

Communication et collaboration dans les processus de création multi-utilisateurs

Xavier Gouchet

DÉPARTEMENT ARTS ET TECHNOLOGIE DE L'IMAGE
MASTER 2 RECHERCHE - UNIVERSITÉ PARIS VIII

Table des matières

Introduction	2
1 Présentation du projet	3
1.1 Résumé du projet SCP Caméra	3
1.2 Historique du projet SCP Caméra	4
1.3 Le projet “Faites du clip”	5
1.4 Description du projet	5
<i>Illustrations</i>	6
2 État de l’art	8
2.1 Introduction	8
2.2 Évolution de l’infographie	8
2.3 Évolution des système de communication (Internet)	9
2.4 Prémices d’une rencontre : Le Net Art	10
2.5 Contact et utopie : le VRML	11
2.6 Développement des jeux en lignes	12
2.7 Communautés et virtuel : un buzz dans la société actuelles	13
2.8 Les technologies web 3D	14
2.9 MovieStorm	15
<i>Illustrations</i>	16
3 Développement d’un système de modules communicants	20
3.1 La théorie	20
3.2 Gestion de projets et utilisateurs	20
3.3 Protocole de communication	21
3.4 Lecture et traitement des données	23
3.5 Gestion de la base de données	24
3.6 Utilisation des modules	24
<i>Illustrations</i>	25
A Rapport de Stage	29
A.1 Présentation de la société Virtual-IT	29
A.2 Compte-rendu de mes activités	30
A.3 Bilan du stage	33
<i>Illustrations</i>	35
Sources	37
Remerciements	39

Introduction

Lors de la seconde année de Master en Arts et Technologie de l'Image, nous avons décidé de reprendre le projet de l'année précédente, "SCP Camera", et d'en faire une application multi-utilisateurs en réseau. L'application aurait pour but de créer un contenu s'apparentant à un film, où chaque utilisateur sur un projet aurait à gérer une partie du contenu.

Une des problématiques dans les studios de production est l'augmentation du rendement, et la baisse du temps nécessaire à la réalisation d'un projet. Or, la meilleure organisation possible pour obtenir un rendement efficace est de faire travailler chaque individu en parallèle. Cependant ce genre de chose est parfois difficile à réaliser.

Ceci est vrai notamment dans les travaux de pré-production, et l'étape appelée prévisualisation. Cette étape consiste à créer une scène minimaliste en 3D, dans laquelle le réalisateur va prévoir son tournage (succession des plans, angles de vue), avec pour but de diminuer le temps de tournage sur le plateau (où l'équipe technique est plus nombreuse et donc plus chère).

Malheureusement, à ce jour, lorsque la scène 3D nécessite une modification, le temps d'édition interrompt la séance de réflexion avec le réalisateur, et augmente les coûts de la pré-production. Lors d'une rencontre avec la société de prévisualisation Halon Entertainment, cette problématique a été le principal sujet de discussion.

Le défi de notre projet était de réaliser des modules capables de fonctionner sur le réseau, soit seul, soit avec plusieurs utilisateurs en parallèle.

Après une présentation rapide du projet de Master 2, nous verrons un historique qui permettra de comprendre où en sont les technologies de la communication et comment elles peuvent être utilisées pour créer des applications collaboratives en général, et ce projet en particulier. Enfin, nous verrons une description du projet réalisé au cours de l'année 2006-2007.

En annexe se trouvent le rapport du stage réalisé de mai à septembre 2007 dans le cadre de mon Master Recherche en Arts et Technologie de l'Image, ainsi qu'une liste des sources bibliographiques et webographiques ayant servies à écrire ce mémoire.

Chapitre 1

Presentation du projet

1.1 Résumé du projet SCP Caméra

Le projet SCP Caméra¹ est une application de réalité virtuelle, réalisée durant l'année scolaire 2006- 2007 par des étudiants de Master 1 en Arts et Technologie de l'Image : Nicolas Serikoff, Rémi Quittard et Xavier Gouchet.

Le but originel du projet était de fournir un outil permettant de former des cameramen au manie- ment d'une caméra. Très vite l'outil est devenu un moyen simple de prototyper des mouvements de caméra, réutilisables par la suite dans un environnement de production en 3D (Maya, XSI, ...).

L'application se compose de plusieurs parties :

- le périphérique SCP Caméra ;
- l'application temps réel ;
- le plug-in d'import maya.

Le périphérique : une interaction intuitive

Le périphérique a été créé à partir d'un boîtier de caméra VHS vide, dont les boutons ont été liés à une interface USB plug & play. Un écran LCD a été ajouté au boîtier afin d'offrir un retour visuel à l'utilisateur. Le principal but et intérêt de ce périphérique est d'offrir une interface intuitive aux utilisateurs de l'application (cf fig. 1.1).

En plus des boutons permettant d'interagir avec l'application, le boîtier peut être équipé d'un capteur de mouvements permettant d'augmenter l'immersion de l'utilisateur dans la scène. Il est possible d'utiliser aussi bien un capteur optique que magnétique, car l'application est adaptable à n'importe quel capteur.

L'application : un outil 3 en 1

L'application est un programme 'temps réel' permettant d'importer des scènes 3D afin d'y créer des mouvements de caméra, avec ou sans le périphérique prévu à cet effet. Créée sous Virtools, elle offre quelques options pour ajouter des informations sur la scène en cours, et d'ajuster les paramètres de la scène (échelle, vitesse de l'animation, contenu).

L'application s'articule en trois modes, c'est-à-dire les trois étapes d'utilisation du projet. Chacun, possédant son propre code couleur et ses propres outils, contribue à aiguiller l'utilisateur lorsqu'il crée une animation avec SCP Caméra.

¹SCP sont les initiales de Shoot, Cut & Play, les trois modes utilisés dans l'application et les trois étapes d'utilisation du projet.

Le premier mode est le mode Shoot. Dans ce mode, l'utilisateur peut créer ses mouvements de caméra. Pour cela il peut soit utiliser son clavier et sa souris, et se déplacer comme dans un jeu vidéo, soit utiliser le périphérique couplé à un capteur de mouvements pour obtenir une sensation plus immersive.

La plupart des contrôles classiques que l'on retrouve sur une caméra ont été implémentés afin d'avoir un maximum d'options lors de la capture de mouvements. Ainsi, l'application enregistre non seulement les mouvements (position, orientation) de la caméra, mais aussi les données du zoom et de la profondeur de champ.

Chaque enregistrement est sauvegardé dans un fichier texte sur le disque dur, visible sous forme de clip dans la timeline, située en bas de l'écran (cf fig. 1.2).

Le second mode est le mode Cut. Ce mode contient des outils pour éditer les clips situés dans la timeline. Celle-ci correspond à la timeline classique d'un logiciel de montage. Chaque clip peut être déplacé, coupé ou supprimé. Le but est de pouvoir créer un montage à partir des mouvements enregistrés, afin de profiter de la fluidité et de la puissance du moteur 3D temps réel.

Le dernier mode est le mode Play, qui donne la possibilité de rejouer les mouvements enregistrés, en tenant compte du montage réalisé dans le mode Cut.

Enfin, il est possible de sauvegarder son montage et de l'exporter vers un fichier texte pour l'utiliser dans un logiciel de production classique. Le programme est léger et facile à appréhender, ce qui en fait un outil très utile pour prototyper ses mouvements de caméra.

Le plug-in : le lien vers le rendu

Le plug-in permet d'importer les données d'un montage directement dans un logiciel de 3D (Maya, XSI, ...), afin de réutiliser les mouvements de caméra créés. Pour chaque clip du montage, une caméra est créée, à laquelle on ajoute ensuite des clefs d'animation sur la position, l'orientation, le zoom et la profondeur de champ.

Les mouvements ainsi créés peuvent servir alors de base pour un film d'animation, ou simplement pour contrôler des caméras de prévisualisation.

1.2 Historique du projet SCP Caméra

Commencé en septembre 2005, le projet est présenté en public pour la première fois au festival Laval Virtual 2006, dans la compétition étudiante. A cette occasion, de nombreux professionnels repèrent le potentiel du projet, et redonnent la motivation et les idées nécessaires pour mener le projet à bien.

C'est durant ce même festival que le projet SCP Caméra rencontre la société Immersion, qui présentait alors un prototype de capteur de mouvements infrarouge : le Cyclope. Un partenariat entre les deux groupes s'installe, et le Cyclope est ajouté rapidement et facilement au projet.

En octobre 2006, le projet est présenté dans une compétition étudiante, le 3D³, organisé par Virtools, où il remporte le prix de l'usage innovant.

Ensuite, entre le mois de mars et le mois de juillet 2007, le projet est repris, amélioré, et corrigé pour deux événements. Tout d'abord le projet participe de nouveau au festival Laval Virtual, présenté sur le stand d'Immersion.

Puis, en août 2007, au SIGGRAPH² 2007 (à San Diego, CA), le projet est présenté dans la catégorie Emerging Technologies. À cette occasion, le groupe prend contact avec différents studios, parmi lesquels le studio ILM³, et Halon Entertainment (un studio de prévisualisation).

²Special Interest Group on GRAPHics

³Industrial Light and Magic, le studio d'effets spéciaux créé par Georges Lucas

1.3 Le projet “Faites du clip”

Après le concours du 3D³, le projet semblait très prometteur. Au fil des présentations, nous avons reçu de nombreuses idées pour des évolutions possibles, aussi nous semblait-il important de continuer le projet.

La première question à régler était de choisir quelle direction prendre parmi les différentes pistes qui s’offraient à nous. Nous avons le choix entre plusieurs options, dont les suivantes :

- réalité augmentée : ajouter une capture vidéo sur le périphérique, et mélanger des images réelles avec des images en 3D ;
- application de pré-production : corriger les derniers bugs de l’application (hardware / software) et en faire un produit de pré-production, compatible avec les systèmes utilisés par les studios ;
- jeu vidéo : créer une application ludique en se basant sur le périphérique, comme par exemple un jeu de reporter virtuel.

Finalement nous avons choisi de créer une application en ligne permettant à un utilisateur de créer son propre univers et d’y tourner un film de courte durée, en y ajoutant une bande sonore de son choix. Ainsi est né le projet “Faites du Clip”.

A cette occasion, et comme nous avons besoin d’aide pour ce nouveau projet, plus ambitieux que le précédent, nous avons demandé à Célia Demere et Chloé Lang-Willar de nous rejoindre sur ce projet.

1.4 Description du projet

Le projet “Faites du Clip” contient plusieurs modules permettant de créer un clip musical.

Le module de création des décors

Ce module, à l’instar de beaucoup d’outils 3D, permet de créer une scène 3D à partir d’une bibliothèque d’objets. La bibliothèque est découpée en différentes catégories selon le type et le style des objets.

L’utilisateur peut alors choisir ses objets et les placer dans la scène. L’objet est manipulable à volonté : position, orientation, taille et couleur. À tout moment l’utilisateur peut sauvegarder sa scène, ou en charger une autre afin de la modifier.

Mise en place des éclairages

Ce module permet d’ajouter des lumières aux décors créés avec le précédent module. Les deux modules sont séparés car les deux utilisent des paramètres et des interactions différentes.

Le module de création des personnages

Ce module est destiné à créer un personnage unique. En modifiant les paramètres du personnage de base (la longueur du nez, la corpulence, ...), l’utilisateur peut obtenir presque n’importe quel personnage. Il peut en outre modifier les vêtements et accessoires du personnage.

Ajouter des animations

Une fois le personnage créé, il est possible de lui ajouter une suite d’animation afin de lui donner vie. Chaque animation est jouée après la précédente.

Créer les mouvements de caméra

Ce module est basé très largement sur le projet SCP Caméra. Il permet d’importer décors et personnages animés, et de créer des mouvements de caméra. Ce module contient également le module de montage auquel ont été ajoutées des options de transitions et des effets visuels.

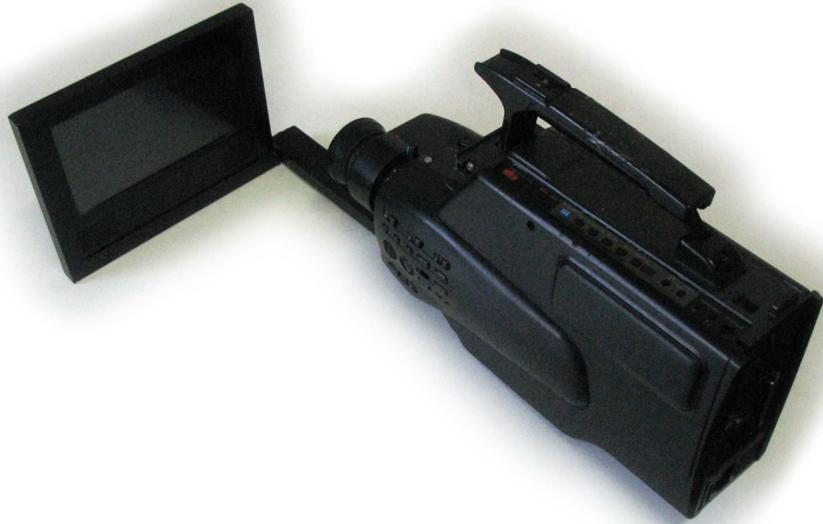


FIG. 1.1 – Le périphérique SCP Caméra

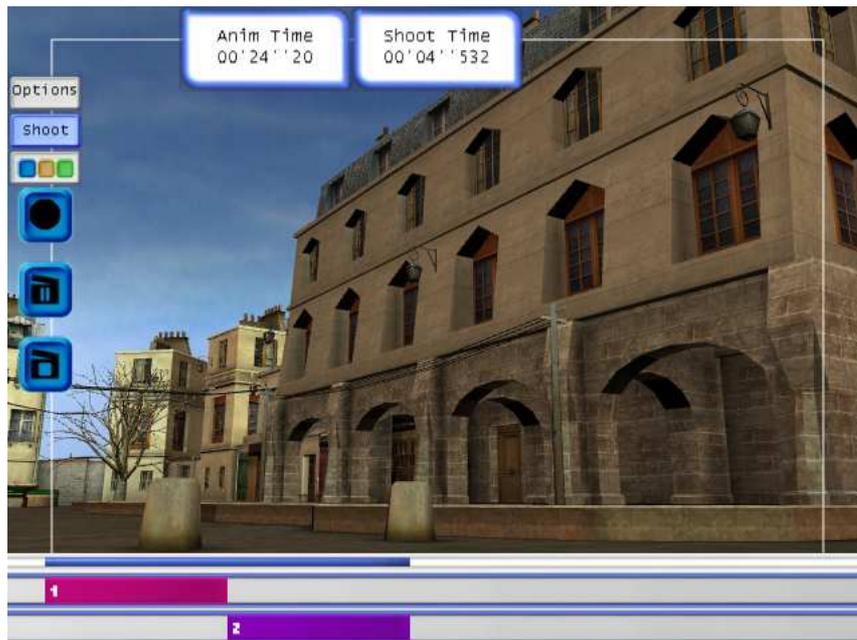


FIG. 1.2 – L'application temps réel



FIG. 1.3 – Le plug-in d'import Maya

Chapitre 2

État de l'art

2.1 Introduction

En 1946 fut construit le premier calculateur capable d'effectuer 300 opérations à la seconde, l'ENIAC ¹. Les sciences informatiques se développèrent depuis de plus en plus rapidement, et parmi elles, deux courants qui ont d'abord évolué parallèlement avant de se rejoindre vers la fin des années 1980.

Ces deux courants sont la communication d'un côté, qui a pris une place importante dans notre société actuelle, et l'imagerie numérique et ses dérivées, qui tient une part non moins importante des sciences informatiques.

Nous allons, après avoir rappelé l'évolution de ces deux courants, comment ils interagissent aujourd'hui, et comment cette interaction a permis de créer le projet SCP Camera.

2.2 Évolution de l'infographie

L'une des premières problématiques suite à la naissance des super calculateurs fut l'affichage des données. Le canon à électron, déjà utilisée pour la télévision depuis la fin des années 1930 fut rapidement adoptée, et détournée en 1950 par Ben Laposky dans un but artistique : en créant des vagues sur un oscilloscope, et en les photographiant, il pu obtenir des images abstraites (cf fig. 2.1).

C'est en 1960 que William Fetter invente le mot "Computer Graphics", plus tard traduit en français par infographie, alors qu'il dessine les plans de cockpits à échelle humaine pour Boeing. Trois ans plus tard, General Motors utilise le premier système de Conception Assisté par Ordinateur, le DAC 1 développé par IBM. L'industrie de l'automobile entraîna de nombreux progrès en matière d'imagerie numérique, parmi lesquelles les fameuses courbes décrites par Pierre Bezier en 1970, alors qu'il travaillait pour Renault.

Les artistes eux aussi s'approprièrent l'outil informatique, suivant l'exemple de Ben Laposky. On peut citer, entre autres, Charles Csuri (dont les premières images remontent à 1963), ainsi que Michael Noll et sa série "Gaussian Quadratics" (cf fig. 2.2). En 1975, les travaux de Benoît Mandelbrot sur les fractales, et dont l'ensemble reste une source d'image quasi inépuisable, donnèrent naissance à une branche entière de l'infographie. Ken Musgrave notamment, utilisa beaucoup les fractales pour générer des paysages (cf fig. 2.3).

L'industrie du film a également participé au développement et à l'amélioration des outils et algorithmes utilisés dans la production et l'affichage d'images numériques. Dès la fin des années 1950, les génériques de film exploitent les possibilités offertes par l'ordinateur, comme Vertigo, d'Hitchcock en 1958 puis plus tard Superman en 1978.

¹Electronical and Numerical Integrator and Computer

Les premiers effets spéciaux utilisant des images générées par ordinateur restent des scènes cultes, comme la scène de l'Étoile noire de "Star Wars" (Lucasfilm, 1977), l'introduction du film "Le trou noir" (Disney, 1979), et les scènes de l'univers virtuel de Tron (Disney, 1982, cf fig. 2.4). Des techniques comme le morphing dans Willow (1987), ou les dinosaures de Jurassic Park (1993) rendirent les effets spéciaux de plus en plus populaires.

Les jeux vidéos participèrent également à l'essor de l'imagerie numérique, avec cette contrainte de calculer les images en temps réel. Depuis le jeu Spacewars en 1961, et le Pong d'Atari en 1972, le jeu devint une industrie, qui connut de nombreux succès. Notamment la série des Myst (cf fig. 2.5), depuis 1993, dont la qualité graphique était toujours en avance sur son temps.

2.3 Évolution des systèmes de communication (Internet)

Dans les années 60, des recherches ont été initiées pour trouver un mode de communication entre deux ordinateurs distants. Le but : des interactions sociales aux possibilités nombreuses, déjà entrevue à l'époque. Cependant il faudra attendre 1966 pour que l'agence de Défense Américaine (DARPA) lance un projet d'envergure : l'ARPANET, reliant des universités américaines ayant passé un accord avec le DARPA.

C'est en 1969 que le premier équipement dédié à ce réseau fut installé à l'université de Californie. Suivirent la Stanford Research Institute, et les universités de Santa Barbara et de l'Utah. C'est enfin en 1972 que les premiers protocoles furent au point et permirent notamment l'échange de courriers électroniques.

Fort de ce succès, Robert Khan (qui avait participé au développement d'ARPANET) travailla avec Vinton Cerf sur le protocole de communication TCP, encore utilisé de nos jours, et qui donnera naissance à l'Internet tel que nous le connaissons aujourd'hui.

Enfin en 1984, alors que les liaisons deviennent mondiales, Tim Berners-Lee, un chercheur du CERN crée un « ensemble de documents scientifiques rattachés les uns aux autres afin de faciliter les recherches », principe désormais utilisé dans l'Internet grand public.

Ce concept, appelé Hypertexte, avait été imaginé dès les années 1960 par Théodore Holm Nelson. Selon lui, cela désigne un réseau de documents liés les uns aux autres. Le principal intérêt de ce système est de ne pas être linéaire, contrairement à un texte classique.

Ted Nelson, avec ce concept, développe également le projet Xanadu, un système utopique basé sur l'hypertexte. Dès 1967, il écrit les 17 règles qui décrivent ce projet. Parmi ces règles, huit sont particulièrement intéressantes :

1. Chaque serveur Xanadu est unique et sécurisé.
2. Chaque serveur Xanadu peut être mis en service séparément ou en réseau.
3. Chaque utilisateur est unique et identifié.
4. Chaque utilisateur peut rechercher, récupérer, créer et stocker des documents.
5. Chaque document peut consister en un nombre quelconque de parts donc chaque élément peut être constitué de quelque genre que ce soit.
- ...
- 11 Chaque document peut avoir des règles d'accès sécurisés.
- 12 Chaque document peut rapidement être recherché, stocké et récupéré sans que l'utilisateur ne sache où il est physiquement situé.
- ...
- 17 Le protocole de communication client-serveur Xanadu est un standard librement publié. Le développement et l'intégration de tierces parties sont encouragés.

Ces huit règles sont les seules utilisées par le World Wide Web. Les autres règles, étant pour une sécurité maximale et contre le copyright, ne peuvent fonctionner sur l'Internet que nous connaissons, ou de nombreux contenus sont protégés et/ou payant.

Tim Berners-Lee lui-même le reconnut lorsqu'il participa à la création du World Wide Web :

"I just had to take the hypertext idea and connect it to the TCP and DNS ideas and – ta-da! – the World Wide Web."²

Aujourd'hui, le projet Xanadu a évolué pour tenter de répondre à une question simple :

"How can computer documents – shown interactively on screens, stored on disk, transmitted electronically – improve on paper?"

"Keep every quotation connected to its original source."³

Ce projet a pour but d'afficher un document lié aux sources de ses citations. En créant une page à partir de citations, cela permet au lecteur d'avoir accès facilement au contexte du sujet de l'article qu'il est en train de lire.

Ce concept se rapproche de celui du Cyberspace ; d'ailleurs, le projet Xanadu est en train de développer un navigateur en 3D sur ce principe. Sur un document, on voit une représentation graphique des sources des citations. (cf fig. 2.6).

2.4 Prémices d'une rencontre : Le Net Art

Le net art est un courant artistique né dans les années 1980. La principale caractéristique de ce courant est qu'il utilise internet comme outil de création et de diffusion.

L'idée est que une oeuvre Net Art ne peut vivre sans être reliée à un réseau. Si une fois déconnectée de tout réseau l'oeuvre continue d'exister, elle ne peut prétendre à l'appellation net art.

Steve Dietz, ancien conservateur du Walker Art Center de Minneapolis, donne cette définition du net art :

"Internet art projects are art projects for which the Net is both a sufficient and necessary condition of viewing/expressing/participating."⁴

Cette définition assez large englobe différents protocoles utilisés sur Internet : email, ftp, telnet, listserv, http, etc. Le web art prend sa source dans l'art électronique (qui s'est développé dans les années 1970 et qui donna naissance au festival Ars Electronica).

Entre 1994 et 1999, des artistes firent connaître le net art, comme Vuk Cosic (célèbre pour ses travaux sur l'ASCII art), Alexei Shulgin et son jeu "Form Art", Valéry Grancher et ses Webpaintings, ou Jodi, le site des artistes Joan Heemskerk et Dirk Paesmans, ou encore l'histoire interactive de Olia Lilliana, basée sur le concept des liens hypertexte.

Après 1999, le net art est tombé dans l'oubli, même si quelques artistes continuent de créer des oeuvres interactives, comme sur le site <http://b-l-u-e-s-c-r-e-e-n.net/>.

²J'ai juste eu à reprendre l'idée de l'hypertexte et la lier aux concepts de TCP et DNS et – ta-da! – le World Wide Web."

³Comment des documents électroniques – affichés interactivement sur des écrans, stockés sur des disques, transmis électroniquement – peuvent s'améliorer par rapport au papier? En gardant chaque citation liée à sa source originale

⁴Les projets Net Art sont des projets artistiques pour lesquels le Net est une condition nécessaire et suffisante pour le voir/créer/participer

2.5 Contact et utopie : le VRML

A l'origine, le VRML (Virtual Reality Markup Language) fut créé pour être le pendant en trois dimensions du format HTML. Une page HTML, comportant du texte et d'éventuelles images ne lui convenait pas pour créer son univers virtuel. Lors de sa conférence au World Wide Web Conférence en Mai 1994, Mark Pesce explique que pour avoir un bon langage décrivant un monde virtuel, il faut d'une part une description des objets qui occupent le monde, ainsi qu'une description de leur agencement dans l'univers virtuel, et enfin un lien URL vers une donnée associée à l'objet en question.

Dans cette même conférence, Mark Pesce explique sa vision du VRML. Avec lui, il compte créer un monde virtuel qu'il nomme CyberSpace, un espace multidimensionnel équivalent à l'Internet, permettant de se déplacer sur le réseau d'une manière plus intuitive.

“For the purposes of continuity in navigation, it is necessary to create a unified conceptualization of space spanning the Internet, a spatial equivalent of WWW. This has been called ‘Cyberspace’, in the sense that it has at least three dimensions, but exists only as a ‘consensual hallucination’ on the part of the hosts and users which participate within it. There is only one Cyberspace, just as there is only one WWW; to imply multiplicity is to defeat the objective of unity.

At its fundamental level, Cyberspace is a map that is maintained between a regular spatial topology and an irregular network topology. The continuity of Cyberspace implies nothing about the internet upon which it exists. Cyberspace is complete abstraction, divorced at every point from concrete representation. ”⁵

Le cyberspace de Mark Pesce est une carte spatiale, une topologie régulière à laquelle sont associées des données du réseau internet qui n'ont aucune coordonnées spatiale. Le système de placement des données se fait par un système métrique basé sur l'adresse IP de la page web, ou plus exactement du serveur qui contient la page web.

Le but était de créer un véritable monde virtuel dans lequel le déplacement se ferait « physiquement », et non plus virtuellement au moyen de liens uniquement textuel.

Le format de fichier 3D associé au VRML est devenu un standard pour l'échange de fichiers destinés à être visualisés en temps réel. De nombreux systèmes donnent la possibilité d'afficher directement dans une page Web une scène en 3D, et permettent ainsi de multiplier les contenus sans avoir les inconvénients d'une installation pour chaque scène.

Devenue une référence en matière d'affichage 3D en ligne, le VRML a été utilisé par certains artistes, comme par exemple Charles Csuri, et on vit fleurir des mondes virtuels tel que CyberTown.

En 2003, un nouveau format standard est apparu : le X3d. Ce format se base sur les technologies XML et VRML, afin de devenir le format pour la 3D temps réel sur Internet (Web3D). En utilisant une syntaxe XML, le X3D partitionne les composants du VRML afin d'offrir une structure plus solide.

En se basant sur l'idée de Cyberspace de Mark Pesce, Christian Babski a créé le ‘_knowscape’. Il s'agit d'un navigateur qui utilise le VRML pour représenter graphiquement, en 3D, les pages web que visite l'utilisateur.

⁵Dans le but d'avoir une continuité dans la navigation, il est nécessaire de créer une conceptualisation unifiée de l'espace englobant l'Internet, un équivalent spatial du World Wide Web. Cela a été appelé ‘Cyberspace’, dans le sens où il possède au moins trois dimensions, mais existe comme une ‘hallucination consensuelle’ pour les hôtes et utilisateurs qui y participe. Il y a un unique Cyberspace, comme il y a un unique World Wide Web; supposer une multiplicité détruit le but d'unité. A son niveau fondamental, Cyberspace est une carte qui est maintenue entre une topologie spatiale régulière et une topologie en réseau irrégulière. La continuité de Cyberspace n'implique par rapport au réseau sur lequel il existe. Cyberspace est complètement abstrait, dissocié en chaque point d'une représentation concrète.

2.6 Développement des jeux en lignes

Parmi les oeuvres Net Art, on vit apparaître des oeuvres qui faisaient penser à des jeux, comme par exemple "Form Art" de Alexei Shulgin. Ce dernier utilise les formulaires HTML comme moyen de déplacement dans une page, le but étant d'en trouver la sortie. On peut également penser à l'histoire interactive d'Olia Liliana, proche des livres de jeux de rôles "Le livre dont vous êtes le héros".

Ce type de jeu de rôle a été recrée dès 1978 avec les MUD (Multi User Dungeon), jeux de rôles multi-utilisateurs utilisant simplement du texte. La puissance des liens hypertexte et la légèreté du texte permettait à plusieurs joueurs de se retrouver dans un même univers avec un simple modem.

À la même époque, les jeux de plateaux classiques tel que Scrabble, les échecs ou le jeu de Go, firent leur apparition en ligne. Ces jeux, pour éviter les coûts importants de communication (payé à la minute), utilisaient un système de tour par tour, basé sur des échanges d'e-mail.

Le jeu Doom (cf fig.2.7), sorti en 1993, fut un véritable pionnier dans le monde du jeu vidéo, non seulement parce qu'il fut l'un des premiers à utiliser un moteur de rendu 3D en temps réel, mais surtout car il possédait un système de match multi-joueurs en réseau local.

Dans le courant des années 1990, des jeux de stratégie furent créés, en permettant de jouer non plus uniquement en réseau local mais également sur internet, parmi lesquels Warcraft II (1996), Age of Empire (1997) ou encore StarCraft (1998).

Enfin, avec l'augmentation du nombre de fournisseur d'accès à Internet, et l'amélioration des connections, des jeux massivement multi-joueurs (appelés MMO pour Massively Multiplayer Online games) virent le jour. Ces jeux sont composés de mondes virtuels grands et riches en détails, dans lequel des milliers de joueurs peuvent interagir (discuter, échanger des objets, combattre, ...). Le plus connu est sans conteste World of Warcraft (2005), dont la richesse n'est pas égalée.

En parallèle, les technologies Java et Flash permirent de (re)créer des jeux visible directement dans un navigateur internet. Très rapidement ces jeux obtinrent un grand succès car ils pouvaient être joués seuls ou à plusieurs sur internet. La plupart des jeux Flash ou Java restent des petits jeux d'arcade, qui ne nécessitent pas de sauvegarde de l'avancement dans le jeu.

Des plateformes de jeux se sont ensuite développés, basés sur les technologies Flash et/ou java, proposant aux joueurs de gagner des cadeaux ou de l'argent en jouant sur le site. Par exemple Prizee fait gagner des points qui peuvent ensuite être échangés contre des cadeaux, le tout étant rémunéré par la publicité sur le site.

Des jeux de gestions sont également apparus, proposant à l'utilisateur d'élever un animal (comme pour les Tamagoshis), le tout à partir d'une classique interface web en HTML. Ainsi des sites comme Neopets (1999) ou Funnypigs (2002, connu comme Kochonland en France) ouvrirent la voie. On retrouve aujourd'hui de nombreux jeux de ce type permettant d'élever/gérer non seulement des animaux, mais aussi des humains.

Plus récemment, le jeu Miniville propose de faire évoluer sa ville, non plus en gérant ses compte, ses routes, ses bâtiments directement (comme dans SimCity) mais simplement en visitant la page correspondante. Chaque personne qui visite la page d'une ville ajoute un habitant à cette ville (une visite par personne par jour). De même une visite sur une page similaire permet d'augmenter le développement industrielle, ou réduire la criminalité. On peut enfin grâce à une dernière page construire une bombe pour l'envoyer sur une ville concurrente.

2.7 Communautés et virtuel : un buzz dans la société actuelles

Le terme de communauté virtuelle prends sa source en 1985, avec la création du WELL⁶, la première communauté d'échange électronique. Il désigne tout regroupement de personnes par un moyen offert par les technologies réseau (usenet, forums, chats, e-mails, ...).

Les chats, e-mails et forums ont peu changé depuis leur origine, si ce n'est par rapport au design et la facilité d'utilisation. Chaque communauté regroupe des personnes avec un intérêt particulier en commun (un jeu, un sujet,...). Les forums en sont la meilleure illustration. Il est très facile de créer un forum, certains sites proposant des outils complets, et de fait, on voit fleurir des forums sur presque tous les sites webs.

Depuis le milieu des années 1990, l'accès à internet s'est démocratisé, et les communautés en tout genre ont fleuri. Le nombre d'utilisateur grandissant, de nouveaux types de communautés ont vu le jour.

L'un des phénomènes les plus marquant reste celui des blogs (ou Web Logs). Il s'agit à l'origine d'une sorte de journal intime que l'auteur rend publique. Chaque visiteur peut alors lui laisser des commentaires. Mais certains internautes ont créé des blog non plus pour parler de leur vie privée mais pour discuter d'actualité ou de sujets qui leur tenait à coeur.

On trouve aujourd'hui de nombreux blogs parlant de politique, de nouveautés technologiques, ou d'économie, mais aussi des blogs d'artistes amateurs qui cherchent à se faire connaître. Souvent, des blogs ayant un sujet en commun se regroupent, c'est-à-dire font des liens les un vers les autres, et créent ainsi une communauté.

Il existe d'autres communautés où les utilisateurs sont aussi créateurs de contenus. Ainsi des sites comme Elfwood (1996) ou FanFiction (1999) propose à des fans d'écrire des textes sur leurs livres/films préférés et de les publier en ligne. Ce genre de concept est devenu très populaire grâce à Youtube (2005), Dailymotion (2005), DeviantART (2000) et Flickr (2004), qui offre aux utilisateurs un espace pour diffuser leurs vidéos et/ou photos.

Enfin, il existe un dernier type de communauté que nous avons évoqué plus haut : les jeux Massivement Multi-joueur. Les jeux vidéos ont, depuis 1993 avec Doom, provoqué des regroupements de joueurs sur le net. Mais depuis l'arrivée de jeux comme Second Life (2003) ou World Of Warcraft, les joueurs sont plongés dans un univers riche, où ils peuvent incarner n'importe quel personnage.

Second Life est un jeu hors du commun, car en réalité il n'y a ni but, ni quête. il s'agit uniquement d'un espace d'échanges dans un univers en 3D. Le but de Second Life est d'être un espace aussi riche et divers que la vie réel, où les utilisateurs peuvent échanger, débattre ou s'instruire sur tous les sujets possibles.

Dés lors, on voit apparaître un phénomène de masses où des utilisateurs subissent une dépendance à ces mondes virtuels. Surnommés les "no-life", ces joueurs fuient la réalité et se réfugient dans l'univers virtuel où ils peuvent être qui ils veulent. Cette vague, surtout présente chez les adolescent, est le résultat d'un besoin d'échapper à la vie quotidienne et à son stress.

Un dernier phénomène, qui n'est pas directement communautaire, est le machinima. Des utilisateurs de jeux multi-joueurs, tels que World of Warcraft, Counter Strike ou Quake, enregistre des vidéos à partir du jeu, et transforment ces vidéos en courts métrages. Ce nouveau style de film est devenu très populaire et a conduit à la réalisation de MovieStorm (2007, cf 2.9 p. 15).

⁶Whole Earth 'Lectronic Link

2.8 Les technologies web 3D

Il existe deux façons de lier les technologies du web et la 3D. La première, et l'une des plus utilisées à l'heure actuelle, est de créer un programme autonome, utilisant le matériel de l'utilisateur, mais se reposant sur son propre système de communication.

Cette façon de faire est assez contraignante car pour chaque application, l'utilisateur doit installer un nouveau logiciel. Hors pour beaucoup d'utilisateur, cette étape est parfois difficile.

Une autre option est d'intégrer directement le moteur 3D dans le navigateur internet de l'utilisateur. C'est le principe utilisé par les plug-ins VRML. L'intérêt principal est qu'une seule installation est nécessaire pour pouvoir afficher tous les contenus.

Le succès des technologies Flash et Java le prouve : ces deux plug-ins sont incontournables, et permettent d'accéder à une quantité énorme de contenu.

Mais pour les moteurs 3D, il n'existe pas encore de référence. Le VRML, qui aurait pu le devenir, n'est plus utilisé que pour son format de fichier, et peu de site propose encore des contenus interactifs qui utilisent cette technologie.

Aujourd'hui il existe plusieurs environnements de développements, qui gèrent du contenu interactif en 3D, proposant des moteur 3D intégrable à une page Web. Parmi ceux là, trois logiciels ont acquis une expérience suffisante.

Créé en 1998, la société Virtools a acquis un savoir faire en matière de 3D temps réel. Leur logiciel permet de créer des comportements facilement. Chaque projet peut être exporté en tant que programme autonome, mais aussi comme une scène visible avec le lecteur, ou même comme application compatible avec des consoles de jeux.

Leur lecteur est assez simple à installer, et l'un des seuls qui soit compatible tant avec Internet Explorer que Firefox (qui sont utilisés respectivement par 76% et 22% des internautes mondiaux). De plus, il est doté de nombreux outils permettant de communiquer avec le navigateur.

Le logiciel Virtools, récemment racheté par Dassault Système, est reconnu comme une référence en matière de 3D temps réel, et malgré son coup élevé pour les petites entreprises, reste la solution la plus complète dans le domaine.

La société Act3D (1997) a développé en 2000 un logiciel similaire, Quest 3D, permettant de créer et manipuler du contenu 3D facilement. Le but est de renouveler la façon dont les projets de 3D temps réel sont créés, en simplifiant les manipulations au maximum.

Les projets peuvent également être exportés en tant que programme autonome, ou comme scène adaptées à leur lecteur. Cependant le lecteur n'est, à ce jour, compatible qu'avec Internet Explorer.

Enfin le logiciel Subdo, conçu en 2004, moins ambitieux que Virtools ou Quest 3D, est assez simple à utiliser pour des débutants, et propose des options comparables à celles des deux autres logiciels.

Son lecteur est, avec celui de Virtools, le seul compatible avec Internet Explorer et Firefox, et est utilisé entre autre par le site d'essayage virtuel du site de vente de vêtements en ligne La Redoute. Subdo offre également la possibilité d'exporter ses scènes au format Pdf3D. Ce format permet d'intégrer des élément en 3D dans un document pdf, et est disponible dans le logiciel Acrobat Reader depuis la version 7.

2.9 MovieStorm

Récemment (juin 2007) est apparu en beta-test⁷ le logiciel MovieStorm (<http://www.moviesstorm.co.uk>). Ce logiciel offre à l'utilisateur la possibilité de créer son propre film en 3D (cf fig.2.8).

Au départ, l'utilisateur choisit son décors, ses acteurs, puis entre dans l'interface de tournage. Un peu de la même façon que dans les Sims, il peut donner des actions à faire aux acteurs et déplacer ses caméras. Il est également possible d'éditer le décors en changeant la place des éléments, ou de modifier les vêtements que portent les acteurs.

L'un des grands avantages de ce logiciel est qu'il permet de faire du lipsync à partir d'enregistrement audio. Le logiciel offre également la possibilité d'exporter un film entier.

Le fait que de telles applications apparaissent prouvent le besoin qu'on les internautes de pouvoir devenir créateurs et non plus juste spectateurs.

⁷Le logiciel peut-être utilisé gratuitement. En contrepartie les utilisateurs doivent notifier tous les bugs qu'ils rencontrent. Cette étape permet de tester un logiciel dans des conditions d'utilisations réelles.

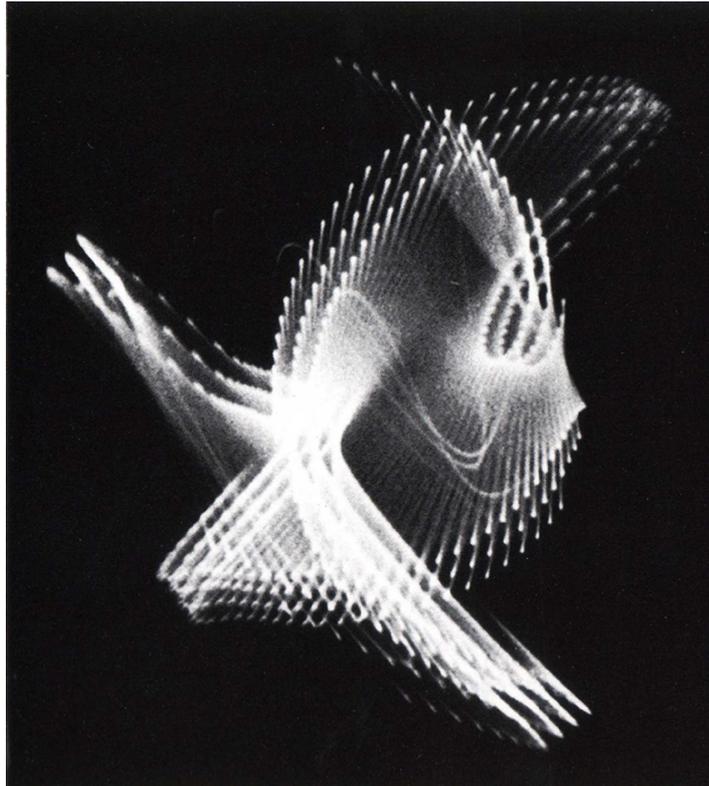


FIG. 2.1 – Oscillon Number Four, Ben Laposky, 1950

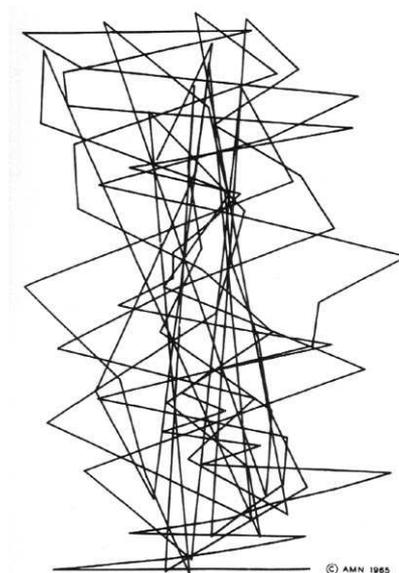


FIG. 2.2 – Gaussian Quadratics, Michael Noll, 1962



FIG. 2.3 – Fractal landscape, Ken Musgrave, 1993

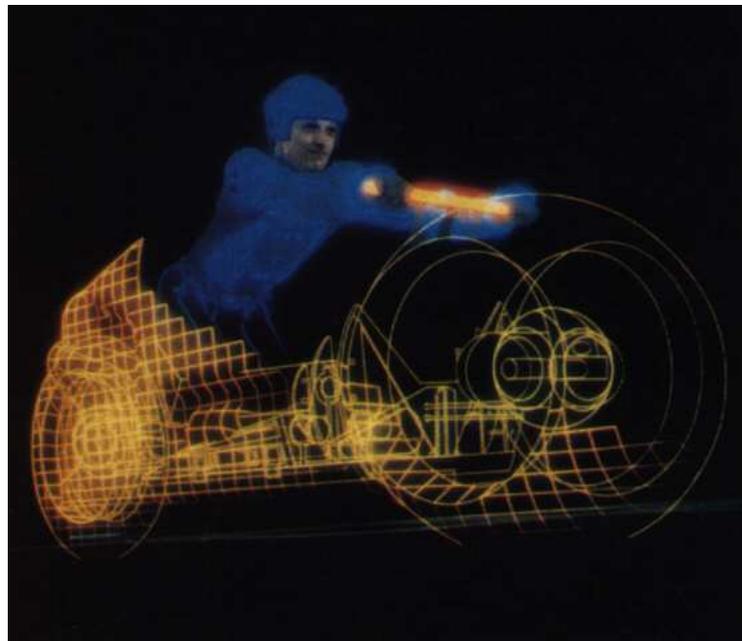


FIG. 2.4 – Un cycle du film Tron, MAGI, 1982



FIG. 2.5 – Une capture d'écran du jeu Myst, 1993

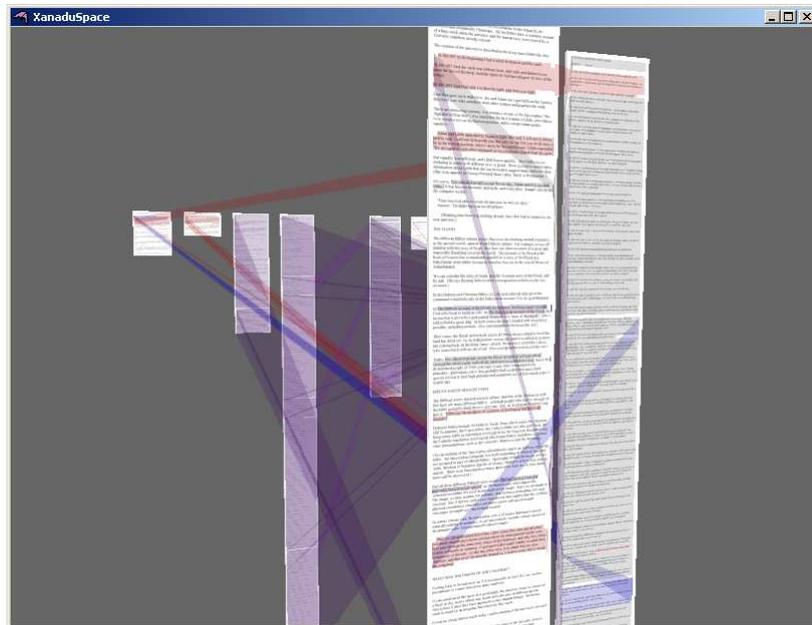


FIG. 2.6 – Une capture d'écran du navigateur 3D : Xanadu Space



FIG. 2.7 – Une capture d'écran du jeu Doom, 1993



FIG. 2.8 – Une capture d'écran du logiciel MovieStorm

Chapitre 3

Développement d'un système de modules communicants

3.1 La théorie

Comme illustré dans la figure 3.1, les différents modules s'articulent, par le biais d'un réseau, autour de fichiers communs. Ces fichiers décrivent les projets créés dans l'application, le contenu et l'historique des modifications apportées au projet, ainsi que l'état actuel du projet.

Chaque module, à son ouverture puis à intervalles réguliers, lit le fichier qui contient l'historique de modification. Lorsqu'une nouvelle modification est détectée, le module met à jour son contenu. Ainsi, même si deux modules sont utilisés en même temps, le temps de mise à jour sera quasiment indétectable.

La figure 3.4 présente le processus de synchronisation entre les différents modules. Il est à noter que certains modules sont indépendants, et ne seront pas concernés par les mises à jour. La figure 3.2 présente la hiérarchie de dépendance des différents modules de l'application.

3.2 Gestion de projets et utilisateurs

Le but de l'application est de fournir des modules afin que plusieurs personnes puissent travailler en collaborant, via un réseau local ou internet. Il a donc fallu mettre au point un système de gestion pour coordonner différentes personnes.

Moyens techniques

Pour cela, nous avons utilisé une base de données MySQL stockée sur un serveur. Dérivé du SQL¹ en 1995, le MySQL est un gestionnaire de données libre très utilisé sur Internet. Le MySQL désigne à la fois la structure de la base de données, et le langage utilisé pour interagir avec cette base de données.

Afin de faciliter la gestion des projets, et l'utilisation de l'application par les différents utilisateurs, une interface a été créée sous forme de pages web créées en PHP. Le PHP² est à l'origine une bibliothèque du langage Perl, et permet de construire des pages Hypertexte.

Contrairement aux langages HTML et JavaScript, qui sont interprétés par le navigateur de l'utilisateur, le PHP est interprété du côté du serveur. Il peut lire et/ou écrire des données dans une base MySQL, et en fonction de ces données crée une page Web pertinente.

¹Structured Query Language

²À l'origine PHP était l'acronyme de Personal Home Page tools (1994) puis est devenu l'acronyme récursif PHP : Hypertext Preprocessor (1998)

Mise en place

Deux tables (tableaux de données) MySQL ont été créées. La première contient les données des utilisateurs, la seconde les permissions.

Utilisateurs

La table ‘utilisateurs’ contient toutes les données concernant les personnes enregistrées, notamment le nom d'utilisateur, mot de passe, et description.

Permissions

Afin d'éviter des manipulations non désirées des données, chaque projet est associé à un certain nombre d'utilisateurs. Seuls ces utilisateurs ont le droit d'utiliser un module pour modifier le projet concerné.

Il est également possible de rendre un projet public, auquel cas n'importe quel utilisateur peut accéder au projet et le modifier.

Interface

L'interface est composée de 4 pages :

- login.php : elle sert juste à s'identifier en tant qu'utilisateur ;
- main.php ;
- new_project.php ;
- project.php .

La page “main.php”

Comme son nom l'indique, cette page est la principale de l'interface. Elle résume les données de l'utilisateur, et permet d'accéder aux différents projets ou d'en créer un.

La page “new_project.php”

Cette page permet de créer un nouveau projet, en indiquant le nom, les personnes autorisées (permissions), et la description du projet. Lors de la création d'un projet, les données sont ajoutées aux tables MySQL sur le serveur, et un dossier est créé contenant le premier fichier de données (l'historique du projet, cf 3.3 p. 22).

La page “project.php”

Cette page récapitule l'historique du projet, les fichiers déjà créés (scène, acteurs, animations, caméras, ...) , et donne accès aux différents modules disponibles pour créer et/ou éditer du contenu.

3.3 Protocole de communication

Le protocole de communication que nous avons choisi, est un ensemble de fichiers au format XML, contenant toutes les données nécessaires du projet.

Le format XML³ est un dérivé du format créé en 1986 par IBM : le SGML⁴. Ce format avait pour but de dissocier le fond et la forme d'un document. En utilisant des balises, le document ne contient que la structure du contenu, sans aucune information de présentation. Ainsi un document est lisible aussi bien par des humains que par des programmes. Le SGML est également à l'origine du protocole HTML⁵, utilisé pour afficher du contenu dans les navigateurs internet.

³EXTended Markup Language

⁴Standard Generalized Markup Language

⁵HyperText Markup Language

La structure du document est considérée comme un arbre, où chaque élément est une branche ou une feuille. Par exemple, dans le cas d'un livre, un document XML pourrait être écrit comme sur le listing ci-dessous.

```
<livre>
|   <auteur>Benoit Mandelbrot</auteur>
|   <année>1973</année>
|   <titre>Les Objets fractals : forme, hasard, et dimension</titre>
|   <contenu>
|   |   <chapitre numéro="1">
|   |   |   ...
|   |   </chapitre>
|   </contenu>
</livre>
```

Chaque projet peut contenir 5 types de fichiers :

- project.xml;
- scenes.xml;
- lights.xml;
- actor_name.xml;
- animation_name.xml;

Le fichier “project.xml”

```
<scp_project name="Demo">
|   <description>Description du projet</description>
|   <creation user="albert">27/07/2007-22 :36 :18</creation>
|   <edit user="bernard" type="lighting">28/07/2007-03 :24 :06</edit>
|   <edit user="charles" type="anim">28/07/2007-03 :35 :54</edit>
</scp_project>
```

Ce fichier, le plus important, et le seul à être vraiment nécessaire, décrit le projet, sa création et toutes les modifications qui y ont été apportées. Pour chaque modification, ce fichier enregistre la date, l'utilisateur ayant apporté la modification, ainsi que le module utilisé pour ajouter/modifier du contenu.

Le fichier “scene.xml”

```
<scene>
|   <object name="demo1">
|   |   <position x="3.02" y="1" z="4.5" />
|   |   <up x="0.00" y="1.00" z="0.00" />
|   |   <dir x="0.00" y="0.00" z="1.00" />
|   |   <scale x="1" y="1" z="1" />
|   |   <color r="255" g="128" b="255" />
|   </object>
|   <object name="demo2">
|   |   ...
|   </object>
</scene>
```

Ce fichier contient la description des éléments qui composent un décor. Évidemment la branche “objet” peut être répétée autant de fois que nécessaire.

Le fichier “lights.xml”

```
<lighting>
|   <light type="point">
|     |   <position x="0.00" y="0.00" z="0.00" />
|     |   <color r="128" g="128" b="255" />
|     </light>
|     <light type="directional">
|       |   ...
|     </light>
| </lighting>
```

Ce fichier, similaire au fichier “scene.xml” permet de décrire les éclairages d’une scène.

Le fichier “actor_name.xml”

```
<actor name="Eugene">
|   <parameter id="2" value="0.764" />
|   <cloth name="white_shirt">
|     |   <color r="125" g="34" a="78"/>
|   </cloth>
|   <prop name="sunglasses">
|     |   ...
|   </prop>
|   ...
| </actor>
```

Ce fichier décrit un acteur. Chaque paramètre correspond à une partie modifiable de l’acteur (taille du nez, des oreilles, corpulence, ...). Les balises “cloth” et “prop” servent à décrire les vêtements et accessoires de l’acteur.

Le fichier “anim.xml”

```
<animation actor="Fernand">
|   <action name="look_around" time="45"/>
|   <action name="walk_to" time="165">
|     |   <destination x="24.3" y="0.0" z="3.67"/>
|   </action>
|   ...
| </animation>
```

Dans ce fichier, les animations sont données successivement, chaque nom correspondant à une animation disponible dans la base de données.

3.4 Lecture et traitement des données

Les fichiers XML, de par leur structure, sont facilement lisibles par un programme. Aussi est-il relativement simple d’en extraire les données dont l’application a besoin.

Lors du chargement d’un module, le dossier du projet concerné est analysé, et chaque fichier XML est lu par un script PHP qui va décrire sous forme de tableau les données qu’il contient.

En utilisant le Building Block “Web Get Data” de Virtools, il est possible de récupérer les données structurées sous forme de tableau (le même que celui créé par la script php). Ce tableau est ensuite facilement exploitable par le module et permet de mettre à jour son contenu.

3.5 Gestion de la base de données

Chaque élément de la librairie est stocké dans un fichier séparé, sur le serveur. Lors de la lecture d'un fichier XML, pour les décors par exemple, la page PHP insère dans le tableau (qui sera téléchargé par Virtools) l'adresse du fichier en question.

Virtools possède des Building Blocks destinés à télécharger et intégrer des objets dynamiquement, donc cette étape est simple. Ensuite les objets sont placés/édités en fonction des paramètres décrits dans le fichier XML, puis regroupés dans un groupe (sorte de dossier interne à Virtools).

L'ensemble de ces étapes est décrit sur la figure 3.3 p. 26.

À l'ouverture de chaque module, en plus des fichiers concernant le contenu, Virtools télécharge les informations relatives au projet (cf 3.3 p. 22), qu'il stocke, lui aussi dans un tableau.

Ensuite, régulièrement (toutes les 15 secondes), Virtools télécharge le fichier de configuration "projet.xml" et le compare avec la dernière version enregistrée. Si une modification a été apportée, il le détecte et sait quel(s) module(s) il doit mettre à jour.

Dans chaque module, à tout moment, l'utilisateur peut sauvegarder son travail. Lorsque c'est le cas, Virtools envoie à un script PHP l'état du contenu 3D (personnage / animation / objets ...), qui est ensuite écrit dans le fichier XML correspondant. Ensuite le script PHP ajoute une ligne dans le fichier historique afin de signaler aux autres modules qu'une mise à jour a été faite (cf fig. 3.4 p. 27).

Cette façon de faire permet de garder une mise à jour quasiment instantanée lorsqu'un changement intervient dans un module en parallèle.

3.6 Utilisation des modules

Le module de création/édition des décors, comme tous ceux qui ont été créés pour "Faites du Clip", sont intégrés dans une page Web. L'interface se compose de deux panneaux Virtools (cf fig. 3.5). Le premier sert à afficher tous les éléments en 3D, le second sert d'interface (affichage des options, menus, ...).

Cette division en deux interface permet de guider l'utilisateur dans une sorte de bureau virtuel (un écran séparé de l'interface utilisateur, similaire à un bureau réel de type écran et clavier).

Le panneau inférieur permet de sélectionner les objets disponibles dans les différentes bibliothèques, et de les importer dans la scène courante. Une fois un objet importé dans la scène, il est possible de le déplacer, l'orienter, l'agrandir et/ou le réduire. Le panneau inférieur offre également la possibilité de modifier la couleur de l'objet.

Le module des lumières est très similaire au module de décors, si ce n'est que l'utilisateur ne peut modifier que les lumières. Celles-ci sont visualisées sous forme de lampes stylisées selon le type de lampe.

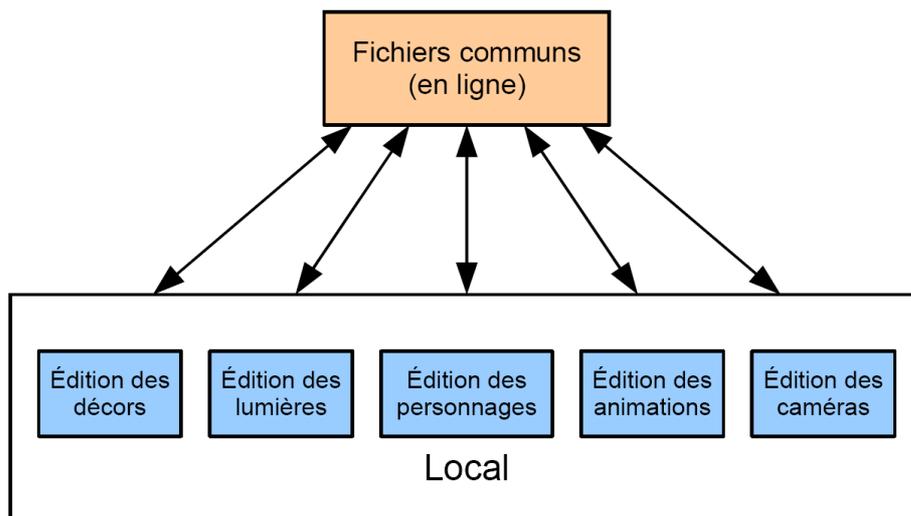


FIG. 3.1 – Organisation de l'application

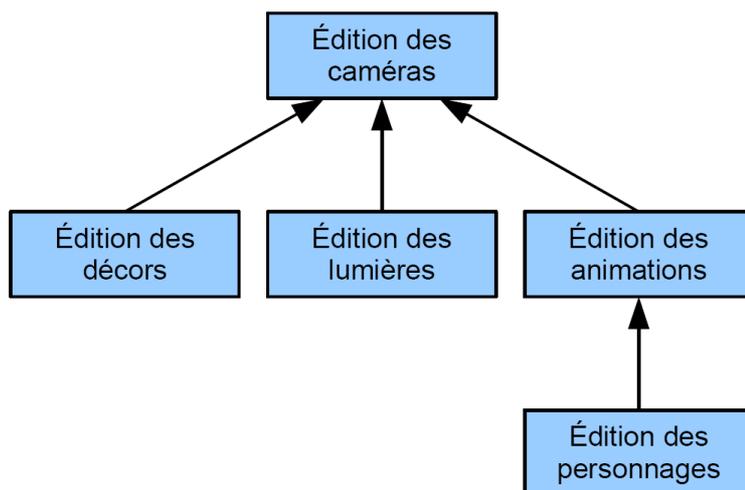


FIG. 3.2 – Hiérarchie de dépendance entre les différents modules

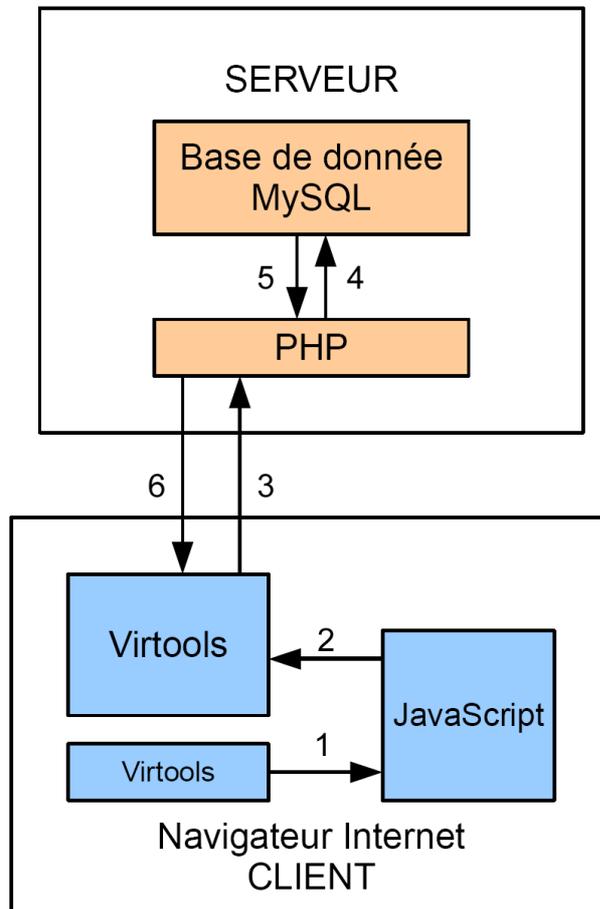


FIG. 3.3 – Processus de communication avec la base de données

1. Le premier panneau Virtools lance un message au navigateur internet, qui est ensuite géré en JavaScript.
2. La commande est alors transmise, toujours en JavaScript, au second panneau Virtools.
3. Le second panneau Virtools, au moyen du Building Block “Web Get Data”, envoie une requête HTML sur un script PHP située en ligne.
4. Le script PHP lance plusieurs requêtes MySQL à la base de données, situées sur le même serveur.
5. Le script PHP reçoit les réponses (contenu des bibliothèques 3D) et crée une page HTML avec ces données.
6. La page HTML est renvoyée, et récupérée par le panneau Virtools qui peut alors utiliser les données.

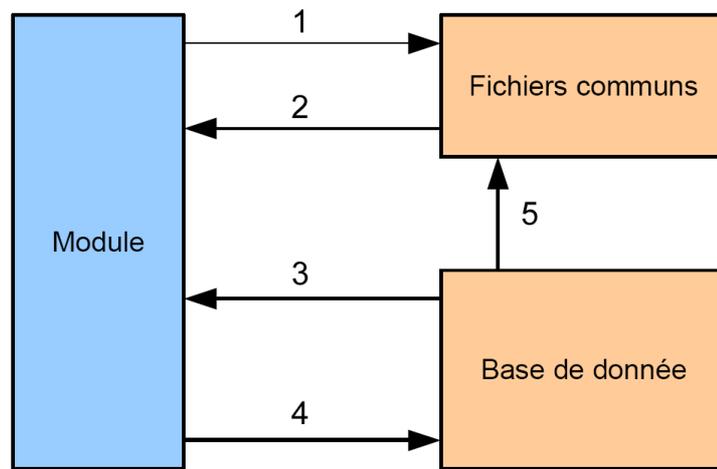


FIG. 3.4 – Processus de Communication entre les différents modules

1. Le module envoie une requête HTML à un script PHP pour récupérer le fichier historique XML.
2. Le fichier lui est renvoyé sous forme de tableau.
3. Si une modification est détectée, Virtools télécharge les fichiers nécessaires sur le serveur.
4. A l'enregistrement, Virtools sauvegarde/modifie le fichier correspondant dans la base de données via un script PHP.
5. Ce script va lui modifier le fichier historique, qui va propager la modification à tous les autres modules en cours d'utilisation.

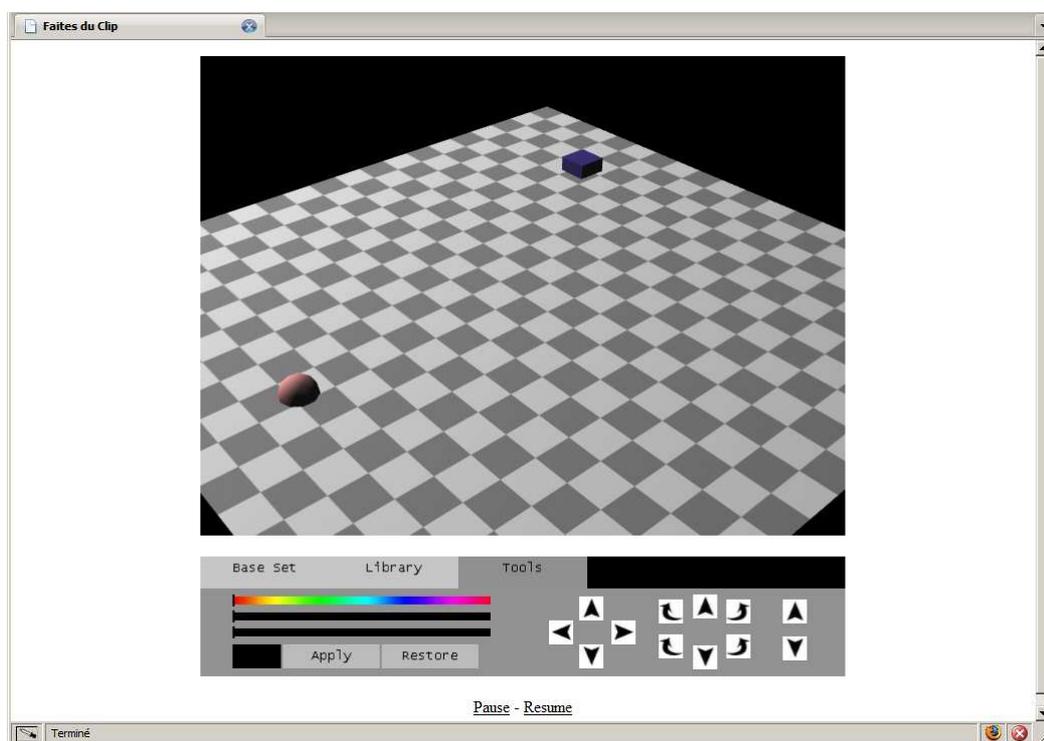


FIG. 3.5 – Découpage de l'interface de "Faites du Clip"

Annexe A

Rapport de Stage

Le stage a été réalisé du 2 mai au 28 septembre 2007, (sauf trois semaines en août pendant lesquelles j'ai présenté le projet SCP Camera au SIGGRAPH), au sein de la société Virtual-IT (<http://www.virtual-it.com>).

A.1 Présentation de la société Virtual-IT

La société Virtual-IT, anciennement Ondim, est une branche du groupe Kort & Lonn. Ce groupe réunit trois sociétés :

- Kort & Lonn Engineering, qui fournit des services dans les systèmes d'information ;
- Studio Chaperon Rouge, qui réalise des créations graphiques en 3D (modélisation, animation, ...) et 2D (design, plaquettes, ...);
- Virtual-IT, qui propose des services dans le domaine de la réalité virtuelle.

La société Ondim, créée en 1992, s'est rapidement spécialisée dans le domaine de la réalité virtuelle, dans la partie logicielle. Elle a réalisé ou participé à des projets dans divers domaines, dont certains cités ci-dessous.

- Un projet d'aide au guidage (système GPS) avec visualisation en 3D pour le tourisme et la navigation, en partenariat avec France Télécom.
- Un simulateur de maintenance avec un périphérique haptique, pour EADS ¹.
- Un système de simulation et d'analyse des nuisances sonores de centrales de production d'énergie, pour EDF.
- Un système d'aide à la reconnaissance d'itinéraires pour non-voyants, développé en partenariat avec PSA Peugeot Citroën (2002).

Le choix de mon stage s'est fait début avril 2007, où j'ai été contacté par l'agence Virtual-IT, qui cherchait une personnes ayant des compétences en Virtools. N'ayant jusqu'alors eu qu'une expérience professionnel autour de la réalité virtuel (mon stage réalisé en 2005 au Collège de France), cette oportunité m'a parut alors idéale.

La première partie de mon stage a principalement porté sur l'analyse des différents produits permettant de créer du contenu 3D temps réel disponibles sur le marché, pour savoir lequel répond le mieux aux besoins de Virtual-IT.

Ensuite, il m'a fallut analyser et modifier dans Virtools un simulateur industriel, afin d'en faire une application montrable et utilisable sur un salon.

Enfin, pour continuer le travail réalisé au début du stage, j'ai redéveloppé le même simulateur pour comparaison dans le logiciel Quest 3D.

¹European Aeronautic Defence and Space company

A.2 Compte-rendu de mes activités

Comparatif de logiciels Web 3D

Ce comparatif a été demandé afin de connaître les différences de quatre logiciels en matière d'utilisation, de capacité et de modulabilité. Ces logiciels sont :

- Virtools ;
- Quest 3D ;
- Eon Studio ;
- Nova Omega.

Ayant déjà une bonne connaissance de Virtools (presque trois ans d'expérience), j'ai concentré mon attention sur les trois autres logiciels, en commençant par Eon Studio.

Eon Studio

Le logiciel Eon Studio est principalement dédié aux simulations de présentations de produits, visites virtuelles et apprentissages pas à pas. Malgré un potentiel intéressant, le logiciel possède une ergonomie et une interface peu intuitive qui freine ses capacités.

L'ergonomie du logiciel apparaît d'emblée comme dense et surchargé. L'espace de travail est divisé en quatre panels contenant des listes et tableaux de propriété (cf fig. A.1).

Eon studio fonctionne sur une logique nodale : chaque élément est représenté sous forme de noeud possédant des propriétés (valeurs numériques, booléenne ou chaînes de caractères). Chaque propriété peut être liée à une autre, via un lien. Lorsque la première propriété est modifiée, le noeud auquel elle est attachée recalcule son état, et les modifications se transmettent de proche en proche.

Malgré la simplicité du mode de développement, le temps d'apprentissage et de développement reste élevé, principalement à cause de son interface peu ergonomique et trop sobre.

Quest 3D

Le logiciel Quest 3D, un peu plus jeune que Virtools, se place en concurrent direct de ce dernier. Ses principales applications sont les jeux vidéos et des visites virtuelles, mais peut également servir à des applications de réalité virtuelle immersive.

Le logiciel se base sur une logique de programmation extrêmement simple, ce qui allège grandement l'apprentissage mais aussi l'interface et en font un outil relativement puissant.

Toutes les données (paramètres, données 3D, textures, ...) sont gérées et visualisées d'une seule et unique façon : sous forme de boîtes appelées channel. Ces channels peuvent soit calculer une valeur à partir de leurs paramètres d'entrée, soit effectuer une action.

Cette simplicité allège beaucoup l'utilisation du logiciel : contrairement à Virtools, il n'existe pas d'interface spécifique pour gérer chaque type d'objet, mais une seule commune, qui est également utilisée pour gérer les interactions et comportement des éléments de la scène.

Le logiciel offre en outre de nombreux outils de gestion des scènes, et donne accès facilement à la façon dont le rendu doit être fait. Son moteur graphique, d'une excellente qualité, et les effets qu'il offre de base en font l'outil idéal pour créer des démonstrations graphiques.

Nova Omega

Le logiciel Nova Omega se destine particulièrement aux graphistes, afin de leur donner des outils pour réaliser des scènes simples aux interactions basiques.

Le logiciel est un plug-in ajouté au software 3D Studio Max. Dans l'interface même du logiciel 3D, des menus sont ajoutés pour attacher des propriétés aux éléments de la scène. Ces propriétés permettent de créer des interactions simples (détections des événements clavier/souris, lancement d'animation, déplacement d'un objet, modification de la visibilité/couleur/taille d'un objet, ...).

Cependant très rapidement l'utilisateur se retrouve limité et doit alors utiliser des langages de script pour obtenir des effets intéressants. En outre même le langage de script (Visual Basic) devient rapidement limité et il devient alors nécessaire de passer par une solution en SDK.

Ce logiciel, pratique pour créer des visites virtuelles ou des visualisations d'objets ou de modèle 3D, ne convient ni pour la réalisation de jeux vidéos, et encore moins pour des applications de réalité virtuelle.

Conclusions

Il sort de cette étude que les logiciels Quest 3D et Virtools sont relativement équivalents. Les deux possèdent des outils poussés, que ce soit en matière de gestion de scène ou de manipulation d'objets. Quest 3D, malgré son manque de maturité par rapport à Virtools, rattrape cette lacune par une interface pratique et un moteur graphique puissant.

Les deux autres logiciels ne sont pas suffisamment compétents et/ou maniables pour être utilisés dans le domaine de la réalité virtuelle.

Analyse et modification d'un simulateur industriel

Durant le mois de juillet, je me suis vu assignée la tâche d'analyser et modifier un simulateur² créé dans Virtools, afin d'en faire une version allégée.

Le simulateur est un projet réalisé pour une société qui construit des usines. Dans deux de ces usines, on utilise un pont roulant pour manipuler le matériel. La société a demandé la création d'un simulateur afin de former des ouvriers au maniement de ce pont roulant, pour éviter de mobiliser l'usine le temps de l'apprentissage.

Le simulateur utilisait de nombreux systèmes, notamment en matière de calculs physiques, afin de paraître le plus réaliste possible. Il utilise également un système de base de données permettant de conserver l'évolution de l'apprentissage grâce à différentes données (temps de réalisation des exercices, questionnaires, erreurs de manipulations, ...).

Analyse

Cette étape fut la plus fastidieuse, et la plus complexe, car il a fallu ouvrir et comprendre pas à pas chacun des soixante scripts et 30 codes sources C++. Le manque de commentaires, et la logique parfois incohérente, entrave le déchiffrement, ce qui prouve l'importance d'une bonne méthodologie et de l'usage de commentaires.

Le premier ensemble de script que j'ai analysé était l'un des plus importants : celui qui communique avec l'automate. L'automate étant un programme autonome simulant les réactions du pont roulant en fonction des actions de l'utilisateur. Ne possédant pas ce programme externe, il m'a fallu, alors que j'examinais les scripts, créer un automate qui me permettrait d'utiliser le simulateur.

²Pour des raisons évidentes de confidentialité, il m'est impossible d'inclure des captures d'écran de mon travail sur ce projet

L'automate dialoguant principalement via des valeurs binaires dans un tableau, il a fallu créer dans un Virtools un décodeur pour gérer simplement ces communications. En effet, un seul paramètre peut servir à donner jusqu'à 32 informations.

Un paramètre entier dans Virtools est codé sur 32 bits, ce qui signifie potentiellement 32 informations binaires. La plupart des paramètres échangés entre l'automate et le simulateur sont des valeurs booléennes (vrai/faux, comme par exemple si un bouton est appuyé ou si une option est active).

Une fois que l'on sait quel bit modifier, il suffit de faire une opération simple pour obtenir la nouvelle valeur du paramètre. La formule utilisée pour modifier le n^{eme} bit d'un entier x est la suivante :

$$x = \begin{cases} x + 2^n & \text{si } x \& 2^n = 0 \\ x - 2^n & \text{si } x \& 2^n = 2^n \end{cases}$$

En utilisant cette formule, j'ai pu créer un script VSL permettant de "switcher" le n^{eme} bit d'un entier.

```
// This script switches the nth bit in an integer
// Input parameters are :
// - Value (int), the initial value;
// - Bit (int), the bit's row.
// Output parameter is :
// - Result (int)
// © Xavier Gouchet, 2007
void main(){
|   if (Bit<32){
|   |   int switchbit = pow(2,Bit);
|   |   if (and(switchbit,Value)){
|   |   |   Value -= switchbit;
|   |   }else{
|   |   |   Value += switchbit;
|   |   }
|   }
}
```

Une fois cette partie réglée, la plupart des autres script furent plutôt simples à comprendre. Certains géraient simplement le moteur physique de la simulation, les autres interactions classiques (menu, pause, ...), les effets visuels, l'interface graphique, et le chargement des exercices.

Modification

La plupart des modifications ont servi à s'affranchir de toutes applications tierces, c'est-à-dire l'automate et la base de données où sont sauveées les données de chaque utilisateur. Ces modifications avaient déjà été commencées à l'étape précédente.

Pour l'automate, il m'a suffit d'utiliser un clavier et de lier chaque touche à une action. Pour la base de donnée, j'ai tout simplement enlevé la partie de communication, et stocké les données dans un tableau Virtools, lui même enregistré dans un fichier sur le disque dur local.

Il y a eu en plus de cela quelques autres modifications mineures pour adapter le projet à une utilisation sur un ordinateur de bureau (notamment par rapport à l'affichage qui était réglé pour l'environnement immersif de la cabine où se faisait la simulation). J'ai ensuite transféré les interactions du clavier vers une manette de jeu possédant deux joysticks analogiques, l'interface qui se rapprochait le plus de la cabine du simulateur (qui possède également deux joysticks).

Enfin, une dernière partie, moins technique, fut la rédaction d'un mode d'emploi/tutoriel pour le simulateur, destiné aux commerciaux qui présenterai l'application dans des salons ou chez des futurs clients.

(Re)développement d'un simulateur

Dans la continuité du comparatif que j'avais réalisé au début de mon stage, il m'a été demandé de refaire un simulateur le plus proche possible de celui qu'ils avaient, en utilisant l'environnement de développement de Quest 3D.

Cela m'a permis de me confronter avec les contraintes de réalisation de ce logiciel, et, en utilisant mes connaissances en Virtools de mieux cibler le potentiel de Quest 3D.

La principale différence qui existe entre Virtools et Quest 3D réside dans la logique de programmation. Pour simplifier, dans Virtools, chaque boîte comportementale est activée séquentiellement. Ainsi, au démarrage, le signal d'activation passe par la première boîte, puis lorsque le code de cette boîte est effectuée, se transmet à la (aux) boîte(s) suivante(s). Si nécessaire, le signal d'activation peut être retardé, ou arrêté.

Dans Quest 3D, le signal parcourt toutes les boîtes à chaque frame. Il ne peut ni être retardé, ni arrêté. Par contre, il est possible, au moyen de certaines boîtes, de l'empêcher de passer dans certaines boîtes. Ce mode de programmation se rapproche beaucoup de l'électronique, plus que le système de Virtools. Il rappelle également la programmation comportementale dans Motion Builder, ou la structure nodale de l'hypergraphe de Maya.

Cependant, étant habitué à la logique de Virtools, il m'a été compliqué au début de m'adapter à cette nouvelle façon de penser. On voit sur les figures A.3 et A.4 la différence de logique pour retarder l'exécution d'une boîte ("Do Something") de 100 frames.

Dans Virtools, il suffit de faire boucler le signal sur la boîte "Counter", et d'attendre que celui-ci arrive à 100 pour transférer le signal à la boîte "Do Something". Dans le second cas, si la variable A possède une valeur inférieure à 100, on l'incrémente, sinon, on entre dans la boîte "Do Something".

Ceci n'est qu'un exemple parmi d'autres, mais présente la différence de logique entre les deux logiciels. Les autres différences sont minimes (interface, raccourcis clavier, nom des fonctions) et sont rapidement assimilées.

Une autre partie qui a nécessité de nombreux réglages fut celle du moteur physique. En effet, dans la version Virtools, le moteur avait été développé en interne et intégré via un SDK au simulateur. Cependant, je n'avais ni le temps, ni les connaissances nécessaires pour redévelopper un moteur physique pour Quest 3D, d'autant que ce dernier en possède un de base.

J'ai donc examiné leur pack physique, basé sur le moteur open-source ODE³. Parfaitement intégré à Quest 3D, il est également très simple à utiliser, de par la logique utilisée par l'environnement temps réel.

En effet, contrairement à Virtools, Quest 3D "oblige" l'utilisateur à gérer lui-même son rendu, c'est-à-dire choisir pour chaque caméra les objets concernés, et pour chaque objet, décrire dans l'interface schématique les composants de l'objet (mesh, matériaux et matrice). Pour déclarer un objet comme physique, il suffit d'ajouter un Channel entre l'objet et sa matrice, qui devient alors la position initiale de l'objet.

A.3 Bilan du stage

Bilan des acquis

Ce stage, portant principalement sur les technologies 3D temps réel, m'a permis de conforter les connaissances que je possédais au préalable, et m'en a offert de nouveaux.

L'analyse du simulateur réalisé sur Virtools a beaucoup enrichi mes connaissances et ma façon d'appréhender le logiciel, que je connais et utilise depuis février 2004.

³Open Dynamic Engine

Ce projet m'a donné l'occasion de découvrir un nouveau mode d'organisation, du point de vue du développement, mais aussi directement à l'intérieur du logiciel. En effet, j'ai appris notamment comment les scripts VSL étaient gérés par Virtools, et comment tirer profit de cette gestion.

De même, l'étude des différents plug-ins a conforté mes connaissances en C++, et dans la gestion de plug-ins Virtools. Certains plug-ins gérant les communications avec les applications tierces m'ont également données des bases de connaissances sur les communications en réseau.

Les comparatifs réalisés sur Quest 3D, Nova Omega et Eon Studio m'ont également apporté une expérience sur d'autres logiciel que Virtools. Chacun de ces logiciels possède sa propre philosophie, ses propres logiques, son propre langage, dont l'étude m'a forcé d'appréhender l'univers de la 3D temps réel sous de nouveaux angles.

Ouvertures

En plus de toutes ces connaissances, ce stage m'a été très profitable d'un point de vue personnel. J'ai eu un contact direct et nouveau avec le monde professionnel dans la recherche et le développement d'applications 3D temps réel, et les différents contacts que j'ai eu m'ont permit de mieux cibler mes ambitions quant à mon avenir professionnel.

En outre, les connaissances acquises durant ce stage ont également profité au projet "Faites du Clip" et à sa réalisation.

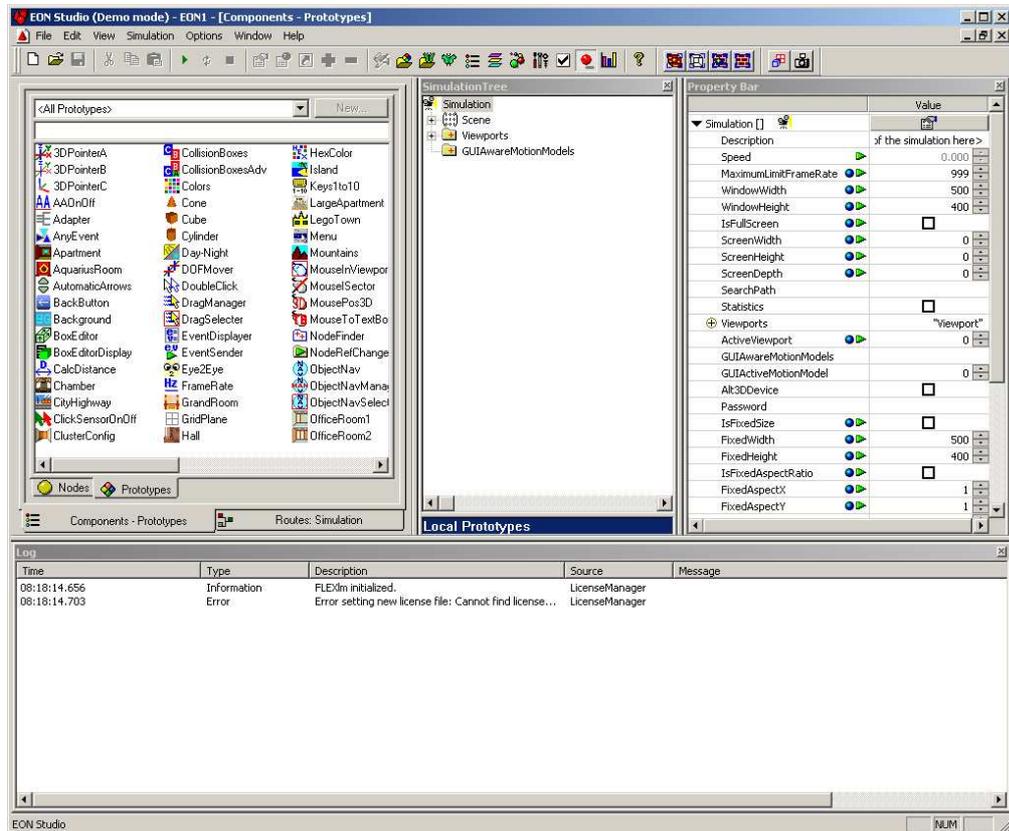


FIG. A.1 – Capture d'écran du logiciel Eon Studio

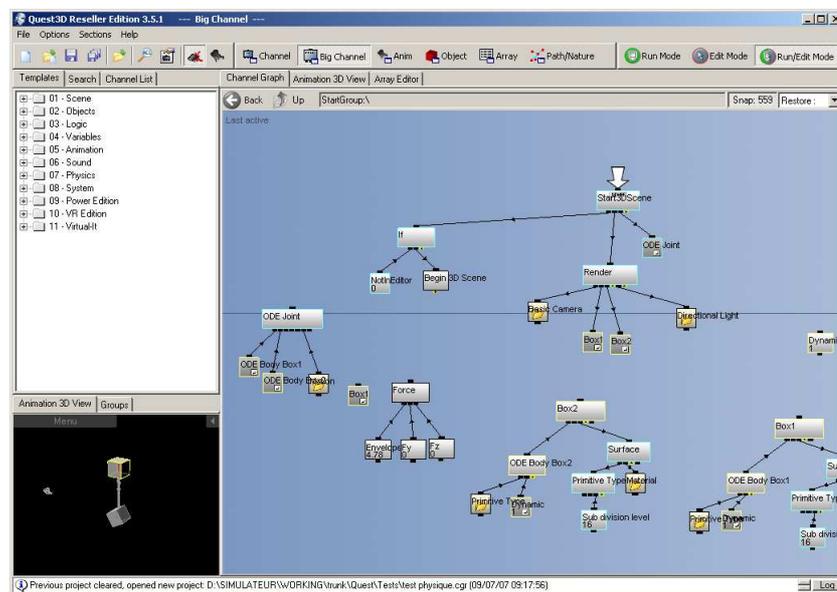


FIG. A.2 – Capture d'écran du logiciel Quest 3D

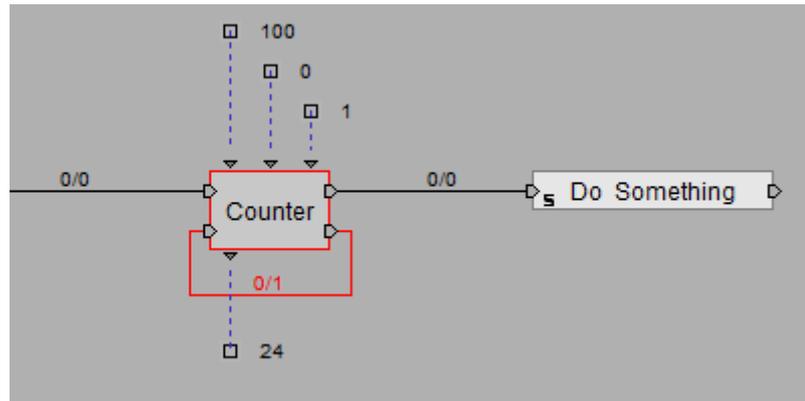


FIG. A.3 – Script retardant l'exécution d'une boîte (Virttools)

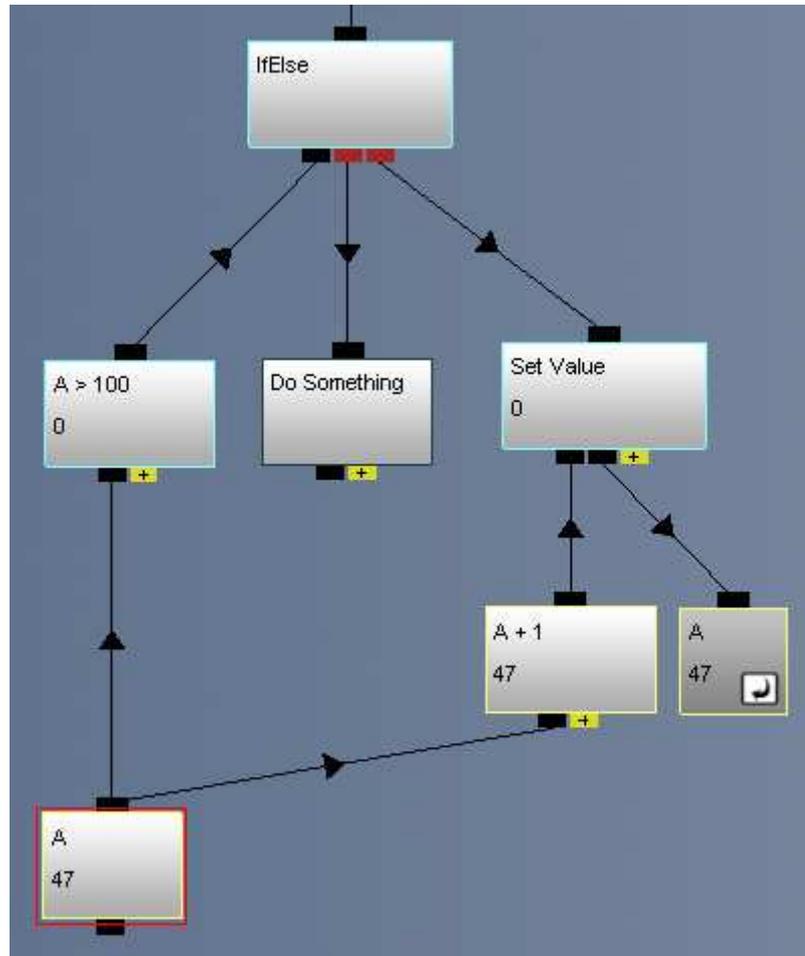


FIG. A.4 – Script retardant l'exécution d'une boîte (Quest 3D)

Sources

Bibliographie

- Perry Hoberman, “Webbed Spaces : Between Exhibition and Network” – SIGGRAPH 1996 conferences
Don Brutzman, “VRML : prelude and future” – SIGGRAPH 1996 conferences
Celia Pearce, Mark Pesce, Erik Davis, “The soul of the machine : the search for spirituality in cyberspace” – SIGGRAPH 1996 conferences
Martin Isenburg, “Geometry Compression for ASCII Scenes” – SIGGRAPH 2002 papers
Alan D Hudson, “The Xj3D Browser : Community-Based 3D Software Development” – SIGGRAPH 2002 conferences
Christian Babski, “_knowscape a 3D multi-user experimental web browser ” – SIGGRAPH 2002 conferences
Tony Parisi, “Flux : lightweight, standards-based Web graphics in XML” – SIGGRAPH 2003 papers
Chris Thorne, “Exploiting an evolutionary accident in web3D communications to integrate application components” – SIGGRAPH 2005 papers

Webographie

Sites Généraux

- Encyclopédie en Ligne Wikipédia : www.wikipedia.org
Site généraliste sur la programmation : www.developpez.com/
Histoire d’Internet : <http://smithsonian.yahoo.com>
Histoire de l’infographie : <http://accad.osu.edu/~waynec/history/timeline.html>
Article sur le Net Art : http://en.wikipedia.org/wiki/Net_art
Article sur les communautés en ligne : http://en.wikipedia.org/wiki/Virtual_community
Articles sur les technologies Web 3D : www.tdt3d.com/

Sites Spécifiques

- Page personnelle de Mark Pesce : www.hyperreal.org/~mpesce/
Mark Pesce, ‘Cyberspace’ (Conférence du W3C 1994, Genève) : www.hyperreal.org/~mpesce/www.html
Histoire du VRML : <http://vtmlworks.crispen.org/history.html>
Site du Web 3D Consortium : www.web3d.org/
La page du projet Xanadu : www.xanadu.com/
- Site web du projet SCP Caméra : www.scpcamera.com
Site web de la société de préviz Halon Entertainment : www.halon.com/
Site web de la société Virtual-IT : www.virtual-it.com/
Site web du groupe Kort & Lonn : www.kort-lonn.fr/

Software

Virtools : www.virttools.com/

Quest 3D : www.quest3d.com/

Subdo : www.subdo.com/

Flux Player (plug-in VRML) www.mediamachines.com/

Movie Storm : www.moviesstorm.co.uk/

Sites Net Art

BlueScreen (ensembles d'oeuvres basées sur les technologies internet) : <http://b-l-u-e-s-c-r-e-e-n.net/>

Le site de Mark Napier (et de son "Shredder") : www.potatoland.org/

"Form Art" d'Alexei Shulgin : www.c3.hu/collection/form/

Un annuaire d'oeuvres Net Art : <http://netartconnexion.net/>

"Pixel Wall", une oeuvre participative : <http://wall.versatile.fr/>

"Web2DNA Project", un site qui convertit une page web en ADN : www.baekdal.com/web2dna/

Remerciements

J'aimerais remercier avant tout et en premier Rémi Quittard et Nicolas Serikoff, ainsi que Célia Demere et Chloé Lang-Willar, qui m'ont supporté pendant près de deux ans, et sans qui les projet SCP Caméra et "Faites du Clip" n'en seraient pas là où ils sont aujourd'hui.

Je souhaite également remercier Marie-Hélène Tramus et Cédric Plessiet, ainsi que toute l'équipe enseignante du département Arts et Technologies de l'Image de l'Université Paris 8, qui nous ont soutenu et encadré sur ce projet depuis deux ans.

Je remercie bien sur tous les partenaires qui nous ont permis de présenter notre projet aux Emerging technologies, durant le SIGGRAPH 2007 à San Diego CA :

- Immersion ;
- Cap Digital ;
- la Comex⁴ et le Conseil Général de la Seine Saint-Denis ;
- Virtools et 3Dvia ;
- Adhesia ;
- CopySud.

Je remercie pareillement Hervé Guillemot, Régis Courtalon, Kamel Hasni et Karim Guennoun – qui m'ont accueilli au sein de la société Virtual-IT – avec qui ce fut un plaisir de travailler durant mes quatre mois de stage.

Enfin, je tiens à remercier ma compagne, mais aussi mes parents et ma famille, qui m'ont soutenue tout au long de cette année 2006-2007, et qui continuent de le faire.

⁴Comité d'expansion économique de la Seine Saint-Denis