

Rapport final
Intégration de vidéos dans un environnement tridimensionnel

Rapport présenté à
Jean-François Lapointe
Michal Iglewski

Par
Marc-André Ménard
Julien Tremblay

Dans le cadre du cours
Projet Synthèse
Groupe 01

À remettre le 17 décembre 2007
Université du Québec en Outaouais

1. Rappel du Projet	2
1.1 Description.....	2
1.2 Objectif du projet	2
1.3.1 Élaboration des spécifications.	3
1.3.2. Élaboration d'un plan des tâches.....	4
1.3.3. Déterminer le langage de programmation et l'API à utiliser	4
1.3.4. Développement de l'application	5
1.3.5. Test de l'application	6
1.3.6. Rédaction de la documentation	7
1.4 Échéancier révisé	8
1.5. Appréciation générale	8
2. Recherches effectuées.....	9
2.1 OpenGL / CSDL.....	9
2.1.1 Description.....	9
2.1.2 Programmation avec CSDL.....	9
2.2 DirectX.....	10
2.2.1 Description.....	10
2.2.2 Programmation avec AudioVideoPlayback.....	10
2.3 DirectShow (composante Video Mixing Renderer 9)	11
2.3.1 Description.....	11
2.4 Window Media Foundation	12
3. Programme	12
3.1. Description	12
3.1.1 Fonctionnement interne de l'application	13
3.2. Guide de l'utilisateur	15
3.2.1 Lancement de l'application	15
3.2.2 Composantes requises.....	15
3.2.3 Formulaire d'entrée de vidéos.	16
3.2.4 Environnement 3D.	17
3.2.5. Appel de la ligne de commande Windows	19
4. Période de test	20
4.1. Application.....	20
4.2. Vidéo	21
4.3. Conclusion	22
5. Bibliographie.....	23
5.1 OpenGL/CSDL.....	23
5.2 DirectX/Audiovideoplayback	23
5.3 VMR-9:	23
5.4 DirectShow	23
5.5 Windows Media Foundation	23
6. Annexe.....	24
6.1. Formulaire d'entrée des vidéos	24
6.2 L'environnement.....	25
7. Glossaire	25
8. Outils utilisés.....	28
8.1 Logiciels	28
8.2 Langage de programmation.....	28
8.3 Librairies.....	28
8.3.1 OpenGL/Csdl	28
8.3.2 AudioVidéoPlayback.....	28
8.3.3 VMR9	28
8.4 Codecs	28

1. Rappel du Projet

1.1 Description

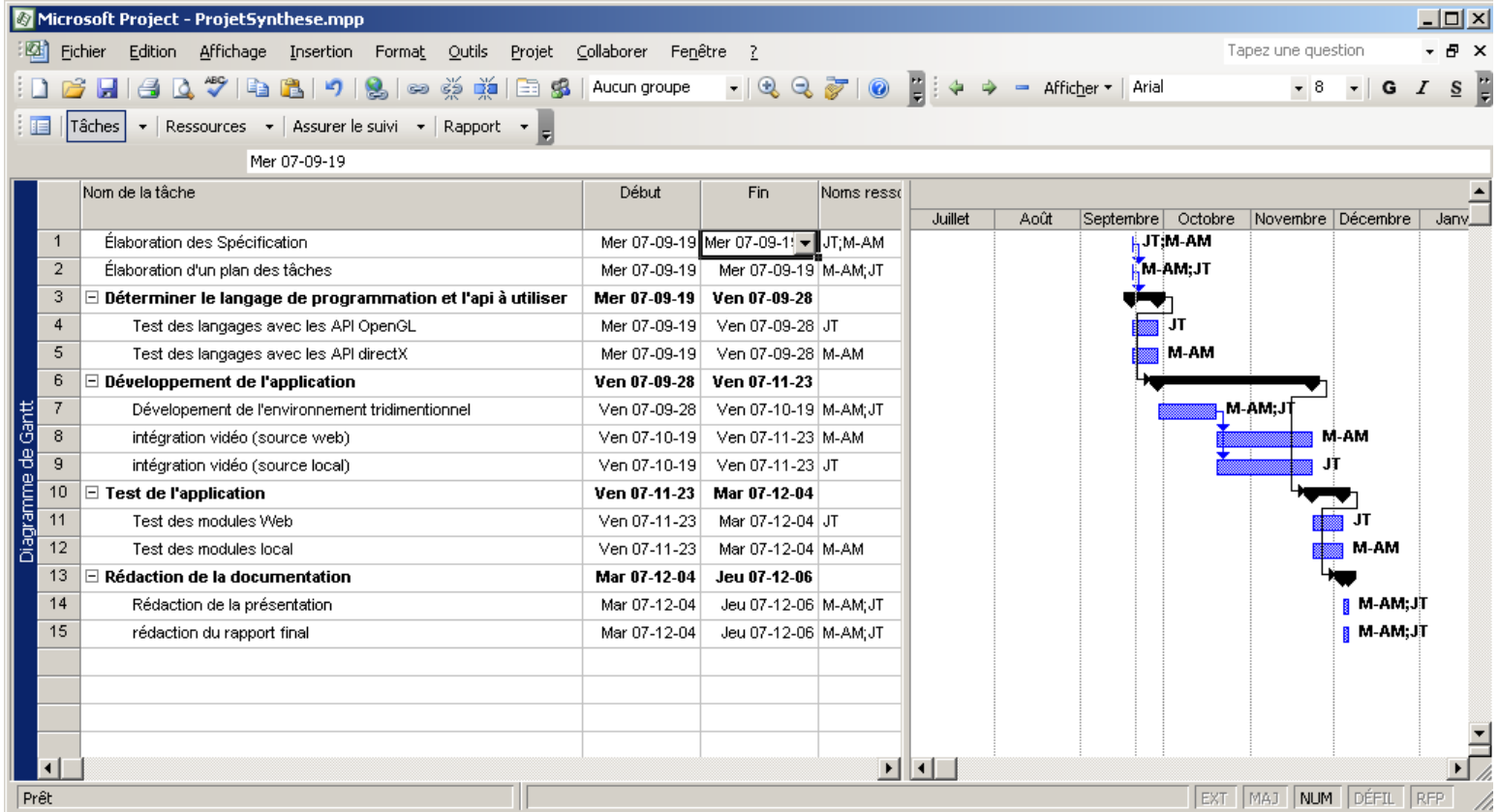
Le projet a pour but d'intégrer des vidéos de différents formats (mpg,mov, avi,wmv,etc.) et de différents codecs (MPEG-1,MPEG-4, Quicktime, etc.) dans un environnement en trois dimensions. La source des vidéos peut être local ou web. Cette application doit être polyvalente c'est-à-dire qu'elle ne doit pas nécessiter une configuration matérielle à haut rendement.

1.2 Objectif du projet

Nous espérons atteindre les objectifs suivants :

- Mettre en pratique nos connaissances en programmation
- Apprendre la programmation en 3D avec des langages de programmation de haut niveau tel C#, VB.net ou J#.
- Se familiariser avec la suite Visual Studio 2005, le infrastructure .NET 2.0 et les API reliés au 3d.

1.3. Échéancier



1.3.1 Élaboration des spécifications.

Date de début : Mercredi 19 septembre 2007

Date de fin prévue: Mercredi 19 septembre 2007

Commentaires : Cette partie n'a pas demandé beaucoup de réflexion.

1.3.2. Élaboration d'un plan des tâches

Date de début : Jeudi 20 septembre 2007

Date de fin prévue: Jeudi 20 septembre 2007

Commentaires : Cette partie n'a pas demandé beaucoup de réflexion. Cependant, il fallut effectuer certaines révisions.

1.3.3. Déterminer le langage de programmation et l'API à utiliser

Commentaires : La période de recherche fut plus longue, car à chaque fois que nous procédions à la recherche d'un API, une autre plus performante ou plus récente s'offrait à nous. Nous avons passé beaucoup plus de temps dans les recherches afin de trouver l'API qui répondait le plus aux exigences définies antérieurement.

1. Test des langages avec l'API OpenGL

Date de début : Vendredi 21 septembre 2007

Date de fin prévue: Vendredi 28 septembre 2007

Date réelle de finition : Lundi 1er octobre 2007

Commentaires : Les tests liés à l'API OpenGL se sont bien déroulés.

2. Test des langages avec l'API DirectX

Date de début : Vendredi 21 septembre 2007

Date de fin prévue: Vendredi 28 septembre 2007

Date réelle de finition : Lundi 1er octobre 2007

Commentaires : Les tests liés à l'API DirectX se sont bien déroulés.

3. Test des langages avec l'API DirectShow

Date de début : Mardi 2 octobre 2007

Date de fin prévue: Samedi 20 octobre 2007

Date réelle de finition : Mardi 30 octobre 2007

Commentaires : Les tests liés à l'API DirectShow se sont bien déroulés. Cette étape a été ajoutée pour enrichir notre recherche.

1.3.4. Développement de l'application

Commentaires : Cette partie fût mal évaluée en grande partie à cause des recherches effectuées. En effet, nous avons changé d'API à plus de trois reprises, ce qui nécessitait des modifications de codage de l'environnement et des fonctionnalités.

1. Développement de l'environnement tridimensionnel

Date de début : Vendredi 12 octobre 2007

Date de fin prévue: Dimanche 25 novembre 2007

Date réelle de finition : Lundi 26 novembre 2007

Commentaires : Nous avons fait face à des problèmes de superposition des polygones. Cela a pu être corrigé en créant un « ZBuffer ». À part ce problème, nous n'avons pas rencontré d'autres problèmes majeurs.

2. Intégration vidéo (web)

Date de début : Vendredi 12 octobre 2007

Date de fin prévue: Dimanche 25 novembre 2007

Date réelle de finition : Lundi 26 novembre 2007

Commentaires : Cette section a été remplacée par l'intégration de vidéo en réseau.

3. Intégration vidéo (local)

Date de début : Vendredi 12 octobre 2007

Date de fin prévue: Jeudi 29 novembre 2007

Date réelle de finition : Vendredi 30 novembre 2007

Commentaires : Difficultés à assimiler le fonctionnement de VMR-9 ainsi que l'allocateur. Après plusieurs heures de recherches, nous sommes parvenus à bien comprendre le fonctionnement de ses deux items.

1.3.5. Test de l'application

Commentaires : Nous avons commencé le test de l'application plus tôt que prévu. Nous avons testé l'intégration vidéo locale et par réseau plus rapidement que prévu. Les détails de cette section peuvent être trouvés au point 4.

1. Test des modules (web)

Date de début : Mercredi 31 octobre 2007

Date de fin prévue: Lundi 12 novembre 2007

Date réelle de finition : Lundi 12 novembre 2007

Commentaires : Cette section a été remplacée par test de vidéo en réseau. Le déroulement des tests a bien été. Certaines vidéos prenaient plus de temps à charger que d'autres.

2. Test des modules (local)

Date de début : Mercredi 31 octobre 2007

Date de fin prévue: Mercredi 14 novembre 2007

Date réelle de finition : Vendredi 16 novembre 2007

Commentaires : Le test des modules s'est bien déroulé. Nous avons trouvé et résolu plusieurs problèmes.

3. Test des limites de l'application.

Date de début : Mercredi 31 octobre 2007

Date de fin prévue: Mercredi 14 novembre 2007

Date réelle de finition : Jeudi 15 novembre 2007

Commentaires : Le test des modules s'est bien déroulé. Vous pouvez voir les résultats à la section 4.

1.3.6. Rédaction de la documentation

Commentaires : La rédaction des documents s'est effectuée plus rapidement que prévu. La présentation a dû être finalisée avant la date prévue pour permettre une révision par notre superviseur. De plus, le rapport final a dû être remis au moins une fois au superviseur pour révision, mais à été remis à temps.

1. Rédaction du plan des tâches

Date de début : Mercredi 19 septembre 2007

Date de fin prévue: Mercredi 19 septembre 2007

Date réelle de finition : Mercredi 19 septembre 2007

Commentaires : Après avoir élaboré le plan, la rédaction s'est faite facilement.

2. Rédaction du rapport de progrès

Date de début : Samedi 20 octobre 2007

Date de fin prévue: Mercredi 24 octobre 2007

Date réelle de finition : Mardi 23 octobre 2007

Commentaires : Le rapport de progrès fût rédigé aisément.

3. Rédaction de la présentation

Date de début : Lundi 26 novembre 2007

Date de fin prévue: Mardi 4 décembre 2007

Date réelle de finition : Lundi 3 décembre 2007

Commentaires : La présentation a été selon nous un succès.

4. Rédaction du rapport final

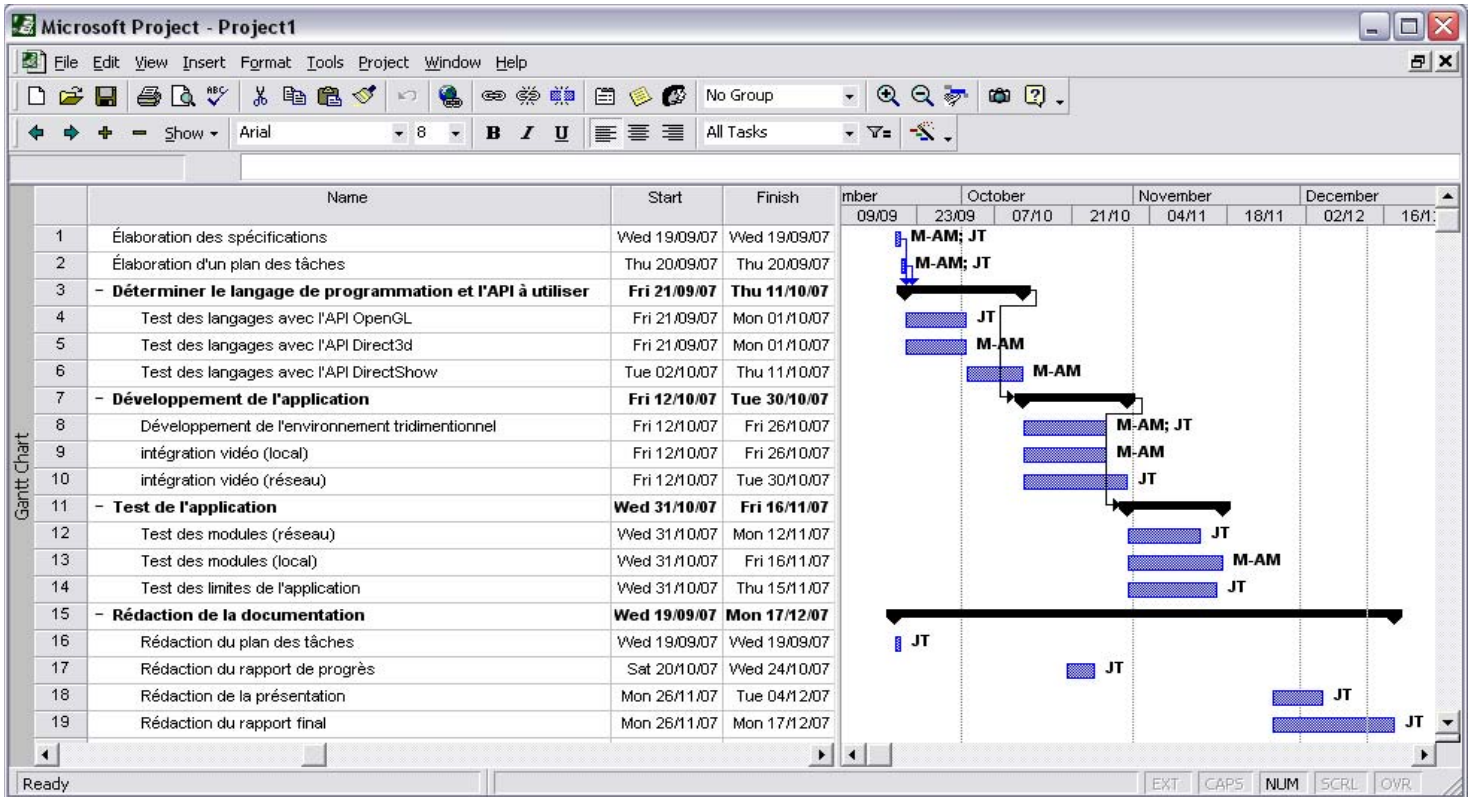
Date de début : Lundi 26 novembre 2007

Date de fin prévue: Lundi 17 décembre 2007

Date réelle de finition : Samedi 15 décembre 2007

Commentaires : La rédaction du rapport final a été difficile à entreprendre, car beaucoup d'éléments ont dû être incorporés.

1.4 Échéancier révisé



1.5. Appréciation générale

Le projet fût à nos yeux une réussite autant dans le domaine de la programmation que dans l'application des connaissances acquises à travers le baccalauréat. En effet, nous avons pu utiliser nos connaissances afin de produire les rapports, concrétiser le programme et montrer une capacité d'adaptation satisfaisante. Ce projet nous a permis d'approfondir nos connaissances en programmation reliées à la 3D et en gestion de projets.

2. Recherches effectuées

Dans les premières semaines, nous avons recherché sur les différentes API disponible : OpenGL, DirectX (composante Direct3D et AudioVideoPlayback), DirectShow (composante Video Mixing Renderer 9) et Windows Media Foundation.

2.1 OpenGL / CSGL

2.1.1 Description

OpenGL est une librairie qui est très populaire dans le langage C++ en ce qui concerne l'implémentation tridimensionnelle. Elle permet de créer des environnements tridimensionnels de haut rendement et procure beaucoup de fonctions flexibles. Cependant, l'API n'est pas adapté à 100% à l'environnement Visual Studio. Effectivement, nous avons démontré, grâce à nos recherches, que l'outil a été transposé qu'avec l'aide de programmeurs bienfaisants qui désiraient voir OpenGL dans Visual Studio. Par conséquent, la librairie n'est pas complètement fonctionnelle et ne possède pas toutes les capacités disponibles sur OpenGL en C++.

2.1.2 Programmation avec CSGL

Nous cherchons à intégrer le plus de types de vidéo possible (avi, mpeg, wmv, mov, etc.) et les recherches ont démontré que CSGL ne peut supporter que le type avi avec la fonction Avfilestreamer. De plus, cette fonction ne permet pas de jouer le son en même temps. Nous devons donc aussi trouver une façon de faire fonctionner le son en même temps que la vidéo sans perte de synchronisation.

Cependant, la librairie nous permet de coder avec aisance un environnement tridimensionnel stable et fonctionnel. Effectivement, créer les vertex et les dessiner dans l'environnement n'est pas difficile. En conclusion, nous avons décidé d'opter pour une autre librairie, car Opengl ne supporte pas facilement la vidéo et l'audio.

2.2 DirectX

2.2.1 Description

Étant un API développé par Microsoft, il est facile d'intégrer les différentes composantes (libraires) de DirectX dans l'environnement Visual Studio .NET 2005. Il faut par contre télécharger le SDK qui est d'une taille relativement importante (468 méga-octets). Grâce à la librairie AudioVideoPlack, il est facile d'intégrer des vidéos dans un environnement tridimensionnel. Construire cet environnement est relativement complexe. La composante de DirectX permettant cette tâche se nomme Direct3D. Un autre avantage se situe dans la documentation relative à DirectX disponible sur le web. Un grand nombre de tutoriaux et d'articles sont disponibles. Le principal inconvénient avec DirectX est qu'il ne peut être intégré dans des applications multi plateformes. Ainsi donc, notre application ne pourra pas fonctionner sous Linux ou Mac.

2.2.2 Programmation avec AudioVideoPlayback

La programmation avec Direct3D est moins intuitive que celle de CSGL. En effet, tout est compris en triangle. Chaque vertex doit être défini dans un groupe de trois et le nombre de vertex à dessiner doit être spécifié avant de créer l'environnement. Donc, si nous créons 5 tableaux de coordonnées de vertex, mais nous ne spécifions que 3 tableaux de coordonnées au moment de dessiner l'environnement, les deux autres triangles seront ignorés. Cependant, l'intégration vidéo est beaucoup plus simple. En effet, AudioVideoPlayback permet d'incorporer la vidéo comme texture et si la vidéo contient de l'audio, AudioVideoplayback joue celle-ci. Nous avons été capables d'incorporer les types suivants de vidéo :

- mpeg
- avi
- avi (avec divx)
- wmv

De plus, nous avons été capables d'incorporer la fonction d'arrêt, de pause et de lancement de la vidéo. Cependant, la capacité de rendement ne permet pas un rendement clair des vidéos. En effet,

lorsque nous faisons jouer plus d'une vidéo sur le polygone, l'environnement commence à clignoter sans arrêt et les vidéos se mélangent. Heureusement, nous avons découvert une alternative possible nommée Video Mixing Renderer 9 (VMR-9) qui résout ce problème.

2.3 DirectShow (composante Video Mixing Renderer 9)

2.3.1 Description

Video Mixing Renderer 9 (VMR-9) est une composante de l'API DirectShow (qui a longtemps été inclus dans l'api de DirectX). DirectShow est un API permettant de lire et capturer des vidéos et de l'audio dans un environnement 2D ou 3D.

VMR-9 permet de transposer une ou plusieurs vidéos sur une surface direct3D (qui est générée par l'API de DirectX) et permet également de faire différents effets spéciaux comme mixer 2 ou plusieurs vidéos ensemble avec des effets de transparences. Il offre également un support des fichiers Quicktime (version 2 et plus bas), support qu'AudioVideoPlayBack ne possède pas. Les performances de VMR-9 semblent être supérieures à AudioVideoPlayBack. En effet, en faisant jouer six vidéos simultanément de qualité équivalente dans chacun des environnements, VMR-9 semble être plus stable et la qualité visuelle des vidéos est meilleure (aucun clignotement).

La librairie de DirectShow que nous utilisons vient sous format DLL et est « Open Source ». Tout son code source est disponible en C#.

Nous pensons effectuer l'intégration de vidéos en utilisant VMR-9. Son implémentation dans une application est plus complexe qu'AudioVideoPlayback, mais à long terme nous croyons que VMR-9 offre beaucoup plus de flexibilité et possibilités dans la manipulation de vidéos. La principale difficulté que nous rencontrons présentement est la manipulation individuelle de chaque vidéo (play, stop, pause).

2.4 Window Media Foundation

Media foundation est un « infrastructure » qui remplacera DirectShow sur la plateforme Vista. La composante VMR-9 sera remplacée par EVR (Enhanced Video Renderer). D'après Microsoft, Media Foundation sera plus performant que direct show. Cependant, ce « infrastructure » n'est pas encore complètement implémenté. Donc, nous devons encore utiliser Directshow pour la plupart des fonctions de l'environnement. Afin que nous ayons une meilleure compréhension de la programmation tridimensionnelle, nous avons décidé de rester avec la composante VMR-9 qui est encore supportée et fonctionnelle avec l'environnement XP et Vista.

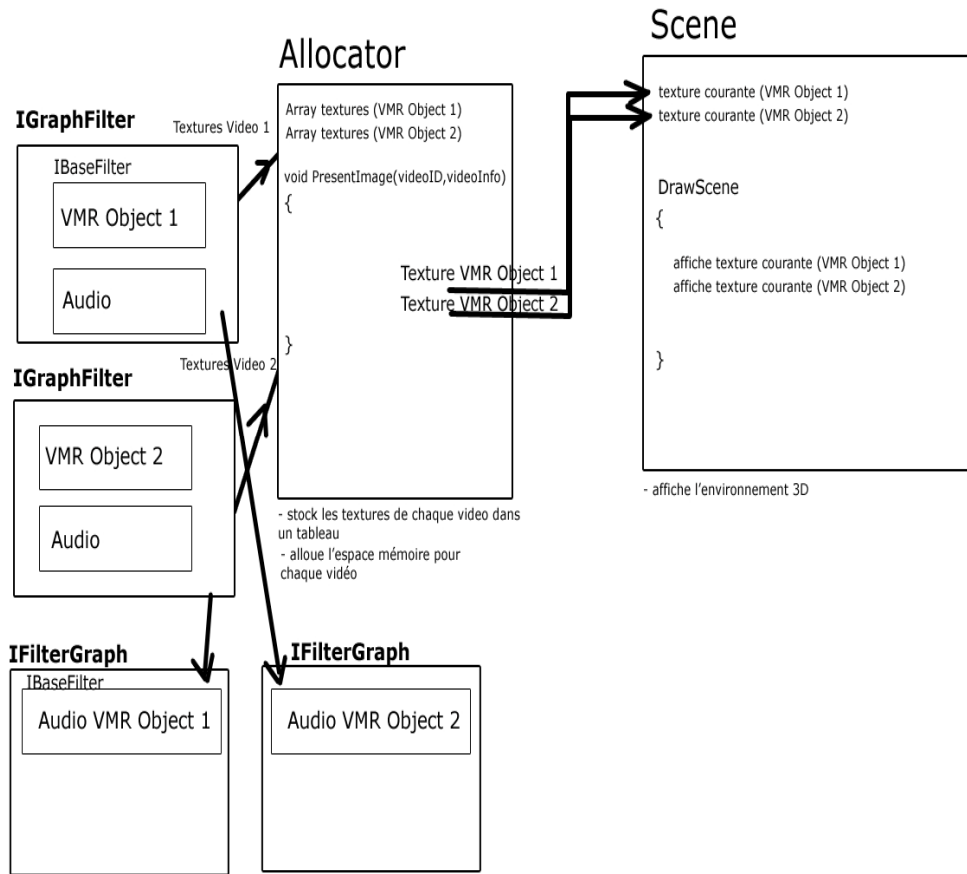
Par contre, Media Foundation remplacera Directshow complètement en temps et lieu.

3. Programme

3.1. Description

Notre application permet de jouer de 1 à 6 vidéos dans un environnement tridimensionnel. Après avoir sélectionné la ou les vidéos, l'application contrôle les vidéos indépendamment et sépare le flux audio et vidéo en deux entités différentes. Donc, si nous désirons diminuer le son d'une vidéo spécifique cela sera possible.

3.1.1 Fonctionnement interne de l'application.



Tout d'abord, un objet de type VideoMixingRenderer9 ainsi qu'un identificateur unique est créé pour chaque vidéo trouvée dans la liste (située sur la première interface visible lors de l'exécution de l'application en mode normal). Chacun de ces objets est ensuite inséré dans un tableau de type IGraphBuilder. L'étape suivante est l'initialisation de l'allocateur.

Le rôle de l'allocateur est d'extraire diverses informations de chaque vidéo trouvée dans le tableau mentionné ci haut. Ces informations sont notamment chaque image qui constitue chaque vidéo ainsi que la largeur et hauteur de ceux-ci. Toutes les images sont stockées dans un objet de type Array. L'allocateur a également le rôle d'allouer l'espace mémoire requis pour chaque vidéo. L'allocateur est finalement responsable de créer un objet de type « scene » dont sa fonction première est l'affichage des polygones et des vidéos en trois dimensions. Il est responsable de définir la résolution des vidéos qui apparaîtront sur chaque polygone. L'affichage des vidéos se produit de la façon suivante : l'allocateur appelle l'une de ses fonctions, « PresentImage », dont le rôle est de fournir à l'objet « scene » la prochaine image d'un vidéo à afficher. L'objet « scene » attrape cette image, la définit comme une texture et l'affiche sur le polygone désiré. La classe scene permet aussi de bouger, tourner et redimensionner chaque polygone. Ainsi donc, l'application est composée de deux classes clés : l'allocateur ainsi que la scène.

Une fois cette étape complétée, l'application initialise la caméra, c'est-à-dire lui donne des coordonnées par défaut. Finalement, le programme extrait le flux audio de chaque vidéo, les conserve dans un tableau de type IBasicAudio et lance l'environnement tridimensionnel. La manipulation individuelle de chaque vidéo est alors bien simple. Ayant stocké le flux vidéo et audio dans deux tableaux distincts, il est facilement possible de mettre en pause la vidéo désirée et contrôler le son de celle-ci sans problème.

3.2. Guide de l'utilisateur

Voici un document qui décrit comment utiliser notre application.

3.2.1 Lancement de l'application

Pour lancer l'application, double-cliquez sur l'icône située dans le dossier VMR9- Projet Final/Application/v5.exe. Ceci démarrera l'application en mode ajout de vidéo. Afin de partir l'application en mode ligne de commande, vous devez lancer une ligne de commande de cette manière : cliquer sur démarrer -> exécuter -> « cmd ».

3.2.2 Composantes requises

Afin que l'application soit fonctionnelle assurez-vous d'avoir les composantes suivantes installées :

- Microsoft Infrastructure 2.0 ou version supérieure.
- Direct X version 9
- librairie DirectShow (DirectShowLib-2005.dll)
- Microsoft.DirectX.dll
- Microsoft.DirectX.DirectSound.dll
- Microsoft.DirectX.DirectInput.dll
- Microsoft.DirectX.Direct3DX.dll
- Microsoft.DirectX.Direct3D.dll

Veillez noter que vous devez avoir les codecs nécessaires afin de jouer les vidéos désirées. Par exemple, si divx n'est pas installé sur votre plateforme, vous ne pourrez voir la vidéo.

3.2.3 Formulaire d'entrée de vidéos.

(Voir tableau 1.1 dans l'annexe)

L'application supporte jusqu'à 6 vidéos et ce formulaire permet d'entrer ceux-ci.

Menu

Fichier

Cette option du menu permet de sortir de l'application.

Aide

Cette option affiche l'information relative au programme.

Bouton

Ajouter Vidéo

Vous devez utiliser ce bouton afin d'ajouter des vidéos. Il ouvrira une fenêtre qui vous permettra de naviguer dans vos dossiers afin de choisir les vidéos désirées. De plus, si vous avez une connexion réseau vous pouvez sélectionner des ordinateurs à l'intérieur du réseau. L'application vous permet de choisir les vidéos avec les extensions suivantes : Avi, Mpeg, Wmv, Mov, Asf, Vob, Qt. Après avoir sélectionné la vidéo désirée, celle-ci sera ajoutée à la liste. La liste n'acceptera pas plus de 6 vidéos.

Retirer

Vous pouvez enlever des vidéos de la sélection. Vous devez sélectionner la vidéo à enlever et ensuite appuyer sur le bouton « retirer ». La vidéo sera alors retirée de la liste.

Démarrer l'environnement 3D

Appuyer sur ce bouton démarrera l'environnement 3d et vous amènera dans le deuxième formulaire de l'application.

3.2.4 Environnement 3D.

Ce formulaire contient l'environnement 3d généré après avoir choisi jusqu'à 6 vidéos. Voici les contrôles localisés sur le formulaire.

Menu

Fichier :

Cette option du menu permet de quitter l'application.

Aide :

Cette option affiche l'information relative au programme.

Section Configuration :

Fermer l'environnement 3D

Ce bouton permet de retourner au formulaire d'entrée de vidéos.

Section Positions prédéfinie des vidéos :

Panoramique

Ce bouton permet de disposer les vidéos de manière à ce qu'ils entourent l'utilisateur. Ce bouton ne fonctionnera que si l'utilisateur entre 6 vidéos.

Par défaut

Ce bouton permet de disposer les vidéos et la caméra à leur position initiale. Ceci est pratique si vous êtes perdu dans l'environnement.

Section Position de la vidéo :

« <- (x axis) »

Ce bouton permet de bouger une vidéo sur l'axe des X (décrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« <- (y axis) »

Ce bouton permet de bouger une vidéo sur l'axe des Y (incrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« <- (z axis) »

Ce bouton permet de bouger une vidéo sur l'axe des Z (décrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« (x axis) -> »

Ce bouton permet de bouger une vidéo sur l'axe des X (incrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« (y axis) -> »

Ce bouton permet de bouger une vidéo sur l'axe des Y (incrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« (z axis) -> »

Ce bouton permet de bouger une vidéo sur l'axe des Z (incrémente cette valeur). Afin d'accomplir ceci, vous devez sélectionner la vidéo désirée dans la liste (voir annexe). Ensuite, appuyer sur le bouton.

« Rotation 90 degrés »

Après avoir sélectionné la vidéo désirée à partir de la liste. Appuyer sur ce bouton afin d'effectuer une rotation de 90 degrés de la vidéo.

Son de la vidéo

Balance

Après avoir sélectionnée la vidéo désirée à partir de la liste dirigez la barre de défilement afin de diriger le son, à gauche pour le haut parleur de gauche et à droite pour le haut parleur de droite.

Volume

Après avoir sélectionné la vidéo désirée à partir de la liste, dirigez la barre de défilement afin de contrôler le volume de la vidéo.

Mute

Cette option permet d'arrêter le son de la vidéo sélectionnée préalablement dans la liste.

3.2.5. Appel de la ligne de commande Windows

Notre application peut aussi se faire appeler depuis la console Windows. Cette fonction ne permet pas le contrôle de multiples vidéos. Elle permet de faire jouer une seule vidéo dans une fenêtre. Voici les appels possibles :

- projetSynthese.exe chemin d'accès du fichier

Description : Cet appel permet d'ouvrir une vidéo spécifiée. Le nom du fichier doit être valide.

- projetSynthese.exe chemin d'accès du fichier positionx
positiony

Description : Cet appel permet d'ouvrir une vidéo spécifiée à l'endroit spécifié par les coordonnées X et Y (coin gauche supérieur de la fenêtre) fourni par l'utilisateur. Le nom du fichier doit être valide et X ainsi que Y doivent être numériques.

- projetSynthese.exe chemin d'accès du fichier positionx
positiony ratio

Description : Cet appel permet d'ouvrir une vidéo spécifiée, à l'endroit spécifié par les coordonnées X et Y (coin gauche supérieur de la fenêtre) fourni par l'utilisateur, avec le ratio spécifié par l'utilisateur. Le nom du fichier doit être valide, X ainsi que Y doivent être numérique et ratio doit être numérique (un « float » est accepté).

- projetSynthese.exe chemin d'accès du fichier -f

Description : Cet appel permet d'ouvrir la vidéo spécifiée en mode plein écran.

4. Période de test.

4.1. Application

Nous avons testé l'application sur les deux plateformes les plus populaires de Microsoft : Windows Vista et Windows XP. Voici nos résultats.

Windows Vista

L'environnement semble donner un bon rendement. Cependant, après un certain temps, les vidéos semblent se multiplier à l'intérieur de l'environnement. Nous pensons que ceci pourrait être causé par la carte graphique et des problèmes de compatibilité avec Vista. De plus, l'application arrête de fonctionner quand l'application tente d'ouvrir une vidéo et qu'elle ne possède pas le codec. Ceci n'est pas vu sur la plateforme XP.

Windows XP

La plateforme XP semble plus stable. En effet, les vidéos ne montrent pas l'effet de multiplication vu dans Vista et l'environnement de s'arrête pas si nous n'avons pas le codec requis. Cependant, le son d'une vidéo de format .mov n'est pas disponible sur cette plateforme. Nous pensons que ce problème est peut-être relié à la version du codec nécessaire pour jouer le format mov.

4.2. Vidéo

Nous avons testé l'application avec des types vidéo différents. Voici l'information sur les machines testées ainsi que les résultats reçus.

Plateforme : Vista

Mémoire vive : 1giga-octet

Processeur : AMD Turion 64.

Mémoire de la carte vidéo : 256 Mo

Modèle : Inspiron 1501

Format	Codec	Fonctionne	Commentaire
MPEG1	MPEG – 1	Oui	
MPEG4	MPEG -4	Oui	
Avi	MPEG-4	Oui	
AVI (Divx 6)	DIVX 6.7	Oui	doit avoir le codec
WMV		Oui	certain ne fonctionne pas (licence requise)
	Quicktime		
QuickTime	2.0	Oui	seulement version 2.0 et plus bas.
RealVideo	Realvideo	Non	
ASF		Oui	
	Macromedia		
FLV	Flash 8	Non	
	MPEG -2		doit avoir le codec, mais n'affiche pas dans l'environnement
MPEG2		Oui	

Plateforme : XP

Mémoire vive : 2 Go

Processeur : intel duo core

Mémoire de la carte vidéo : 256 Mo

Modèle : Inspiron 9400

Format	Codec	Fonctionne	Commentaire
MPEG	MPEG – 1	Oui	
MPEG	MPEG -4	Oui	
Avi	MPEG4	Oui	
AVI	DIVX 6.7	Oui	Dois avoir le codec
WMV	Wmv	Oui	certain ne fonctionne pas (licence requise)
	Quicktime		
Mov	2.0	Oui	seulement version 2.0 et plus bas. aucun son
RAM	Realvideo	Non	
ASF	Asf	Oui	
	Macromedia		
FLV	Flash 8	Non	
	MPEG -2		Dois avoir le codec, mais n'affiche pas dans l'environnement
MPEG		Non	

Voici la résolution des vidéos utilisées lors de la démonstration.

Nom du fichier	Largeur (en pixel)	Hauteur (en pixel)
MPEG1.mpeg	320	240
MPEG4.mpg	352	263
Surfbus3.wmv	320	240
Niceday.asf	352	288
DIVX.avi	352	240
COBTV_part1.mov	640	480

4.3. Conclusion

Test de l'application

Bien que Windows XP est une plateforme qui disparaîtra pour laisser place à Vista, il est clair que certains problèmes reliés à la compatibilité sont encore présents dans Windows Vista. Windows XP a démontré une meilleure performance en ce qui a trait au roulement de l'environnement. Cependant, il faut prendre en compte, que Windows XP est actif depuis beaucoup plus longtemps que Vista. Les problèmes de performances sous Vista seront probablement réglés dans le futur.

N b : Nous avons découvert que toutes les applications développées en Visual Studio .NET 2005 nécessitent le .NET frame work version 2 ou supérieur.

Test vidéo

En fin de compte, la plateforme ne change rien à la manière dont les vidéos se font lire. Tant que l'utilisateur possède le bon codec, notre application permet de lire la vidéo.

5. Bibliographie

5.1 OpenGL/CSgl

- <http://csgl.sourceforge.net/>
- <http://www.taoinfrastructure.com/>
- <http://nehe.gamedev.net/>
- <http://msdn2.microsoft.com/en-us/library/ms706736.aspx>

5.2 DirectX/Audiovideoplayback

- <http://morpheus.developpez.com/directxdotnet/#LIID>
- <http://www.riemers.net/Forum/index.php?var=427&var2=0>
- <http://www.gamedev.net/reference/articles/article1020.asp>
- <http://www.pluralsight.com/wiki/default.aspx/Craig.DirectX.FR/Direct3DTutorialIndex.html>
- <http://funkydata.developpez.com/csharp/directx/primitives/>
- <http://funkydata.developpez.com/csharp/directx/presentation/>
- <http://msdn2.microsoft.com/fr-fr/library/1761b12c-6429-4c36-838b-8638ff499354.aspx>
- <http://www.riemers.net/eng/Tutorials/DirectX/Csharp/Series1/tut4.php>
- <http://msdn2.microsoft.com/fr-fr/library/cbbf9c88-82d5-46ab-9024-3f61af486185.aspx>
- <http://odelmotte.developpez.com/tutoriels/manageddirectx/audiovideoplayback/csharp/#L3>

5.3 VMR-9:

- http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/directx9_c_summer_03/directx/htm/usingthevideomixingrenderer.asp

5.4 DirectShow

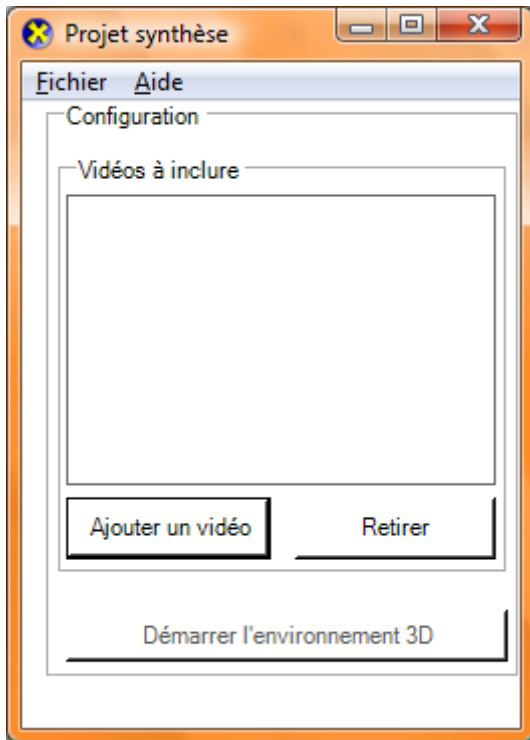
- <http://directshownet.sourceforge.net/about.html> (bibliothèque que nous utilisons)
- <http://msdn2.microsoft.com/en-us/library/ms786508.aspx>

5.5 Windows Media Foundation

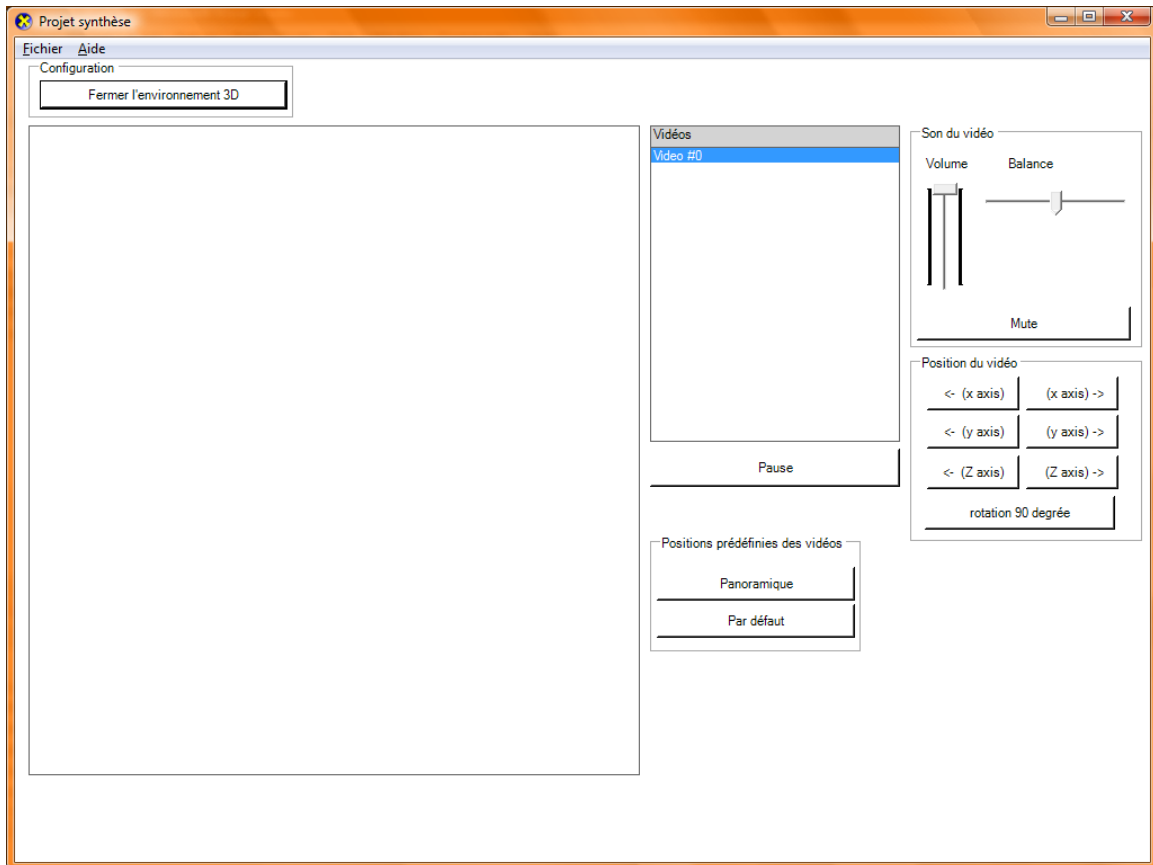
- http://en.wikipedia.org/wiki/Media_Foundation
- <http://msdn2.microsoft.com/en-us/library/ms694197.aspx>
- <http://forums.poweruser.tv/index.aspx?forumid=344&aid=86654&c=1>

6. Annexe

6.1. Formulaire d'entrée des vidéos



6.2 L'environnement



7. Glossaire

Format

Un format définit comment le codec est affiché sur l'écran.

WMV

Windows Media Video

Ceci est un format fermé qui ne peut jouer que les vidéos générées. Ce format est généré permet de jouer les asf,wmv etc.

ASF

Advanced System Format. Utilisé par Microsoft pour jouer des vidéos sur le web. N'as pas de spécification en ce qui attrait l sorte de codec nécessaire afin de jouer

FLV

Flash Video. Format vidéo utilisé pour les vidéos créées par l'application Adobe Flash Player.

MOV

Format créé par Quicktime afin de supporter des vidéos de multi piste. Chaque piste contient un élément du vidéo : audio, vidéo, etc....

MPEG

Format vidéo qui tient son nom du groupe qui le développe. Le **Moving Picture Experts Group**. Le format supporte tous les codecs créés par MPEG.

AVI

Audio Video interleave. Format vidéo utilisé fréquemment. Ce format sépare l'information en trois sections afin de mieux traiter le flux de donnée.

RAM(Realvideo)

Format vidéo créé par RealNetwork ans les années 1997. Il est utilisé plus souvent dans la visualisation de vidéo sur interne ou pour visualiser la télévision live.

Codec

Logiciel ou programme utilisé pour coder et décoder des signaux digitaux de données.

Divx

Codec dérivé de MPEG4, il compresse les formats vidéo de grande taille et les compresse en plusieurs petits morceaux tout en gardant une bonne qualité.

MPEG1

Ce codec a été inventé afin de développer une manière d'intégrer des vidéos à l'intérieur d'un cd-rom. Créée par la compagnie Moving Picture Experts Group.

MPEG2

Ce codec est utilisé en grande partie pour compresser les formats vidéo en format dvd ou télévision haute définition.

MPEG4

Standard utilisé pour compresser les données digitales. Introduit dans les débuts des années 1998. Ses utilisations inclues le web, la distribution cd, vidéoconférence et la télévision.

WMV

En 2003, Microsoft a élaboré un projet de codec vidéo basé sur son codec WMV 9 et l'a soumis à la norme SMPTE pour la normalisation. La norme a été approuver officiellement en mars 2006 rendant ainsi le codec WMV une norme propriétaire. Depuis, WMV est devenu l'un des trois codecs vidéo obligatoires pour les BD-ROM et HD DVD-ROM.

Realvideo(rv40)

RealNetworks a créé un codec pour supporter leur vidéo depuis realvideo version 8.

Macromedia Flash 8

Le codec créée pour flv est une variante h.263 qui utilise une suite de capture écran.

VMR 9

Video Mixing Renderer 9. Composante de DirectShow. Permet d'intégrer la vidéo dans un environnement 3D. Plus performant qu'AudioVideoPlayback.

CSGL

C Sharp Graphic Library. Cette composante est ce que certains programmeurs ont créé afin de rendre OpenGL disponible sur la plateforme de Windows Visual Studio 2005.

8. Outils utilisés

8.1 Logiciels

- Codecviewer 1.4.1: utilisé pour trouver le codec des formats avi.
- Microsoft Project 2003 : utilisé pour préparer notre échéancier.
- Microsoft Office 2003 : utilisé pour écrire nos rapports et notre présentation.
- Microsoft Visual Studio 2005 : Utiliser pour programmer l'application même.

8.2 Langage de programmation

- C# version visual studio 2005

8.3 Librairies

8.3.1 OpenGL/Csgl

- csgl.dll version : 1.4.11
- TaoInfrastructure version: 2.0

8.3.2 AudioVidéoPlayback

- SDK DirectX version: 9

8.3.3 VMR9

- DirectShow2005.dll version: 2.0.0.0

8.4 Codecs

- Divx version 5.6
- MPEG version: 1
- MPEG version: 2
- MPEG version: 3
- WMV version: 9
- ASF version: 1.20.03
- Quicktime version: 2.0
- Macromedia Flash version: 8
- RealVideo version: rv40

