
Cahier des charges

PFA - De la 3D vers la 2D

Année scolaire 2014-2015

Client : BLANC Carole, DESBARATS Pascal

Encadrant : LOMBARDY Sylvain

Equipe : CABON Yohann - CHAUVAT Magali - LEVY Akané

MARCELIN Thomas - MAUPEU Xavier

PHILIPPI Alexandre - BOHER Anaïs



Table des matières

1	Domaine	3
2	Etat de l'art et existant	4
2.1	Logiciels existants	4
2.2	Projection vers deux dimensions	4
2.3	Les flipbooks	5
2.4	Les autostéréogrammes	5
2.5	Les anaglyphes	8
3	Sujet	9
	Cahier des besoins	10
4	Besoins fonctionnels	10
4.1	La scène	10
4.2	Les objets	10
4.3	La caméra	11
4.4	Le passage en deux dimensions	11
5	Besoins non fonctionnels	13
5.1	La fluidité d'affichage	13
5.2	Fluidité d'obtention des rendus	13
5.3	Portabilité du logiciel	13
5.4	Extensibilité du logiciel	13
6	Contraintes	13
6.1	Superposition des objets dans la scène	13
6.2	Utilisation des bibliothèques	13
6.3	Langage de programmation	14
6.4	Open Source	14
7	Briques logicielles	14
8	Maquette	14
9	Prototype test	14
10	Architecture du projet	14

1 Domaine

Ce projet s'inscrit dans le domaine de la synthèse d'images et de la visualisation de modèles en trois dimensions. Depuis le quinzième siècle, grâce à la peinture, la perspective apparaît sur des supports en deux dimensions. Aux XIXème et XXème siècles, l'utilisation de stéréoscopes, tel que le stéréoscope de Holmes, permettait la visualisation de relief à partir de deux images planes et d'un dispositif optique. Dans la deuxième moitié du XXème siècle, l'utilisation du numérique permet de modifier les images et d'obtenir une meilleure visualisation de la profondeur sur des supports en deux dimensions.

On peut ainsi créer des anaglyphes, des autostéréogrammes ou des flipbooks, qui sur papier ou sur écran permettent d'apercevoir la profondeur d'une scène grâce à des techniques adaptées. Ces différents rendus seront présentés plus tard dans ce cahier des charges. De nos jours, il existe également des logiciels, tels que Meshlab et Blender qui sont gratuits, open source et permettent d'ores et déjà la visualisation en trois dimensions sur un écran. L'utilisateur peut tourner autour d'un objet et le voir sous tous ses angles grâce à un ensemble de projections successives autour de l'objet.

On appelle synthèse d'image l'ensemble des techniques qui permettent de visualiser des objets en trois dimensions en perspective sur un écran d'ordinateur, en tenant compte de lumières et de textures appliquées à l'objet. Il existe un grand nombre de techniques et les résultats obtenus peuvent eux aussi varier (perspective isométrique, perspective conique...). Nous nous préoccupons par la suite de la perspective conique, dite aussi vue naturelle.

Bien souvent, la synthèse d'image utilise le principe de scène. Il s'agit d'un espace à trois dimensions dans lequel des objets peuvent être placés. Ces derniers sont décrits par un ensemble de points placés dans l'espace.

Pour pouvoir observer la scène et les objets, il est nécessaire de demander à l'ordinateur de les modéliser, c'est-à-dire d'afficher un rendu qui correspondrait à une vision de cette scène si elle était réelle. Pour cela, la machine simule le point de vue de l'utilisateur à l'aide d'une « caméra ». A partir de cette scène en trois dimensions, la caméra peut réaliser des projections ou photographies permettant de créer des anaglyphes, autostéréogrammes ou flipbooks. Plusieurs méthodes de projection existent, mais seule celle par matrice de projection sera utilisée.

Ces matrices sont décrites à l'aide de coordonnées homogènes. Celles-ci ont été introduites afin que l'ensemble des transformations de type rotation, translation et homothétie puissent être écrites sous forme de matrice. Ainsi, le produit des matrices de transformation peut être calculé en amont pour pouvoir appliquer la matrice de la transformation résultante à l'ensemble des points de l'objet sans avoir à recalculer le produit pour chaque point.

Pour pouvoir visualiser un modèle 3D il faut prendre en considération la lumière et sa réflexion sur l'objet. Si une sphère rouge était représentée dans un espace avec uniquement une lumière ambiante, il n'en ressortirait qu'un disque rouge, sans relief. En effet, la lumière ambiante atteint l'objet de la même façon en tout point. On ne peut donc pas savoir depuis un plan fixe s'il s'agit d'un objet en deux ou en trois dimensions. Si maintenant une lumière est ajoutée dans l'espace où est situé l'objet, celle-ci ne va pas atteindre tous les points de l'objet de la même façon. Elle sera plus faible sur un point plus éloignée, voire inexistante sur un point caché. En tenant compte de cette lumière, on peut obtenir une image comme présentée sur la figure 1.1.

1. <http://linut.free.fr/omgsp10kuberwebbloglolz0r/?2010/02/01/93-raytracer-que-la-lumiere-soit>

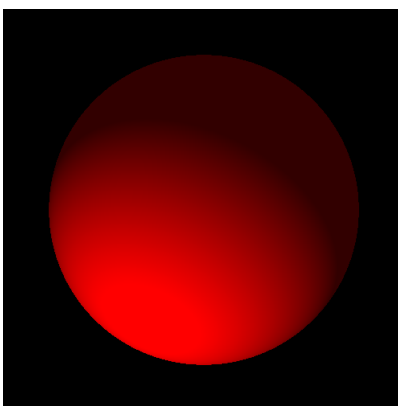


FIGURE 1.1 – Application d’une lumière diffuse à une sphère rouge ¹

Pour la création d’un anaglyphe, deux images espacées par une petite distance (qui correspond à la distance entre les deux yeux par exemple) sont générées. La composante rouge de l’une de ces images et la composante bleue de l’autre sont gardées et ensuite superposées dans une même image. Cette image est ensuite transformée en une image Rouge-Cyan, qui peut être visualisée à l’aide de lunettes Rouge-Bleue : l’image apparaît en trois dimensions.

Pour la création d’un autostéréogramme, une image permettant d’observer un objet en relief par vision parallèle est générée. Cette image est obtenue à partir d’une texture de base ou de points aléatoires pour l’image de fond.

Pour la création d’un flipbook, plusieurs images sont prises à intervalles réguliers par une caméra suivant un trajet prédéterminé dans ou autour de la scène. Le flipbook est visualisable en faisant rapidement défiler ces images tout en respectant l’ordre des prises de vue. Ce flipbook peut être transformé en GIF pour obtenir une visualisation animée des images.

2 Etat de l’art et existant

2.1 Logiciels existants

StereoPhoto Maker ² et Anaglyph Maker 3D ³ sont des freewares permettant, entre autre, depuis deux photos d’obtenir l’anaglyphe rouge / cyan correspondant. La qualité de l’anaglyphe obtenu dépend du décalage entre les deux photos et de la qualité de la prise de vue.

Enfin, de nombreux logiciels de création d’autostéréogrammes sont disponibles, comme Stereographic Suite (©IndaSoftware) et 3DMiracles (©Urry Software Lab) qui permettent de créer des autostéréogrammes à partir d’une carte de profondeurs (image en deux dimensions en niveaux de gris).

2.2 Projection vers deux dimensions

Le passage d’un objet en trois dimensions à une image en deux dimensions se fait par projection. Une caméra présente dans la scène est déterminée par sa position et le vecteur direction qui indique vers quel

2. StereoPhoto Maker : <http://stereo.jp.n.org/eng/stphmkr/>

3. Anaglyph Maker 3D : http://www.stereoeye.jp/software/index_e.html

endroit elle regarde dans la scène [1]. La vue elle-même est obtenue par projection de la scène dans le repère de la caméra comme sur la figure 2.1.

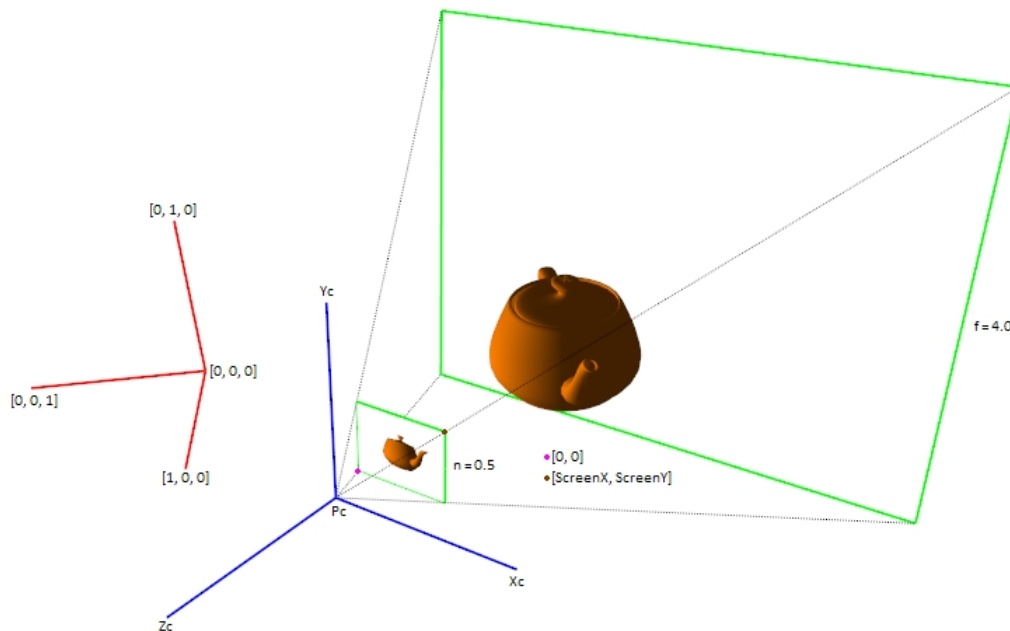


FIGURE 2.1 – Projection d'un objet en trois dimensions sur un support en deux dimensions ⁴

2.3 Les flipbooks

Le principe d'un flipbook, ou folioscope en français, est de créer une suite d'images successives d'une scène par rapport à une trajectoire. Il suffira ensuite de mettre toutes ces images dans l'ordre les unes derrière les autres, et de les faire défiler rapidement pour avoir l'impression d'un rendu en relief et en mouvement comme sur la figure 3. C'est également le principe des fichiers d'extension .GIF, qui font défiler une liste d'images.

La création d'un flipbook est possible en créant une animation à l'aide d'un logiciel de manipulation d'objets 3D et en ne capturant que certaines images, par exemple avec Blender. Ainsi l'impression de ces images successives permet de réaliser un flipbook (cf. figure 2.2). En combinant par exemple avec le logiciel Gimp (outil d'édition et de retouche d'image) l'animation peut être obtenue en GIF.

2.4 Les autostéréogrammes

Un autostéréogramme est une image qui cache une visualisation en trois dimensions d'un objet. La visualisation en trois dimensions peut être difficile à obtenir, et demande une réelle gymnastique oculaire. Cette image est construite de sorte qu'à chaque point de l'objet en trois dimensions soient associés deux points de l'image. L'utilisateur doit observer chaque point d'un couple de points avec un seul œil, ce qui donne l'illusion au cerveau d'observer deux images différentes avec les deux yeux et donc l'incite à traiter cette information comme l'observation d'un objet en trois dimensions, comme illustré par la figure 2.4.

4. OpenGL perspective projection : <http://www.3dcpptutorials.sk/index.php?id=2>

5. <http://tracieliu.blogspot.fr/2010/08/flipbook-storyboard.html>

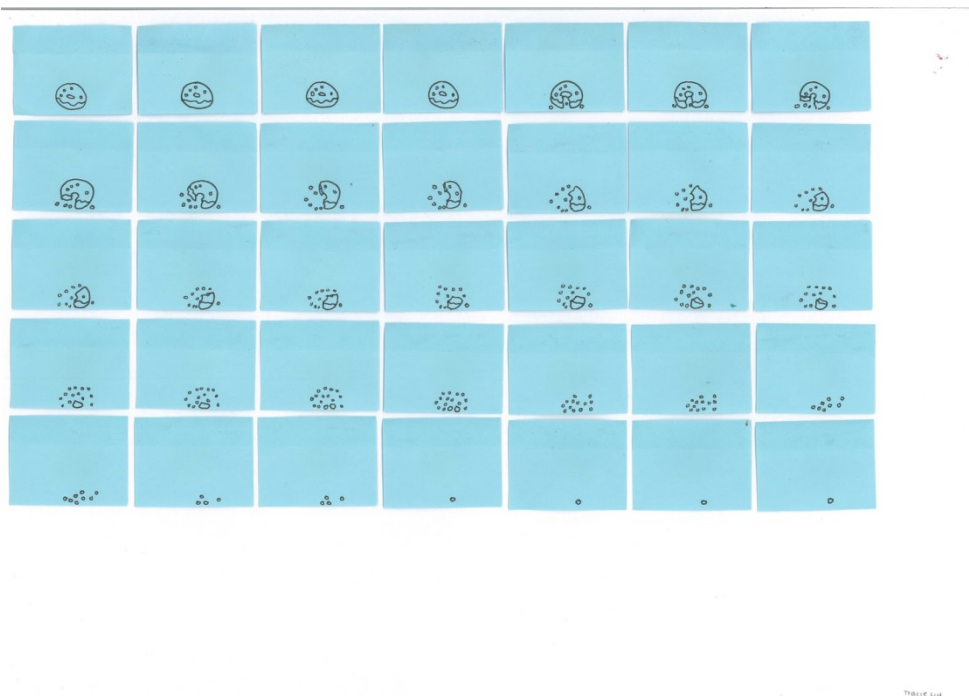


FIGURE 2.2 – Story-board d'un flipbook⁵

La visualisation en trois dimensions peut être difficile à obtenir, et demande une réelle gymnastique oculaire. Il faut pouvoir fixer le regard en avant ou en arrière de l'image, pour réussir à y voir l'objet caché. La plupart des autostéréogrammes sont observables en vision parallèle, c'est-à-dire qu'il faut faire le point au-delà de l'image pour pouvoir observer l'objet.

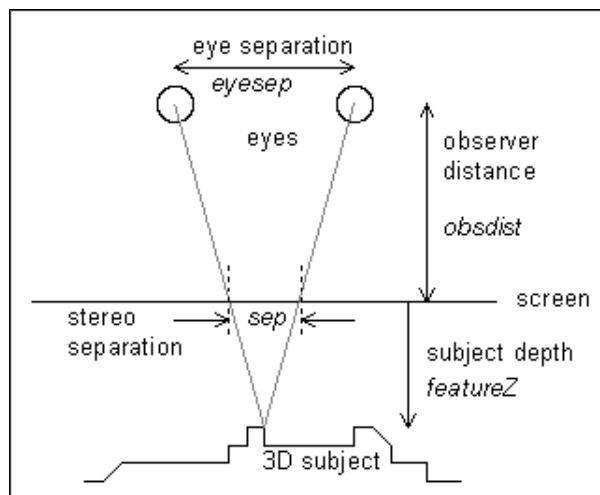


FIGURE 2.3 – Visualisation d'un autostéréogramme en vision parallèle⁶

6. <http://www.techmind.org/stereo/geometry.gif>

Pour générer un autostéréogramme, il faut utiliser une carte des profondeurs, ou carte de disparité, de l'objet à dissimuler. Cette carte s'obtient grâce à deux visions d'une même scène prises à deux endroits différents, et permet de mettre en avant les informations sur la profondeur de l'objet. Par exemple, la carte des profondeurs de la figure 2.4 a permis d'obtenir l'autostéréogramme de la figure 2.5.

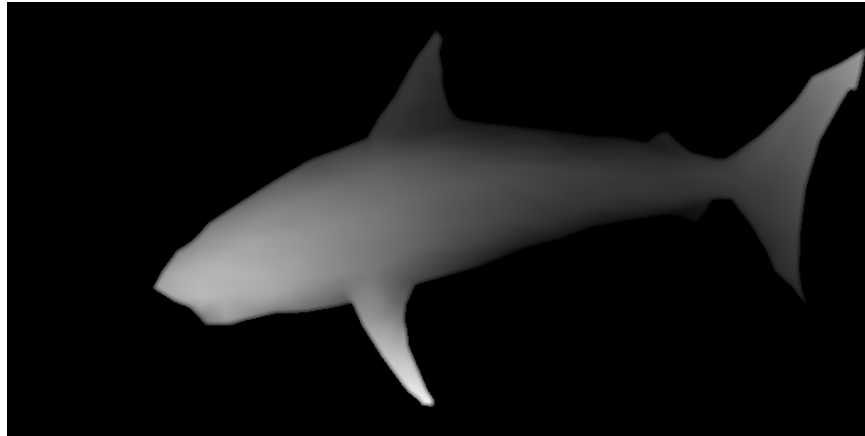


FIGURE 2.4 – Carte des profondeurs⁷

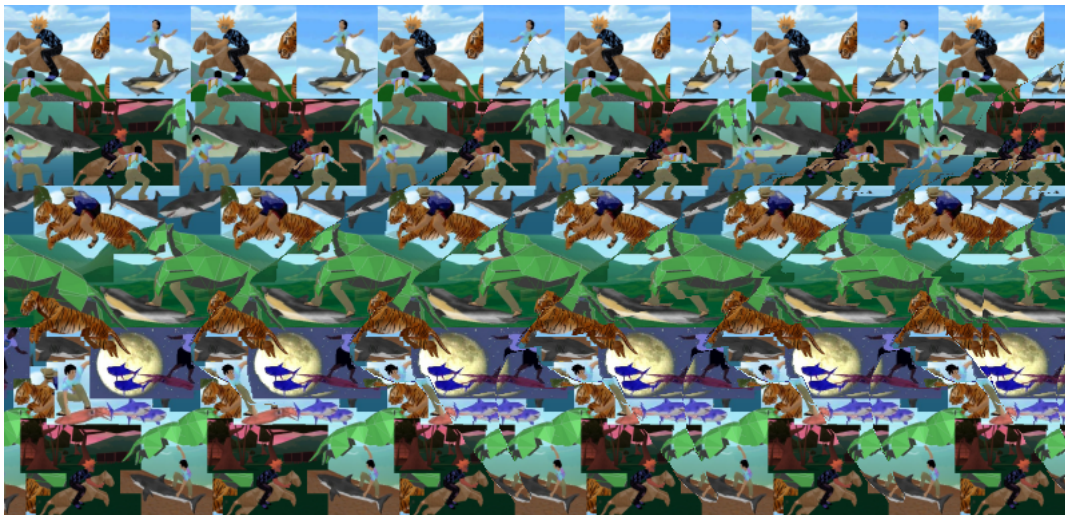


FIGURE 2.5 – Autostéréogramme obtenu⁸

Il existe des algorithmes de génération d'autostéréogrammes très simples, comme celui proposé par Gary Beene [3]. Cependant un algorithme aussi peu optimisé produit des autostéréogrammes défectueux, comportant par exemple des échos (répétition d'une partie de l'objet 3D) ou des artefacts (défaut de l'objet 3D observé).

7. <http://en.wikipedia.org/wiki/Autostereogram>

Des algorithmes plus poussés ont été développés, comme celui présenté par Harold W. Thimbleby, Stuart Inglis et Ian H. Witten [4], et celui proposé par W. A. Steer [1], associés à une réalisation en C. Ils permettent la génération d'autostéréogrammes SIRDS (Single Image Random Dots Stereogram) à partir d'une image 2D. Ces articles présentent également des méthodes de correction de problèmes tels que l'écho ou la gestion des faces cachées (faces de l'objet tridimensionnel visibles par un seul œil). W. A. Steer propose de plus une méthode de génération d'autostéréogrammes utilisant un motif bitmap comme image de base plutôt qu'un nuage de points aléatoires, ce qui pallie à certaines limites des SIRDS comme le manque de détails et la grossièreté des surfaces courbes.

2.5 Les anaglyphes

Un anaglyphe est une image sur laquelle on superpose deux vues, si possible différentes, d'une scène. La meilleure distance entre ces deux visions est la même que celle entre les deux yeux, afin que le cerveau puisse recréer la même vision en trois dimensions que dans la réalité.

Les anaglyphes les plus fréquents sont les anaglyphes dits rouge-cyan. Ils se nomment ainsi car ils sont constitués d'une image sur laquelle on passe un filtre magenta, et une autre avec un filtre cyan (cf. figure 2.6). Pour pouvoir visualiser le relief sur une telle image, on utilise une paire de lunettes rouge-cyan, dont chaque verre est un filtre pour l'une des deux couleurs de l'image. Le plus souvent, le filtre magenta est placé sur l'œil gauche, le cyan sur l'œil droit. En regardant l'image, l'œil gauche ne verra alors que la composante cyan, et inversement pour l'œil droit. Les deux images ayant un léger décalage, le cerveau va percevoir l'image comme si elle était en trois dimensions. Les anaglyphes rouge-cyan sont principalement intéressants sur des images en noir et blanc. En effet, quand il s'agit d'images en couleur, celles-ci sont souvent détériorées par l'usage des filtres, car les couleurs sont généralement composées de plusieurs composantes. A l'inverse, quand l'image est en nuances de gris, l'image n'est pas modifiée, juste mise en relief.

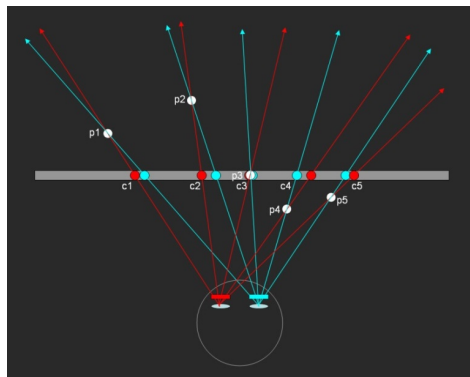


FIGURE 2.6 – Le décalage de la partie rouge et cyan permet à l'œil de percevoir l'image non plus dans un plan XY mais dans l'espace¹⁰

Le paragraphe précédent concernait la création d'un anaglyphe depuis deux prises de vue d'une même scène avec un léger décalage. Mais il est également possible de le faire à partir d'une image en deux dimensions grâce à l'algorithme de Dubois [2], qui définit une vision matricielle de l'anaglyphe (cf. figure 2.7). Dès lors, une matrice de transformation est introduite, qui permet de passer du taux RGB d'un pixel

8. <http://en.wikipedia.org/wiki/Autostereogram>

10. <http://www.david-romeuf.fr/3D/Anaglyphes/TCAnaglypheLSDubois/TransformationCouleursPourAnaglyphe.html>

de l'image originale, aux deux taux RGB des anaglyphes gauche/droite. Les coefficients de cette matrice sont calculés pour satisfaire une bonne restitution de l'image originale, tout en supprimant les rivalités colorées induites par des lunettes bicolores. Ainsi, pour visualiser une image rouge pure, il faudra changer le taux RGB de ce pixel pour que les deux yeux puissent le voir après les filtres. Sinon l'information ne circulera que dans un œil, et l'on perdra la vision stéréoscopique, et donc la notion de profondeur.



FIGURE 2.7 – Rendu final d'une image par l'algorithme de Dubois ¹¹

3 Sujet

Le projet concerne la réalisation d'un logiciel permettant d'obtenir des projections en deux dimensions, des anaglyphes, des autostéréogrammes ou encore des flipbooks à partir de scènes virtuelles en trois dimensions.

L'objectif premier est de permettre la visualisation, sur un support en deux dimensions tel qu'un écran d'ordinateur ou une feuille de papier, d'un espace en trois dimensions. A partir de la visualisation d'objets 3D dans une scène il faudra donc réaliser des photographies qui une fois traitées donneront lieu à des anaglyphes, autostéréogrammes ou des animations type flipbook.

Afin d'atteindre cet objectif, un logiciel s'appuyant sur le moteur 3D OpenGL devra être réalisé. Il permettra la création d'une scène ou s'inséreront des objets dont la position, la taille et l'orientation seront paramétrables. Une caméra permettra de se déplacer dans la scène, de s'en rapprocher ou s'en éloigner.

Une fois la scène mise en place, il faudra pouvoir prendre des photographies de celle-ci sous différents angles afin d'obtenir, après application d'algorithmes de traitement d'images :

- des anaglyphes rouge-cyan, qui permettront une visualisation en trois dimensions grâce à des lunettes adaptées ;

11. <http://zour.deviantart.com/art/Wernigerode-Boulevard-Dubois-Anaglyph-HDR-3D-276542278>

- des stéréogrammes, qui sont des images dissimulant un contenu qui apparaît quand on fixe le dessin de façon spécifique ;
- des flipbooks ou images animées, correspondant à une succession d'images suivant une trajectoire qui permettent en les faisant défiler de donner une impression de mouvement.

Cahier des besoins

4 Besoins fonctionnels

4.1 La scène

Cette partie montre l'ensemble des cas d'utilisation relatifs à la scène.

Création de la scène :

Au démarrage du logiciel, il sera possible de créer une nouvelle scène ou d'en charger une préexistante. Celle-ci permettra d'afficher et disposer plusieurs objets 3D avant d'être sauvegardée pour une utilisation ultérieure.

Sauvegarde de la scène :

On distinguera deux types de sauvegardes indépendantes entre elles : la sauvegarde automatique et la sauvegarde manuelle. La première s'effectuera après chaque modification apportée avec pour but d'éviter la perte de données en cas de crash du logiciel, elle se fera de manière discrète (c'est à dire en arrière-plan sans figer le logiciel le temps de la sauvegarde). La seconde se fera au format XML (dont le prototype est détaillé ultérieurement) sur demande de l'utilisateur. Il sera possible de choisir entre une importation des objets utilisés dans le répertoire courant de la scène ou d'une simple sauvegarde des chemins d'accès aux modèles 3D. Par ailleurs, les deux types de sauvegarde étant indépendants, une sauvegarde manuelle n'écrasera pas une sauvegarde automatique.

Chargement de la scène :

Comme vu précédemment, lors de la sauvegarde, les données relatives aux objets de la scène seront stockées dans un fichier au format XML. A l'ouverture du logiciel, l'utilisateur pourra choisir d'ouvrir une scène qu'il aura sauvegardée précédemment. Dans ce cas, les données enregistrées seront récupérées pour recréer la scène telle qu'elle était avant la fermeture du logiciel.

Suppression de la scène :

Le logiciel ne proposera pas directement de méthode pour supprimer une scène. Pour cela, il faudra directement aller supprimer le dossier associé à la sauvegarde à l'intérieur du dossier contenant le logiciel.

4.2 Les objets

Une fois la scène créée ou chargée, l'utilisateur aura la possibilité d'y placer des objets.

Chargement des fichiers objets :

Les objets en trois dimensions utilisés seront sous la forme de fichiers objets. Trois types seront acceptés : les fichiers d'extension .OBJ (version 3.0, ASCII) et .PLY (versions 1.0, ASCII et binaire). Ils devront permettre de définir des objets à facettes triangulaires. Les objets pourront être ceux de l'utilisateur ou de la bibliothèque implémentée dans le logiciel.

Ces objets ne devront contenir aucun trou, par exemple une face absente sur un cube ou un triangle non généré par le fichier de chargement. Le fonctionnement du logiciel n'est garanti que sur des objets topologiquement valides.

Si un fichier à charger possède trop de polygones, un algorithme de décimation de faces permet d'en réduire le nombre jusqu'à un certain seuil (cf. prototype). Ce traitement n'est pas automatique et l'utilisateur pourra ou non l'appliquer.

Placement d'un objet :

Après le chargement de l'objet il sera possible de le placer dans la scène via les trois translations du repère 3D. Ce placement n'est pas définitif et pourra être modifié ultérieurement lors de la constitution de la scène.

Modification d'un objet :

Trois caractéristiques d'un objet pourront être modifiées : son emplacement, son orientation et sa taille.

Modification de l'orientation d'un objet :

L'orientation d'un objet dans la scène pourra être modifiée en fonction des trois axes de rotation usuels.

Modification de la taille d'un objet :

L'objet pourra être agrandi ou réduit selon le besoin.

Mode de modification d'un objet :

Le placement et le paramétrage de l'objet (orientation, taille ...) n'est pas définitif et peut-être revu par la suite via un mode de modification après avoir sélectionné l'objet à modifier. Toute sortie de ce mode de modification entraînera une sauvegarde automatique de la scène.

Sélection d'un objet :

Il sera possible de sélectionner un objet pour réaliser des modifications.

Suppression d'un objet :

Il sera également possible, après sélection, de supprimer un ou plusieurs objets de la scène. Le ou les objets ainsi supprimés ne seront plus référencés dans la sauvegarde.

4.3 La caméra

La caméra possède un repère qui lui est propre et une distance correspondant à celle entre l'origine du repère de la scène (centre de rotation de la caméra) et le sien.

Observation de la scène :

La caméra permettra de se déplacer dans la scène en suivant les trois axes de rotation et les trois axes de translation usuels, ainsi qu'un zoom. Si l'on choisit de se déplacer relativement aux axes de translation, le centre de rotation de la caméra sera modifié.

Type de projection :

Deux types de projections seront disponibles : la projection orthographique (qui sera l'option par défaut) et la projection en perspective.

Zoom de la scène :

Il sera possible de se rapprocher ou s'éloigner de la scène via une fonctionnalité de zoom. Il ne sera pas possible d'atteindre le centre de rotation de la caméra.

Remise à zéro du centre de la caméra :

La position de la caméra pourra, à tout moment, être réinitialisée pour se retrouver dans les conditions initiales de visionnage de la scène.

4.4 Le passage en deux dimensions

Mode de passage en deux dimensions :

Différents rendus seront accessibles :

- Photographie
- Anaglyphe via un algorithme améliorant le rendu post-impression [5] ou s'appliquant sur des paires d'images pseudo stéréoscopique [6].

Pour les images générées par les modes présentés ci-dessous, il faudra choisir entre une qualité de 75 ou 300 pixels par pouce. Préciser les algorithmes

Mode photographie :

Le mode photographie permettra d'obtenir une image proche de la scène virtuelle créée. Pour ce mode, des informations concernant la qualité de l'image et sa taille devront être fournies.

Mode anaglyphe :

Le mode anaglyphe permettra d'obtenir une image rouge, une image cyan, ainsi qu'une image superposant les deux. Pour ce mode, des informations concernant la qualité de l'image et sa taille devront être fournies.

Mode autostéréogramme :

Le mode autostéréogramme permettra d'obtenir un autostéréogramme fixe à une image. Pour ce mode, des informations concernant la qualité de l'image et sa taille devront être fournies.

Mode flipbook :

Le mode flipbook permettra d'obtenir un ensemble d'image sur une trajectoire donnée. Pour ce mode, des informations concernant la qualité de l'image et sa taille, mais également des informations nécessaires au lancement et à la construction du flipbook devront être fournies par l'utilisateur. Ainsi, il devra tout d'abord choisir une trajectoire parmi les trois axes de rotation, les trois axes de translation ou le zoom, puis un point de départ. Au choix, il pourra alors sélectionner un point d'arrivée et un nombre d'images, ou un nombre d'image et un angle pour connaître la mesure entre deux prises. Une barre de progression indiquera le nombre prises de vues faites par rapport au total. Si le temps d'attente est trop long, il sera toujours possible d'annuler l'action et redéfinir les paramètres.

Affichage du rendu :

Les algorithmes des modes présentés ci-dessus permettront d'obtenir un rendu correspondant aux attentes de l'utilisateur. Ce dernier sera ensuite affiché dans une nouvelle fenêtre contenant le ou les produits demandé qu'il sera possible de sauvegarder avant de quitter.

Enregistrement du rendu :

Si l'utilisateur choisit d'enregistrer les images générées par les algorithmes, les rendus qu'il souhaite sauvegarder devront être choisis parmi les rendus obtenus.

Enregistrement de la photographie :

Pour le cas simple de la photographie, il faudra nommer le fichier et donner son emplacement avant qu'il ne soit enregistré au format PNG.

Enregistrement de l'anaglyphe :

Les images rouge et cyan pourront être enregistrées séparément ou non. Deux options de rendus seront disponibles : pour l'impression et pour l'ordinateur. Elles permettront d'optimiser l'affichage et l'utilisation des lunettes dans les deux cas.

Enregistrement de l'autostéréogramme :

Pour l'autostéréogramme, il faudra nommer le fichier et indiquer son emplacement avant de le sauvegarder au format PNG.

Enregistrement du flipbook :

Dans le cas du flipbook, les images pourront être enregistrées une à une au format PNG, par tranche au format PNG ou sous forme d'animation GIF. Dans le cas d'une sauvegarde séparée de chaque image il faudra les nommer et donner leur chemin d'accès. Par tranche il faudra indiquer la première et la dernière image ainsi qu'un nom commun et un répertoire. Pour l'animation il faudra la nommer et donner son chemin d'accès.

5 Besoins non fonctionnels

5.1 La fluidité d’affichage

Pour que l’affichage soit fluide la machine devra avoir au minimum 2Go de mémoire vive, un processeur récent avec au moins 2 cœurs cadencés au minimum à 2GHz et les conditions suivantes devront être validées :

- Pour une machine avec processeur graphique type Intel HD Graphics 4000 ou plus récent, au maximum 100 000 points pour un affichage à 25 fps, ou 150 000 points pour un affichage à 20 fps ;
- Pour une machine avec une carte graphique type NVIDIA GTX 720 ou plus récente, au maximum 350 000 points pour un affichage à 25 fps, ou 500 000 points pour un affichage à 20 fps.

Test de validation : ouverture de plusieurs modèles 3D, présentés en annexe, dans Meshlab sur différentes machines. Les résultats sont également disponibles en annexe.

5.2 Fluidité d’obtention des rendus

A l’heure actuelle on ne peut garantir un nombre de points assurant l’obtention d’un rendu en un temps raisonnable. C’est pourquoi une barre de progression s’affichera pour que l’utilisateur ait une idée du temps d’attente avant l’affichage du résultat.

5.3 Portabilité du logiciel

La portabilité rend le logiciel accessible à un plus grand nombre de personnes. Les systèmes d’exploitation visés seront Windows XP, Windows Seven, Windows 8.1, Ubuntu 14.04, Fedora 20 et Debian Jessie, en plateforme 32 bits et 64 bits.

Test de validation : un prototype, qui pourra être utilisé comme base du projet, a été exécuté sur l’ensemble de ces systèmes d’exploitation (cf. la partie 9 Prototype test).

5.4 Extensibilité du logiciel

Le client souhaite que l’extensibilité du logiciel soit facilitée par son architecture. En effet, il envisage par la suite d’ajouter de nouvelles fonctionnalités ou de nouveaux algorithmes au logiciel, et demande à ce que de telles modifications soient aisément envisageables.

La facilitation de cette extensibilité sera permise par l’utilisation d’une architecture spécifique. Celle-ci est détaillée dans la partie 10 Architecture du projet de ce cahier des charges.

6 Contraintes

6.1 Superposition des objets dans la scène

Pour faciliter le traitement d’une scène, on considèrera que deux objets placés dans une scène ne pourront pas être superposés. On parle ici d’intersection des boîtes englobant des objets et non de la gestion des parties non visibles des objets lors de l’affichage.

6.2 Utilisation des bibliothèques

Afin de minimiser les dépendances et valider le besoin en portabilité les bibliothèques Qt et OpenGL seront utilisées. CMake permettra de générer les scripts de compilation quel que soit la plateforme.

6.3 Langage de programmation

Le langage C++ sera utilisé.

6.4 Open Source

Pour permettre l'utilisation du logiciel et son extensibilité à quiconque souhaiterait le modifier, on choisira de laisser le code du projet en Open Source. Pour garantir cela, on s'assura bien que l'on dispose des droits nécessaires pour n'importe quel algorithme ou fragment de code utilisé au cours de ce projet.

7 Briques logicielles

7.1 Meshlab

En développement depuis 2005, Meshlab est un logiciel libre de traitement de maillage 3D disponible sur Microsoft Windows, Mac OS X, GNU / Linux et ses dérivés. Il propose différents outils comme des filtres de nettoyage du maillage, la possibilité de supprimer le bruit etc. C'est un logiciel très apprécié dans le domaine universitaire pour la reconstruction de surface, l'impression 3D etc. car il s'agit d'un logiciel non seulement gratuit, mais également Open Source. Cela permet d'avoir une base commune et grâce à une communauté active, de nouvelles fonctionnalités peuvent être ajoutées et utilisées par tout le monde.

7.2 Blender

Blender est, depuis 2003, un logiciel libre de modélisation, d'animation et de rendu 3D disponible sur de nombreuses plateformes comme Microsoft Windows, Mac OS X, GNU / Linux et ses dérivés. C'est un logiciel extensible, léger, ne réclamant pas un ordinateur très puissant pour des actions simples. Sa communauté de fan très développée a permis la réalisation d'une base de données de greffons conséquente. Preuve de sa puissance, celui-ci a été utilisé pour réaliser entièrement un film d'animations (Tears of Steel) et pour les effets spéciaux de plusieurs films tels que Mr. Nobody.

8 Maquette

9 Prototype test

10 Architecture du projet

Références

- [1] Steer W. A. Stereograms : Technical details. <http://www.techmind.org/stereo/stech.html>, Juin 2002.
- [2] Eric Dubois. A projection method to generate anaglyph stereo images. *IEEE International Conference*, 3 :1661–1664, 2001.
- [3] Beene G. Stereogram algorithms – basics. <http://www.garybeene.com/stereo/rds-alg-basic.htm>, 2006.
- [4] Stuart Inglis, Harold W. Thimbleby, and Ian H. Witten. Displaying 3d images : Algorithms for single image random dot stereograms. *IEEE Computer*, 27 :38–48, 1994.
- [5] Elena Patana, Ilia Safonov, and Michael Rychagov. Adaptive 3d color anaglyph generation for printing. *GraphiCon'2012 Conference*, Conference Proceedings :55–60, 2012.

- [6] Alkhadour W, Ipson S, Zraqou J, Qahwaji R, and Haigh J. Creating a color anaglyph from a pseudo-stereo pair of images. *4th ICIT*, pages –, 2009.