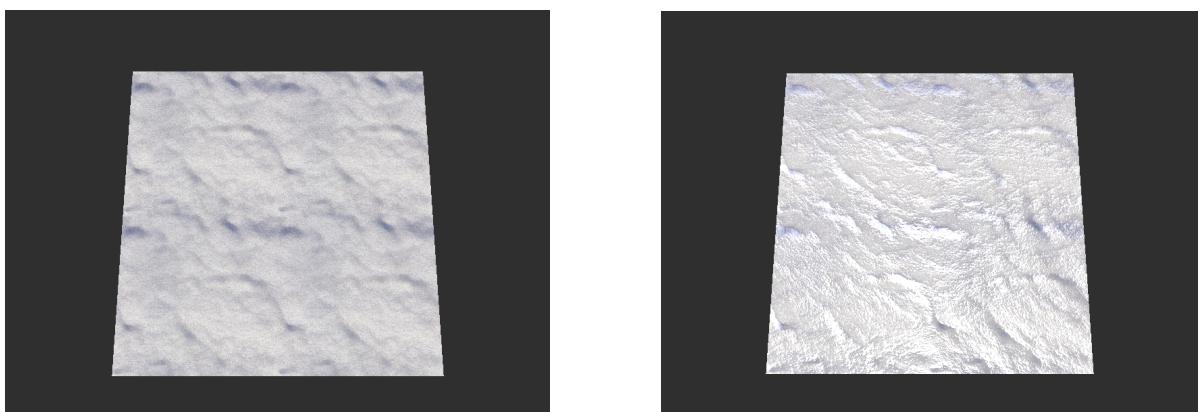


Figure 4-23 Scène sans et avec *normal mapping*

mémoire CPU où les données sont habituellement stockées. L'accès à la mémoire CPU suit un cycle qui peut ralentir le rafraîchissement de la scène lorsque le nombre de primitives est élevé.

4.3.7.3 Utilisation des textures comme sources d'informations

L'utilisation des textures pour stocker les informations des empreintes permet l'utilisation des nuanceurs pour pouvoir afficher des empreintes au fragment près. Contrairement au premier prototype, la couleur des fragments de la scène est directement modifiée si les fragments sont à proximité d'une empreinte. Les tests de proximité et les modifications de couleurs sont effectués pour chaque fragment de chaque polygone de la scène. Le dessin d'empreintes atteint par conséquent une grande précision.

4.3.8 Point négatifs et positifs

4.3.8.1 Moins de formes d'usures que le premier prototype

Avec le premier prototype, l'application pouvait reproduire n'importe quel type de dégradation dans l'environnement. En effet elle ne faisait qu'apposer un polygone dans la scène et ce polygone texture pouvait porter n'importe quelle texture. Dans le deuxième prototype, l'application se concentre uniquement sur la reproduction d'empreinte dans la scène. L'étendue de dégradation est plus limitée mais le résultat est plus travaillé.

4.3.8.2 Plus grand nombre d'empreintes

Grâce au mécanisme de partition de l'environnement, l'application peut traiter un grand nombre d'empreintes. La limite de hauteur d'une texture étant de 8 192 texels, il est possible d'apposer 8 191 empreintes par partition (les informations sur les coordonnées de la partition et le nombre d'empreintes dans la partition occupent le premier texel). Cependant l'application peut aller jusqu'à 4 000 empreintes sans ralentissement sérieux. Ces mesures

ont été effectuées avec un processeur Intel Core i5 à 2.3GHz et une carte graphique Intel HD Graphics 3000 de 384 Mo.

CHAPITRE 5

DISCUSSION

5.1 Difficultés surmontées

Les difficultés qui se sont présentées ont eu plusieurs origines. Certaines apparaissaient au cours du développement de l'application. La librairie libQGLViewer est encore peu documentée sur le web et trouver des réponses aux problèmes qui surviennent requiert de la patience. D'autres difficultés se situaient dans la définition du projet en lui-même telles que les exigences informelles et minces. Dans un document traitant du projet « environnements susceptibles », il est question de dégrader l'environnement par des marques d'usures et autres techniques. Le projet de maîtrise a donc commencé dans cette optique, mais ce ne fut qu'après la première réunion que les exigences concernant le type de dégradation changea et l'application devait avoir pour nouveau but de laisser des empreintes de pas. De plus il n'y avait pas d'indication concernant les modes d'interaction à implémenter, ni de certitude sur l'utilisation d'OpenSceneGraph. SPIN se basant sur OpenSceneGraph, développer le projet de maîtrise avec OpenSceneGraph aurait par conséquent garantit la compatibilité du projet de maîtrise avec d'autres sous-projets d'environnements susceptibles. Seulement le choix d'utiliser la plateforme SPIN pour « environnements susceptibles » n'était pas définitif à ce stade. Cela laissa plus le cours à l'exploration d'autres outils de développement tels que libQGLViewer ou Qt dans l'attente d'une confirmation.

Par ailleurs ça n'est qu'à la dernière rencontre que plusieurs exigences ont été spécifiées : garder l'information de la courbe où est passé l'utilisateur, ajuster les traces selon la vitesse de mouvement de l'utilisateur (marche vs course), effacer les traces graduellement, et l'impossibilité d'utiliser plusieurs textures par surface dans SPIN qui remet en cause l'utilisation de plusieurs textures dans les nuanceurs (texture d'informations d'empreinte, de sol neigeux et de différentes empreintes entre autres).

5.2 Valeur ajoutée de l'application

L'application a pour valeur ajoutée d'utiliser un nuanceur pour lire des informations d'empreintes au travers de textures et proposer un affichage efficace de celles-ci. L'astuce d'utiliser les textures pour cet effet plutôt que de simplement récupérer des informations de couleur n'est pas intuitive mais permet de communiquer avec le nuanceur aisément et de lui envoyer des informations très complètes sur chaque empreinte. Il est de plus possible de pouvoir placer beaucoup d'empreintes simultanément dans l'environnement sans en affecter la performance.

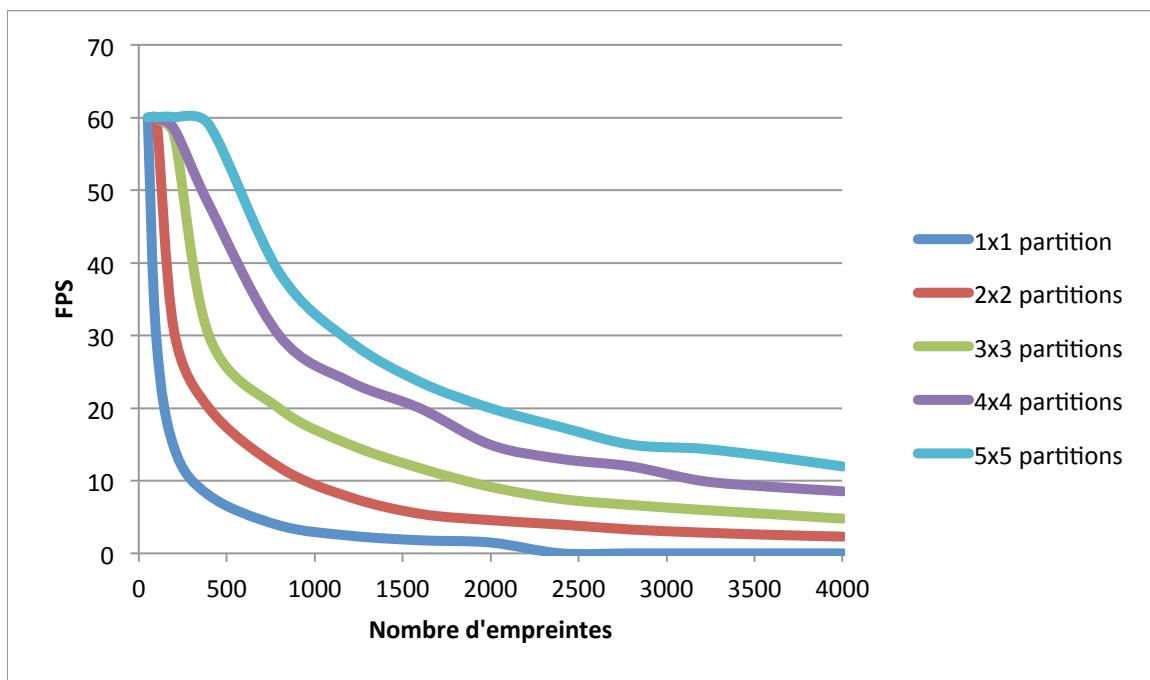


Figure 5-1 Évolution du nombre d'images par secondes par le nombre de partitions

5.2.1 Travail sur le stockage d'information par les textures

Le travail sur le stockage et l'envoi d'information par texture tire un grand profit de la partition de l'environnement et de la mise au point de « listes d'empreintes » par partition. En partitionnant l'environnement, l'incidence directe a été l'augmentation du nombre d'images par secondes en fonction du nombre d'empreintes simultanées dans l'environnement. La

Figure 5-1 démontre qu'au bout de 2 000 empreintes avec 5x5 partitions, l'application atteint un FPS équivalent à celui de 300 empreintes avec une partition 1x1. Plus le nombre de partitions dans l'environnement est grand, plus le nombre d'images par seconde est élevé et l'application fonctionne de façon plus fluide. En effet, les calculs effectués dans le nuanceur sont diminués car chaque fragment d'une partition donnée est traité contre moins d'empreintes. Ce phénomène est appuyé par les données présentées dans la Figure 5-2. Dans cette figure, l'évolution des courbes montre que plus l'environnement compte de partitions, plus le nombre d'empreintes par partition est faible. Le partitionnement de l'environnement diminue donc le nombre d'empreinte par partition. Il diminue les temps de calculs du nuanceur et augmente le nombre d'images par secondes.

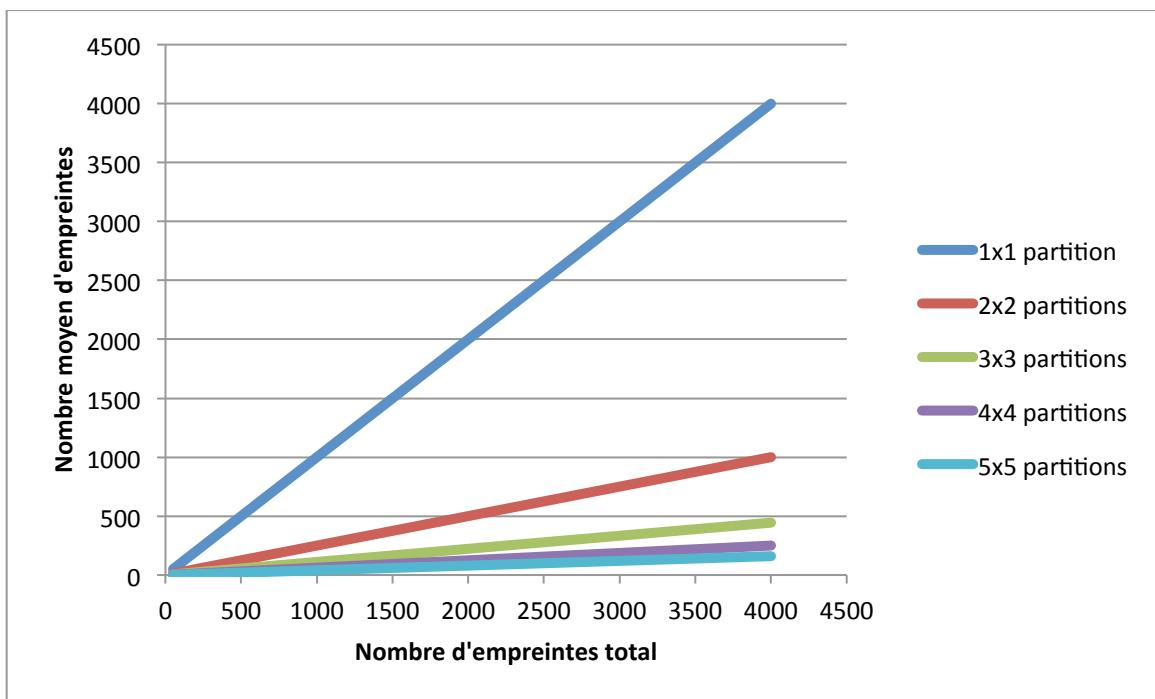


Figure 5-2 Évolution du nombre moyen d'empreinte par partition par le nombre de partitions

La Figure 5-3 démontre aussi le rôle clé que le nuanceur a dans les temps de calcul. Plus la taille d'une empreinte est petite, et plus le nombre d'images par secondes augmente. Ce phénomène s'explique par le fait que diminuer la taille d'une empreinte diminue le nombre de fragments qui sont transformés pour former une empreinte. De ce fait moins de calculs sont effectués dans le nuanceur et le nombre d'images par secondes est plus élevé.

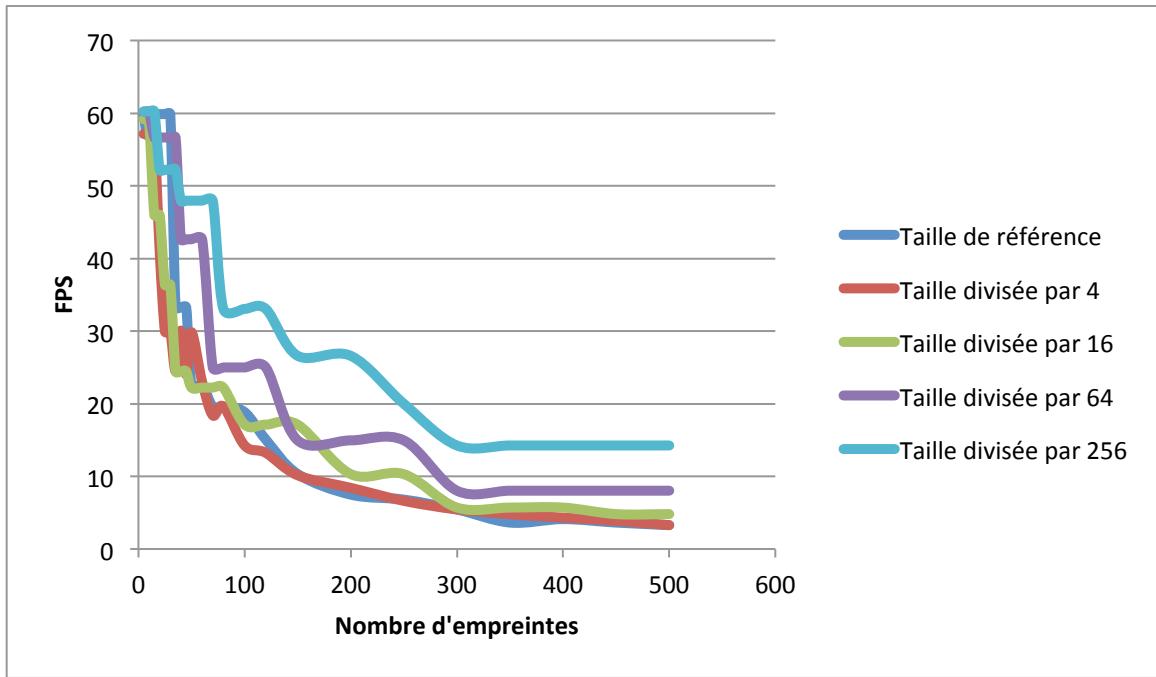


Figure 5-3 Évolution du nombre d'images par secondes en fonction du rayon des empreintes

5.2.2 Précision des empreintes

La précision des empreintes est atteinte grâce aux tests effectués dans le nuanceur pour chaque fragment. Cette précision dépend directement de la qualité des images d'empreintes fournies à l'application. La précision ne peut être améliorée dans le code mais uniquement avec les images d'empreintes. Elle est cependant suffisamment élevée au stade actuel.

La Figure 5-4 montre que pour apercevoir les pixels de l'empreinte de pas, il faut zoomer de très près. Une telle proximité ne sera sûrement pas atteinte si jamais un utilisateur observe son environnement. Comme l'image (b) de la Figure 5-4 l'illustre, zoomer très proche d'une empreinte est le seul moyen d'apercevoir ses pixels.

En considérant une approche où les empreintes de pas seraient copiées sur une texture des dimensions du sol plutôt que dans une texture d'informations, la texture aurait des dimensions dépendantes directement de celles du sol. Dans un cas réel, un utilisateur aurait besoin d'un sol de dimensions de 20 m^2 pour pouvoir se déplacer de façon intéressante. En se déplaçant, il laisserait une empreinte de pas tout les mètres environ et il pourrait en laisser

moyennement 3 de suite dans un mètre ce qui équivaudrait à 9 empreintes dans un m^2 . Un sol couvrant 20 m^2 pourrait donc contenir 180 empreintes au total. Une texture d'empreinte contient 512 pixels en hauteur par 512 pixels en largeur, ce qui équivaut à 262 144 pixels. Dans le contexte du prototype, l'approche d'une texture globale ayant une capacité de 180 empreintes juxtaposables aurait une résolution de 47 185 920 pixels au total.

Toujours dans l'optique d'une texture globale, le format de cette texture serait d'un octet pour chaque composant R, G, B et A (Format RGBA_8888), soit quatre octets au total pour un pixel. Cette texture occuperait donc 180 mégaoctets.

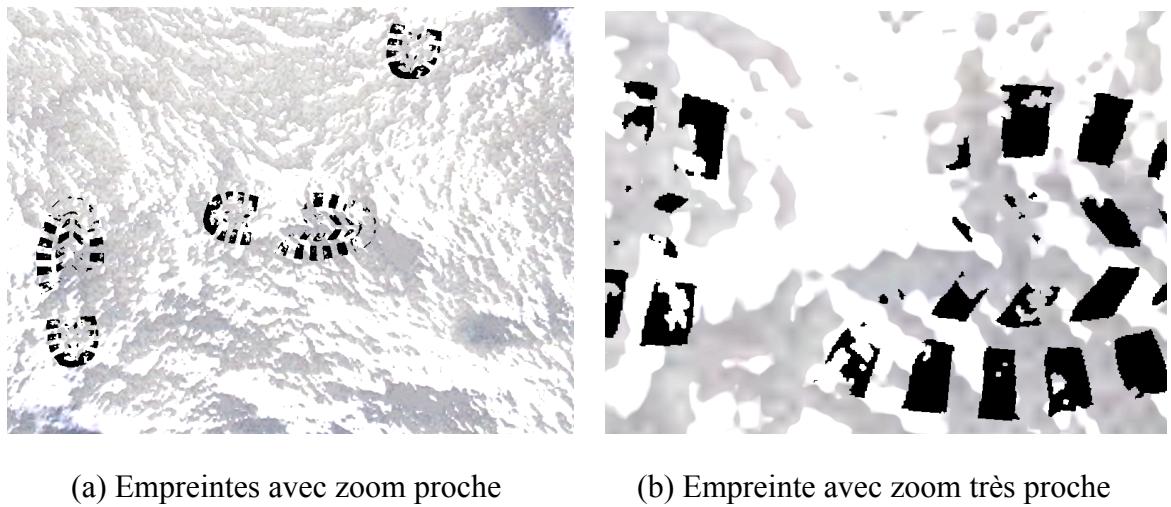


Figure 5-4 Zoom sur des empreintes de pas

Dans l'optique de la texture d'informations du prototype, une empreinte n'occupe pas 262 144 pixels mais un pixel. Le format de ce pixel est en nombre à point flottant. Les informations d'une empreinte sont réparties de ce fait sur 16 octets en mémoire (4 octets pour chaque composant). Pour 180 empreintes, cette texture occupe 2 880 octets (2,8125 kilooctets). La différence en consommation en mémoire est flagrante et la solution d'une texture d'informations pour stocker les empreintes s'impose. La limite pour une texture en terme de nombre de pixel est de 8 192 pixels en hauteur et 8 192 pixels en largeur. Dans l'approche d'une texture globale, sachant qu'une empreinte occupe 512x512 pixels et que la limite pour une texture est de 8 192x8 192 pixels, la texture ne pourrait contenir que 256 empreintes juxtaposées. Cela constituerait une deuxième limite pour cette approche et demanderait de rajouter des textures globales supplémentaires dans les cas où le nombre

d'empreinte est supérieur à 256 empreintes. Les nombreux processus fonctionnant en parallèle lors de l'exécution de l'application tels que le transfert sur la mémoire GPU des données de sommets, les calculs du *normal mapping* ou le déplacement obligent également la texture contenant les informations d'empreintes à consommer très peu de mémoire.

Cette analyse a été conduite dans le cas où un utilisateur se déplaçait dans l'environnement. Si trois utilisateurs interagissaient dans cet environnement, il faudrait tripler la surface de déplacement, donc multiplier la taille de la texture globale par 3 et multiplier également sa consommation en mémoire.

5.2.3 Plusieurs méthodes d'interaction

Plusieurs méthodes d'interaction sont proposées : le mode par clic qui permet d'étudier une empreinte en particulier et le mode par chemin qui permet de tracer un trajet d'empreintes. Le deuxième mode est plus intuitif et met plus en valeur l'application mais le premier reste essentiel à toute situation de débogage de l'application. Le mode de déplacement par caméra est un mode d'interaction implémenté actuellement mais qui ne laisse pas d'empreintes. Ajuster le mode de déplacement par caméra pour qu'il dépose des empreintes serait facile. En déplaçant la caméra dans l'environnement, l'application pourrait récupérer ses différentes positions et créer une trajectoire. En s'inspirant du mode par chemins dessinés à la souris, la trajectoire de la caméra serait alors utilisée pour calculer des positions d'empreintes de pas au lieu de la trajectoire dessinée par les mouvements de la souris.

5.2.4 *Normal mapping*

Le *normal mapping* apporte plus de réalisme et de détails à la scène. Cette technique d'éclairage améliore la performance du prototype dans le sens où jouer sur l'éclairage de la scène est moins coûteux en mémoire que rajouter des polygones pour accroître les détails. Cependant le relief est donné au sol plutôt qu'aux empreintes. Donner du relief aux empreintes constitue un travail plus complexe. Dans le *normal mapping*, les informations sur le relief sont déjà connues et fixées dans la carte de normale. Dans le cas des empreintes, ce

relief n'est pas fixe, peut varier par rapport à certains paramètres et vient se supplanter au *normal mapping* déjà en cours. Ces paramètres pourraient être la profondeur de l'empreinte de pas, la vitesse du déplacement ayant causé cette empreinte de pas ou l'apparence causée par la superposition de nombreuses empreintes à un même endroit

5.3 **Travail plus accentué sur les fonctionnalités que sur le réalisme graphique**

Habituellement lorsque des empreintes de pas sont laissées dans la neige, un réel relief apparaît, tel que la neige tassées au fond ou la profondeur de l'empreinte dans la neige. Passé dans le monde virtuel, chaque détail devient un aspect qui peut prendre beaucoup de temps à implémenter ou qui ne rentre pas forcément dans les objectifs premiers d'une démarche exploratoire. La démarche ici s'est plus orientée vers la mise au point d'une application qui permettait de laisser des empreintes de pas sur un sol dans le contexte d'une application artistique n'ayant pas besoin d'être complètement réaliste. Se concentrer sur les fonctionnalités était important puisque les requis n'était pas clairs et passer trop de temps à peaufiner des fonctionnalités alors qu'elles risquent de changer aurait été risqué et inutile. Elles devaient être passées en revue rapidement sans trop de détails et devaient jeter des idées globales dans l'approche du prototype.

5.4 **Discontinuités légères dans le déplacement de la caméra**

Le déplacement de la caméra tire profit d'interpolations fluides entre positions. Cependant il vient toujours un léger ralentissement entre plusieurs interpolations consécutives et lors d'un long déplacement ces ralentissements réguliers peuvent se faire sentir. Cela n'a pas d'incidence sur le fonctionnement de l'application à l'état présent mais pourrait poser un défi si l'application se devait de proposer un mouvement fluide de la caméra. Ce déplacement reste très différent du déplacement du premier prototype. Dans le premier prototype, le déplacement avec le zoom de la souris et avec les flèches du clavier n'était pas intuitif et était inadapté. Les possibilités offertes par le déplacement dans le deuxième prototype permettent

un mouvement fluide vers l'avant, l'arrière et latéralement. De plus il rend possible l'orientation vers le haut ou vers le bas, déplacement permettant d'observer des traces de pas sous des angles supplémentaires.

5.5 **Lecture de la texture d'informations d'empreintes de pas**

Le test effectué dans le nuanceur pour afficher les pixels d'une empreinte est fait grâce à un seuillage des valeurs R, G, B de la texture de l'image d'empreinte de pas. Les différentes empreintes sont de couleur noire sur fond blanc ce qui permet d'établir un seuillage sur les texels des empreintes vérifiant que leurs valeurs R, G, et B sont proches du noir. Lorsqu'un fragment se trouve dans le rayon d'une empreinte et que son texel correspondant dans la texture d'image d'empreinte de pas est proche du noir, sa valeur RGB prend la valeur du texel. Cette méthode présente une faille car elle ne fonctionne que pour des images d'empreintes de pas noires sur fond blanc. Si jamais la couleur de l'empreinte est différente du noir, le seuillage ne fonctionne plus. Une solution serait d'avoir l'image d'empreinte de pas sur fond transparent. Ainsi lorsqu'elle serait chargée dans l'application, cette dernière stockerait dans le canal alpha un pour un texel d'empreinte et zéro pour un texel du fond transparent. Dans le nuanceur le test de seuillage se réduirait à étudier ces deux valeurs du canal alpha des textures d'images d'empreintes. L'empreinte de pas pourrait prendre n'importe quelle couleur. Finalement le test ainsi simplifié pourrait améliorer la performance du rendu.

5.6 **Économie de mémoire dans la texture d'informations**

Pour une économie de mémoire, les emplacements vides de la texture d'information d'empreintes de pas pourraient être supprimés et chaque empreinte ajoutée sans sauter des emplacements vides. Pour ce faire le nuanceur pourrait utiliser le nombre d'empreintes de chaque partition stocké dans les premiers éléments du tableau pour savoir où l'index qui parcourt les empreintes d'une partition peut commencer et s'arrêter. Cette méthode

engendrerait un algorithme plus complexe dans le nuanceur mais la texture d'information d'empreintes de pas occuperait moins de mémoire. La nécessité de fixer sa hauteur maximale en fonction du plus grand nombre d'empreinte pour une partition serait retirée pour une hauteur adaptée à un tableau complètement rempli.

5.7 Aspect modulable des listes d'empreintes par partitions de la scène

Il existe une limite à la méthode de partitionnement de l'environnement : certaines empreintes peuvent chevaucher plusieurs partitions en même temps. Cette situation est courante, surtout dans le mode d'interaction par chemin et un des moyens trouvés pour le surmonter est de répartir ce type d'empreintes dans toutes les partitions qu'elles pourraient affecter. L'algorithme pour ce faire est de rajouter l'empreinte dans les partitions contenant son centre ou un sommet de sa boîte englobante. Cette méthode ne fonctionne cependant pas pour des empreintes dont la taille dépasserait la dimension d'une partition de l'environnement.

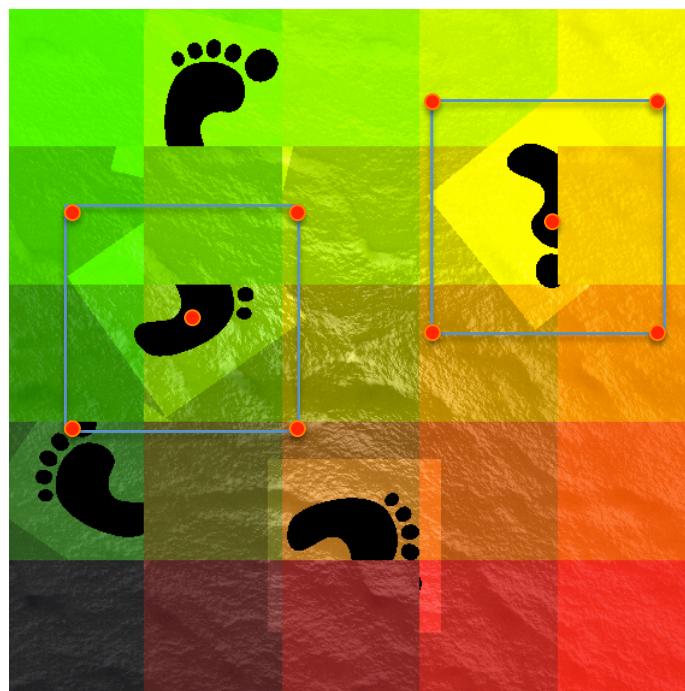


Figure 5-5 Empreintes dont la taille est supérieure à celle d'une partition

Dans ce cas particulier, la méthode ajoutant les informations d'une empreinte aux partitions suivant leurs boîtes englobante peut manquer certaines partitions adjacentes. Dans la Figure 5-5, deux empreintes sont mises en évidence. La boîte englobante de ces deux empreintes est en bleu, tandis que les coins et le centre sont en rouge. Les partitions que les empreintes traversent mais qui ne contiennent ni de coins de leur boîtes englobantes ni le centre de l'empreinte n'ont pas été prises en compte lors de l'ajout de ces empreintes dans la texture d'information. Elles ne détectent donc pas ces empreintes et les empreintes ne s'y affichent pas.

L'approche par partition présente donc plusieurs limites. Si la taille des empreintes devient trop grosse, la taille des partitions devrait être changée ou l'approche pour calculer les partitions affectées par une empreinte devrait être revue. De plus, à mesure que la taille des empreintes se rapproche de celle des partitions, l'information d'une même empreinte se duplique dans plusieurs partitions.

5.8 Flux de données entre l'interface et l'environnement

À l'aide des signaux de Qt, mettre à jour les informations de l'environnement grâce aux modifications causées par les évènements utilisateur du visualisateur et de l'interface graphique était aisé. Cependant, la pluralité des paramètres que l'utilisateur peut modifier et des évènements à interceppter rendent cette communication complexe. Un patron MVC aurait pu être implémenté avec l'environnement prenant la place du modèle de l'application. Le visualisateur libQGLViewer couplé à l'interface représenterait le visualisateur du patron. Un contrôleur aurait ensuite centralisé et géré toute communication entre le modèle et le visualisateur. La nécessité d'un tel patron n'est apparue que vers la fin du projet de maîtrise. En effet, l'aspect exploratoire rendait la pertinence de l'implémentation d'un tel patron incertaine. En effet ce type de patron devient très utile lorsque la communication entre différentes classes d'une interface est dense. Au début du projet de maîtrise, les communications entre l'interface, le visualisateur et l'environnement n'étaient pas assez complexes pour établir ce modèle. Avec l'apparition de différents modes de déplacement et du partitionnement de l'environnement, cette nécessité est apparue. L'implémentation de

nouvelles fonctionnalités accentueraient de ce fait le besoin d'un patron MVC pour organiser la communication entre les différentes classes du prototype.

5.9 **Changement du cahier des charges**

Le changement du cahier des charges fut inattendu. Survenu à la première rencontre, ce changement a transformé l'objectif de dégrader l'environnement en un objectif d'y laisser des traces. L'approche de développement envisagée par l'équipe du Métalab était beaucoup plus « exploratoire » que ce qui était attendu par l'équipe de l'ÉTS.

5.9.1 **Mise de coté de certains aspects du projet**

Les dégradations liées à l'usure de l'environnement par égratignure, rouille ou impacts ont été mises de coté. Des travaux de recherches proposaient déjà des solutions et les implémenter dans le cadre du projet de maîtrise avec la notion de présence aurait enrichi les méthodes d'interaction. Ils auraient cependant grandement complexifié l'interaction et nécessité beaucoup plus de puissance de calcul. De plus ils auraient nécessité beaucoup plus de temps d'implémentation, ce qui ne cadrait pas avec le travail de raffinement et de précision des requis.

5.9.2 **Simplification d'autres aspects**

Se focaliser sur les dégradations par empreintes de pas a réduit le nombre de dégradations. Cette décision fut au final plutôt bénéfique car elle a permis à l'application de se focaliser sur un type de dégradation précis, puis de se focaliser sur la technique d'utilisation de texture d'information dans le nuanceur qui elle pourrait être utilisée pour plusieurs types de dégradations.

5.10 Relation avec l'équipe du projet « environnements susceptibles »

L'équipe de projet « environnements susceptibles » avait autant un coté de collaborateur que de client.

5.10.1 Côté collaborateur

Le Métalab de la SAT utilise Redmine (Lesyuk, 2013) pour gérer ses projet et le logiciel de gestion de version git (Loeliger, 2009). Le serveur du Métalab héberge tous les projets sur lesquels le Métalab travaille actuellement ou a travaillé. Le support informatique de la SAT a préparé des comptes pour les membres de l'équipe de l'ÉTS. Ce système de versionnage permet de partager les fichiers de l'application avec la SAT et permet de faire persister ces données.

Bien que la conversion sous OpenSceneGraph de l'application est restée une hypothèse, il était sûr et certain qu'OpenGL devait être la librairie de développement graphique de base. L'utilisation de libQGLViewer était permise car cette librairie de plus haut niveau encapsule OpenGL. Pour ce qui est d'OpenScenegraph, la même raison permettrait d'adapter l'application sans repartir à zéro.

L'application d'empreintes de pas développée dans le cadre du projet « environnements susceptibles » n'est pas le seul sous-projet en cours de développement. Implémenté sous l'environnement SPIN, le projet « walkinsnow » utilise une carte d'élévation et un nuanceur pour simuler un terrain enneigé où la caméra laisse un sillage dans le sol dans son déplacement. Il fait partie d'un sous-projet sur lequel travaille un membre de l'équipe du Métalab. Idéalement le projet « walkinsnow » pourrait apporter une dimension plus réaliste aux empreintes de pas du projet de maîtrise. Dans un autre sens le projet de maîtrise pourrait apporter une dimension plus réaliste dans l'interactivité, dans l'application de véritables empreintes de pas dans l'environnement et dans la gestion de leurs informations. Coupler les nuanceurs de chaque projet pourrait permettre de modifier la carte de hauteur utilisée pour générer le relief des empreintes de pas. La caméra, en se déplaçant dans l'environnement,

laisserait un sillage parsemé d'empreintes de pas marquées par leur relief. Le projet « walkinsnow » aurait toutefois une résolution moindre et garderait seulement l'aspect général du passage des utilisateurs.

5.10.2 Côté client

Le projet « environnements susceptibles » est un projet de recherche création qui sera implanté à la SAT même. L'équipe du projet « environnements susceptibles » intervient donc en tant que collaborateur sur le projet de maîtrise, de contributeurs au projet et aussi en tant que client car ils recevront ultimement le projet de maîtrise et ses livrables. Luc Courchesne et Zack Settel sont des chercheurs mais aussi des artistes. Ils ont comme particularité de prôner l'exploration dans leur démarche. Ce trait reflète l'exploration dont le projet a fait preuve dans la recherche de ses exigences et dans son implémentation. Les exigences ont donc pris forme au fur et à mesure que le projet avançait.

Les deux chercheurs étaient les sources de demandes en ce qui concerne les exigences de l'application. Ils ont notamment demandé l'utilisation d'empreintes de pas, puis incorporé des exigences sous forme de distance entre les pas dépendant de la vitesse de déplacement, de traces disparaissant dans le temps et de courbes conservant l'information de déplacement de l'utilisateur de l'application.

CONCLUSION

Le projet environnements susceptibles est un projet recherche-création géré par le Métalab de la Société des arts technologiques. Il a pour but de créer un environnement immersif et collaboratif sujet à des dégradations par les visiteurs et ce dans le but de traiter des notions telles que la présence et l'existence de l'autre. Ce rapport traite du prototype, faisant état du projet de maîtrise, comptant parmi les précurseurs du projet « environnements susceptibles ». Ce prototype est une application graphique visant à laisser des empreintes de pas sur le sol. Implémentée en OpenGL, Qt et libQGLViewer, l'application permet d'afficher des empreintes de pas en temps réel sur n'importe quel élément de la scène avec grande précision. Elle tire profit des cartes graphiques programmables en exploitant la puissance des nuanceurs. Le stockage d'informations des empreintes y est optimisé par la partition de l'environnement en sous espaces dont chacun possède sa propre liste d'empreintes. Peu de mémoire et peu de transmission seraient nécessaires dans le cas où les informations d'empreintes étaient échangées entre des sites distribués.

Ce projet de maîtrise a pu clarifier la pensée artistique en fournissant des solutions concrètes aux problématiques de notions de présence, d'interaction entre utilisateurs et de modélisation d'un type de dégradation. De plus l'évolution entre le projet de maîtrise et le prototype « walkinsnow » a été fructueuse, d'autant plus que ces deux prototypes sont complémentaires. La collaboration a donc été un succès autant pour la SAT que l'ÉTS.

Dans un avenir immédiat, cette application serait convertie dans une des librairies de développement graphique telle qu'OpenSceneGraph et SPIN puis intégrée au projet « walkinsnow ». À long terme, d'autres aspects du projet « environnement susceptibles » pourront être incorporés tels que d'autres types de dégradation aussi bien visuelle qu'auditives, le déploiement dans les dômes immersifs dont la SAT est munie et le support de plusieurs utilisateurs dans un même environnement.

RECOMMANDATIONS

Plusieurs fonctionnalités directement liées à l’application pourraient voir le jour. Une gestion des mouvements de coté ou vers l’arrière pourrait venir enrichir le mode d’interaction par chemin. En effet celui-ci ne simule que la marche avant. La prise en compte d’un changement de direction à l’opposé pourrait garder les empreintes dans le même sens alors que la trajectoire va à l’opposé. Pour gérer les mouvements de coté, il serait envisageable d’introduire un nouvel événement : proposer les mouvements latéraux typiques des jeux avec caméra à la première personne (FPS). L’application conserverait alors la dernière direction enregistrée et les empreintes de pas générées garderaient cette direction. La distance entre les pas pourrait varier en fonction de la vitesse de déplacement.

La forme des pas pourrait être modifiée en fonction de la vitesse. Les empreintes de pas des coureurs de vitesses sont différentes que celles de personnes marchant à une allure normale. L’étude des variations de distances entre les empreintes de pas serait requise pour explorer l’implémentation de ce phénomène dans l’application.

Des phénomènes naturels qui effacent les traces dans la réalité tels que la neige, la pluie et le vent pourraient être des facteurs influant sur les empreintes. Les empreintes de pas pourraient, par exemple, avoir une durée de vie. Plus le temps passerait, plus elles s’effaceraient et ce en fonction du moment auquel elles auront été posées dans le sol.

Les trajectoires des chemins peuvent apparaître trop linéaires. Plusieurs techniques existent pour estimer des courbes à partir d’ensemble de points consécutifs : les courbes splines par exemple sont beaucoup utilisées dans le milieu du développement d’application graphique pour retracer de telles trajectoire avec plus d’acuité.

La gestion du « multi-personne » sera omniprésente dans le projet « environnements susceptibles », et l’approfondissement de cette piste pourrait être envisageable par différents moyens toujours au niveau du prototype. L’application ne permet de gérer qu’une souris pour tracer un chemin. La correspondance entre une souris et un chemin ouvre la voie à la proposition de deux souris pour deux chemins. De même, le déplacement par chemin avec la souris pourrait être substitué par un déplacement avec le clavier et différentes parties du clavier pourraient permettre à deux personnes de poser des empreintes dans la scène. Un

périphérique multi-tactile pourrait remplacer l'usage de la souris et du clavier. Plutôt que de devoir analyser des évènements reçus par deux souris, l'application pourrait gérer plusieurs évènements tactiles. Deux personnes pourraient, à titre d'exemple, dessiner des chemins différents en utilisant un tel périphérique. Les défis de répartitions des évènements tactiles associés à chaque utilisateur viennent donc se rajouter. Cette notion de multi-personne est directement reliée au traitement parallèle et à la gestion de différents processus en simultané, piste à explorer.

La mise au point d'une texture sans emplacements vides serait un premier pas vers une texture d'information dont la taille est optimisée. Actuellement une texture peut stocker jusqu'à 8 912 empreintes par partitions. Sachant que sous OpenGL une texture peut mesurer 8 192 texels en hauteur et largeur, avec cette approche le nombre maximal d'empreintes serait 67 108 864. Si jamais ce nombre est atteint, l'utilisation de texture supplémentaires pourrait doubler voire tripler ce nombre. L'application pourrait attendre que la taille maximale d'une texture soit atteinte pour en utiliser une autre. Cependant une texture dont le nombre d'empreinte maximum a été atteint pèserait 1 Go et elle affecterait grandement le fonctionnement de l'application. L'application pourrait donc ajouter une texture supplémentaire à un palier moindre que celui maximal autorisé dans OpenGL.

Les empreintes de pas sont le premier type de dégradation suggéré et en cours de réalisation. La scène est actuellement constituée d'un sol plat. Mettre des empreintes sur un sol à l'aide d'une carte d'élévation (2 1/2D) serait possible avec un réalisme correct. Dans la continuité avec le projet « environnements susceptibles », la fusion avec le projet de cartes d'élévation rendrait possible ce réalisme. Cette fusion permettrait de palier au manque de « vraie neige » dans l'environnement. Le prototype d'empreinte de pas devra être transposé dans la librairie OpenScenegraph, voire le framework SPIN. Les difficultés principales dans cette transposition sont que sous SPIN, le principe du *multi-texturing* n'est pas encore supporté, principe qui permet l'utilisation de plusieurs textures dans le nuanceur. Cette fusion requerrait donc un changement à l'architecture de SPIN.

Laisser des empreintes dans de l'herbe donne des effets différents que dans la neige, effets intéressants à reproduire. Coupler la neige et la végétation dans une même scène pourrait être une première étape : une fois posée dans la neige, l'empreinte de pas révélerait de la

végétation. Le changement de saisons pourrait alors modifier le climat et l'apparence de la scène. Tantôt de la neige tomberait, couvrant le sol. Au bout d'un certain laps de temps, de la chaleur la ferait fondre et laisserait réapparaître de l'herbe.

Rajouter des objets à dégrader à l'aide d'empreintes serait un autre moyen de dégrader l'environnement. Cependant, des changements importants seraient nécessaires afin que les empreintes puissent être mises sur n'importe quel objet 3D. A titre d'exemple, la dégradation par changement de forme et topologie viserait à manipuler des objets 3D en ajoutant et retranchant des faces, arrêtes, sommets selon la déformation de l'objet. Ainsi les objets concernés changerait de forme. OpenSceneGraph possède des fonctionnalités qui supportent la manipulation d'objets au niveau de leurs faces. Le paradigme d'interaction serait cependant à revoir et il faudrait définir comment l'objet doit être modifié. James F. Obrien propose dans son corps de travail plusieurs techniques dans cette direction (Chentanez et al., 2006; Narain et al., 2013; Wang et al., 2010; Wicke et al., 2010). L'une d'elle (Narain et al., 2013) simule la déformation plastique appliquée à des couches de métal et de vêtements froissés. Cette technique pourrait être utilisé sur du métal ou encore des objets composés de tissu. Une autre technique (Wang et al., 2010) s'intéresse à simuler des marques de pression ou de tension sur des objets rigides. Elle agit sur des paramètres de tension invariants des coordonnées de la surface et simule un comportement isotropique. La dégradation de la forme d'un objet pourrait, avec cette technique, jouer sur des caractéristiques élastiques d'un objet. Une technique (Wicke et al., 2010) s'intéresse à des comportements élasto-plastiques. Elle simule des maillages à leurs états de repos et leurs états déformés et y stocke des décalages de la surface pour représenter les parties non déformables de la déformation plastique.

Le craquèlement de peinture ou l'apparition de rouille sur des matériaux d'objets pourraient être des dégradations envisageables sur des objets 3D. Le défi qu'elles poseraient serait de proposer des méthodes intuitives d'interaction avec les utilisateurs. Le craquèlement de la peinture pourrait être la conséquence d'un impact avec la surface qu'elle recouvre ou le séchage d'un liquide jeté par l'utilisateur. La rouille pourrait suivre le même principe de jet de liquide, apparaissant donc au contact de l'eau. Ces dégradations ont un avantage similaire à celui des empreintes de pas : les localiser et fournir des informations sur elles est possible.

D'autres types de dégradations pourraient être implémentés. Les empreintes de pas représentent un déplacement sur terre ferme. Dans l'hypothèse d'un déplacement dans les airs, un type de dégradation possible serait la fumée. La difficulté dans cette entreprise tiendrait plus au déplacement assimilable à celui du vol d'un avion qu'à la fumée aisément réalisable à l'aide d'un système de particule.

Le son sera également un facteur de « présence ». Il pourrait être la voix d'un utilisateur ou un son émis par l'interaction d'un utilisateur avec l'environnement. Dans cette dernière optique, il pourrait être directement lié à des dégradations visuelles tels que des sons de pas, sons de chocs, etc. M. Settel a déjà réalisé des travaux en captation, en diffusion et en spatialisation sonore dans les espaces physiques, augmentés et virtuels.

Les environnements seront projetés dans des Panoscope 360, dispositifs immersifs conçus par Luc Courchesne. Ces dômes utilisent une technologie de projection de vidéo simplifiant la production de présentation de contenus panoramiques et immersifs. Ils représentent une alternative à d'autres systèmes tels que le système CAVE coûteux ou les lunettes vidéo ne permettant pas à plusieurs personnes de partager le même espace. Le projet « environnements susceptibles » devra permettre une interaction collaborative, interaction simulant la notion de présence et celle de l'autre. Avant d'interagir entre eux, les utilisateurs devront interagir avec l'environnement et pouvoir s'y déplacer avec aisance. Le système proposé par le projet « environnements susceptibles » devra proposer une méthode de déplacement intuitive concordant avec l'utilisation du Panoscope 360. Les utilisateurs interagiront entre eux et plusieurs idées ont déjà été proposées à ce sujet. Dans un premier temps, ils ne se voient pas, mais voient leurs dégradations. Les utilisateurs pourraient par exemple ne pas voir leurs avatars dans l'environnement mais ne voir que leurs dégradations. Ce procédé mettrait davantage en valeur l'utilité des empreintes et des dégradations : la présence de l'autre ne serait plus symbolisée par la vision de l'autre mais la perception des traces qu'il laisserait derrière lui. Dans un deuxième temps, les utilisateurs laisseraient des dégradations derrière eux perceptibles par la vue et l'audition. Étendre les dégradations à l'audition conduira à une interaction plus riche, complexe et ludique.

L'interaction collaborative nécessitera certaines exigences non-fonctionnelles et plusieurs problématiques peuvent déjà être identifiées. La première question qui pourrait venir à l'esprit se rapporte à la nature des données qui seront communiquées entre les utilisateurs. La carte de l'environnement pourrait être diffusée à chaque utilisateur en continu, en particulier la carte d'élévation représentant le relief de la scène. Chaque dégradation de l'environnement affecterait donc une carte « centrale ». Une limite de cette démarche serait qu'elle ne permet pas de dégrader des objets ou procéder à des dégradations n'affectant pas la carte d'élévation. Une alternative serait de seulement communiquer des listes de dégradations entre utilisateurs. Chacun en aurait une, et elle serait envoyée à tout autre utilisateur de l'environnement. En restant dans l'optique de dégradations sous forme d'empreintes, une dernière option viendrait découpler les empreintes de leur trajet, gardant ce dernier sous la forme d'un historique de trajet. Cet historique serait ultérieurement communiqué entre les utilisateurs. La seconde question qui se pose est la méthode de partage des données envisagées. La communication point à point lie les systèmes d'utilisateurs dans des relations point à point ou point à multipoint. À l'opposé, la communication client-serveur utilise un réseau entre plusieurs logiciels. Les clients envoient leurs requêtes au serveur, le serveur attend les requêtes et y répond. Cette communication se doit de suivre un protocole de communication établissant des règles de communication sur le réseau. En suivant ce modèle, chaque utilisateur représenterait un client communiquant avec un serveur hébergé par la Société des arts technologiques. Le pair au pair est un modèle proche du modèle client serveur pouvant également être considéré : dans un modèle client-serveur, il existe un ou plusieurs serveurs pour un ou plusieurs client. Dans le pair au pair chaque client est également un serveur (Berson, 1996).

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Berson, A. 1996. *CLIENT/SERVER ARCHITECTURE. 2nd edition, édition en anglais.* McGraw-Hill.
- Bosch, Carles, Pierre-Yves Laffont, Holly Rushmeier, Julie Dorsey et George Drettakis. 2011. « Image-guided weathering: A new approach applied to flow phenomena ». *ACM Trans. Graph.*, vol. 30, n° 3, p. 1-13.
- Chen, Yanyun, Lin Xia, Tien-Tsin Wong, Xin Tong, Hujun Bao, Baining Guo et Heung-Yeung Shum. 2005. « Visual simulation of weathering by gamma-ton tracing ». In *ACM SIGGRAPH 2005 Papers*. (Los Angeles, California), p. 1127-1133. ACM.
- Chentanez, Nuttapong, Tolga G. Goktekin, Bryan E. Feldman et James F. O'Brien. 2006. « Simultaneous coupling of fluids and deformable bodies ». In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (Vienna, Austria), p. 83-89. 1218075: Eurographics Association.
- Dorsey, Julie, Hans Kohling Pedersen et Pat Hanrahan. 1996. « Flow and changes in appearance ». In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. p. 411-420. ACM.
- Endo, Yuki, Yoshihiro Kanamori, Jun Mitani et Yukio Fukui. 2010. « An interactive design system for water flow stains on outdoor images ». In *Proceedings of the 10th international conference on Smart graphics*. (Banff, Canada), p. 160-171. Springer-Verlag.
- Gu, Jinwei, Ravi Ramamoorthi, Peter Belhumeur et Shree Nayar. 2007. « Dirty glass: rendering contamination on transparent surfaces ». In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. (Grenoble, France), p. 159-170. Eurographics Association.
- Gu, Jinwei, Chien-I Tu, Ravi Ramamoorthi, Peter Belhumeur, Wojciech Matusik et Shree Nayar. 2006. « Time-varying surface appearance: acquisition, modeling and rendering ». *ACM Trans. Graph.*, vol. 25, n° 3, p. 762-771.
- Lesyuk, A. 2013. *Mastering Redmine*. Packt Publishing, Limited.

Loeliger, J. 2009. *Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development*. O'Reilly Media, Incorporated.

Lu, Jianye, Athinodoros S. Georghiades, Andreas Glaser, Hongzhi Wu, Li-Yi Wei, Baining Guo, Julie Dorsey et Holly Rushmeier. 2007. « Context-aware textures ». *ACM Trans. Graph.*, vol. 26, n° 1, p. 3.

Mérillou, Nicolas, Stéphane Mérillou, Eric Galin et Djamchid Ghazanfarpour. 2012. « Simulating how salt decay ages buildings ». *IEEE Comput. Graph. Appl.*, vol. 32, n° 2, p. 44-54.

Merillou, Stephane, Jean-Michel Dischler et Djamchid Ghazanfarpour. 2001a. « Corrosion: simulating and rendering ». In *Proceedings of Graphics interface 2001*. (Ottawa, Ontario, Canada), p. 167-174. Canadian Information Processing Society.

Merillou, Stephane, Jean-Michel Dischler et Djamchid Ghazanfarpour. 2001b. « Surface scratches: measuring, modeling and rendering ». *The Visual Computer*, vol. 17, n° 1, p. 30-45.

Narain, Rahul, Tobias Pfaff et James F. O'Brien. 2013. « Folding and crumpling adaptive sheets ». *ACM Transactions on Graphics*, vol. 32, n° 4, p. xxx:1-8.

Pacanowski, Romain, Oliver Salazar-Celis, Christophe Schlick, Xavier Granier, Poulin Pierre et Cuyt Annie. « Rational BRDF ». *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, n° 11, p. 1824-1835.

Paquette, Eric, Olivier Clément et Jocelyn Benoit. 2007. « Efficient editing of aged object textures ». In *Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. (Grahamstown, South Africa), p. 151-158. ACM.

Paquette, Eric, Pierre Poulin et George Drettakis. 2001. « Surface aging by impacts ». In *Proceedings of Graphics interface 2001*. (Ottawa, Ontario, Canada), p. 175-182. Canadian Information Processing Society.

Paquette, Eric, Pierre Poulin et George Drettakis. 2002. « The Simulation of paint cracking and peeling ». In *Proceedings of Graphics Interface*. inria , version 1. p. 10. Canadian Human-Computer Communications Society.

Rost, Randi J., Bill Licea-Kane, Dan Ginsburg, John M. Kessenich, Barthold Lichtenbelt, Hugh Malan et Mike Weiblen. 2009. *OpenGL Shading Language*. Addison-Wesley Professional, 792 p.

Sun, Bo, Kalyan Sunkavalli, Ravi Ramamoorthi, Peter N. Belhumeur et Shree K. Nayar. 2007. « Time-Varying BRDFs ». *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, n° 3, p. 595-609.

Thelin, J. 2007. *Foundations of Qt Development*. Apress.

Wang, Huamin, James O'Brien et Ravi Ramamoorthi. 2010. « Multi-resolution isotropic strain limiting ». In *ACM SIGGRAPH Asia 2010 papers*. (Seoul, South Korea), p. 1-10. 1866182: ACM.

Wang, R., et X. Qian. 2010. *OpenSceneGraph 3.0: Beginner's Guide*. Packt Publishing, Limited.

Wicke, Martin, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk et James F. O'Brien. 2010. « Dynamic local remeshing for elastoplastic simulation ». In *ACM SIGGRAPH 2010 papers*. (Los Angeles, California), p. 1-11. 1778786: ACM.