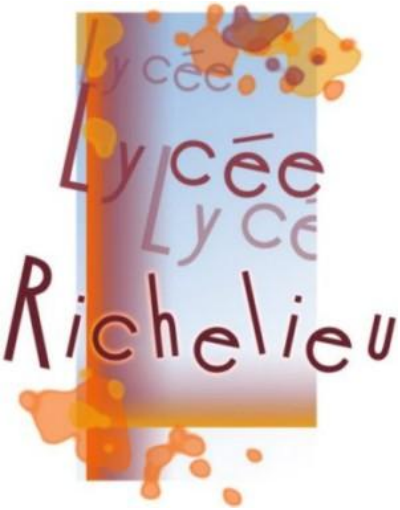
Richelieu3D

Une visite virtuelle du CDI du nouveau lycée Richelieu



Travail Personnel Encadré

Problématique: Comment réaliser la visite virtuelle d'un lieu ?
Objet d'étude: CDI du futur lycée Richelieu
Année: 2009-2010

Louis Lagrange	1ère S8
Yugo Hirata	1ère S4
Louis Berthelon	1ère S4

Sommaire

I/ Introduction

A) Présentation

Notre projet de TPE s'intitule « Richelieu3D ». Pourquoi l'appeler comme cela ? Tout simplement parce que c'est un nom facile à retenir, et qu'il est assez explicite. On ne s'étonne donc pas lorsque l'on apprend que Richelieu3D est un programme informatique permettant de visiter en 3 dimensions le lycée Richelieu.

Le nom ne dit cependant pas tout : vous pouvez visiter avec ce programme le CDI du futur **nouveau** lycée Richelieu ! Nous avons en effet pensé qu'il serait utile de pouvoir se faire une idée concrète de ce que va être le nouveau CDI, qui est l'un des espaces communs de la vie lycéenne actuelle.

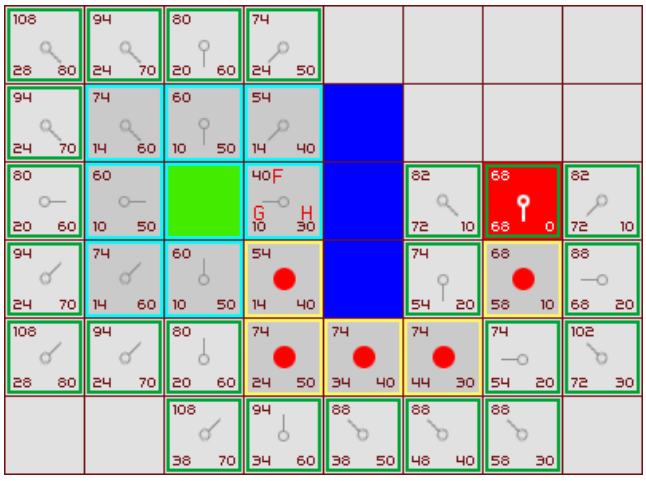
Lors d'une des premières séances de TPE, beaucoup de bruits étaient émis par les travaux. C'est en pensant à ces travaux que nous avons eu l'idée de faire une visite virtuelle du nouveau lycée Richelieu. Le sujet nous paraissant sortir de l'ordinaire et étant intéressant, nous l'avons retenu.

B) Court historique

Au départ, nous avions deux finalités possibilités:

1. Faire une visite virtuelle du nouveau lycée Richelieu entier;
2. Expliquer comment fonctionne un algorithme de *pathfinding*. (recherche de chemin)

Nous avons réalisé qu'atteindre complètement ces deux buts était impossible au vu de la quantité du travail énorme qui nous aurait attendus, que ce soit au niveau de la modélisation ou de la programmation. Il fallait donc faire **un** choix. L'étude sur le *pathfinding* se révélant moins intéressante à réaliser, et moins attractive que la visite virtuelle, nous avons finalement choisi la visite virtuelle.



L'algorithme de Pathfinding A en une image…*

Afin d'être sûrs d'avoir assez de temps, nous avons décidé de nous concentrer sur le nouveau Centre de Documentation et d'Information du lycée, ce qui représente quand même une surface de plus de 650m² !

C) Le programme

Le programme est composé de deux parties :

1) Les menus

Ils sont simples. C'est un choix voulu, nous le verrons dans la partie sur le cahier des charges fonctionnel. Les menus sont composés de deux parties, l'une étant l'écran d'accueil, l'autre servant à spécifier si l'on veut des explications concernant la position actuelle dans le CDI ou non. Il suffit d'appuyer sur un bouton pour réaliser les actions demandées.

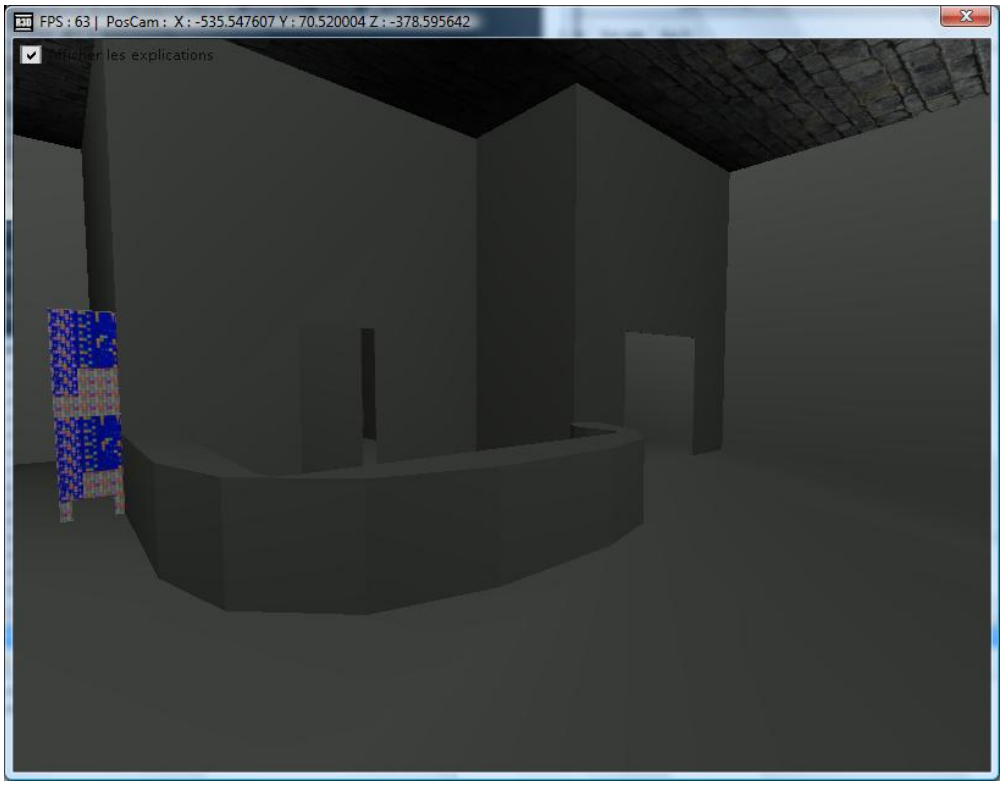
Capture de l'écran d'accueil
(version non finale)

2) La visite virtuelle

Elle est entièrement en 3 dimensions. Ci-dessous un tableau récapitulant les contrôles :

Touche	Action
Z	Avancer
Q	Aller à gauche
S	Reculer
D	Aller à droite
Espace	Sauter
Souris	Orienter la caméra
E	Activer/Désactiver les explications

Les contrôles sont classiques.



Aspect de la visite virtuelle à ses débuts

II/ Cahier des charges fonctionnel

Afin d'obtenir un produit de qualité et d'obtenir une forte compétitivité, nous avons fait un CDCF.
Nous essaierons de faire ce cahier des charges fonctionnel de la manière la plus complète possible, sachant que seule l'expression fonctionnelle du besoin est au programme de Première S-SI.

A) Présentation générale du problème

1) Le produit et son marché

- **Concept général :** Le produit servira à visiter virtuellement le CDI du nouveau lycée Richelieu.
- **Débouchés prévus :** Il pourra être téléchargé sur Internet. En fonction de la qualité du produit final et de la réaction des professeurs et élèves, le produit sera vendu par

l'intermédiaire du lycée ou distribué gratuitement. Nous pourrons aussi mettre en place un système de dons.

- **Espérance de vie commerciale :** Tant que le CDI du lycée Richelieu ne fera pas l'objet d'une nouvelle rénovation (ce qui ne devrait pas être le cas avant 10 ou 15 ans), le produit ne sera pas en fin de vie. Il faut cependant savoir que sa période la plus « active » ira de la naissance du produit jusqu'à 6 ans environ après la rénovation du lycée. Il pourrait tout à fait être téléchargé par nos futurs employeurs lorsque nous aurons quitté le lycée.
- **Concurrence :** Aucune.
- **Opportunité :** Elle est importante, le lycée étant en pleine construction et la concurrence nulle, aucun projet de ce type n'étant à notre connaissance prévu par le lycée, la ville de Rueil-Malmaison ou le Ministère de l'Education.

2) Contexte du projet

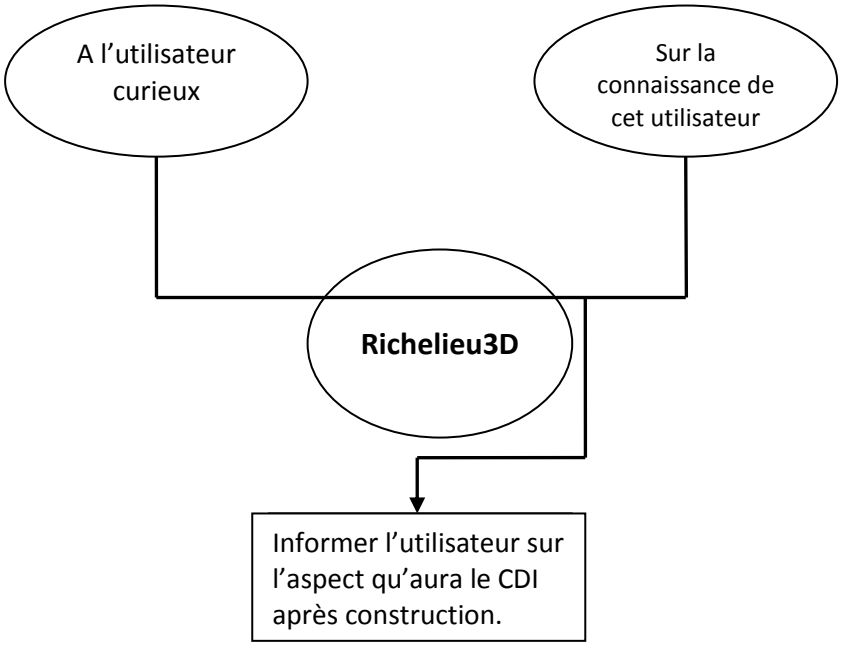
- **Nature du projet :** Programme informatique.
- **Limites :** Nous nous limiterons au CDI.
- **Contraintes :** Il faut que l'application soit exécutable, en priorité sur un système d'exploitation Microsoft Windows. Des versions pour les systèmes Linux et/ou Mac OS X pourront être ensuite produites.
- **Budget :** Même si aucun budget n'est a priori nécessaire, nous avons prévu 50€/pers.

B) Expression fonctionnelle du besoin

1) Analyse du besoin

<u>Enoncé du besoin</u>

A qui rend service le produit ? — *Sur quoi agit-il ?*



Dans quel but ?

- **Pourquoi ce besoin existe-t-il ?**

Le besoin existe car le lycée est en reconstruction, donc les personnes qui sont quotidiennement (ou qui vont l'être) dans le lycée sont intéressées par ce qu'il va devenir. Il n'existe actuellement aucune manière de pouvoir se représenter clairement le futur CDI.

- **Pour quoi ce besoin existe-t-il ?**

Comme énoncé plus haut, afin d'informer l'utilisateur sur l'aspect qu'aura le CDI après construction.

- **Qu'est-ce qui pourrait faire évoluer ou disparaître ce besoin ?**

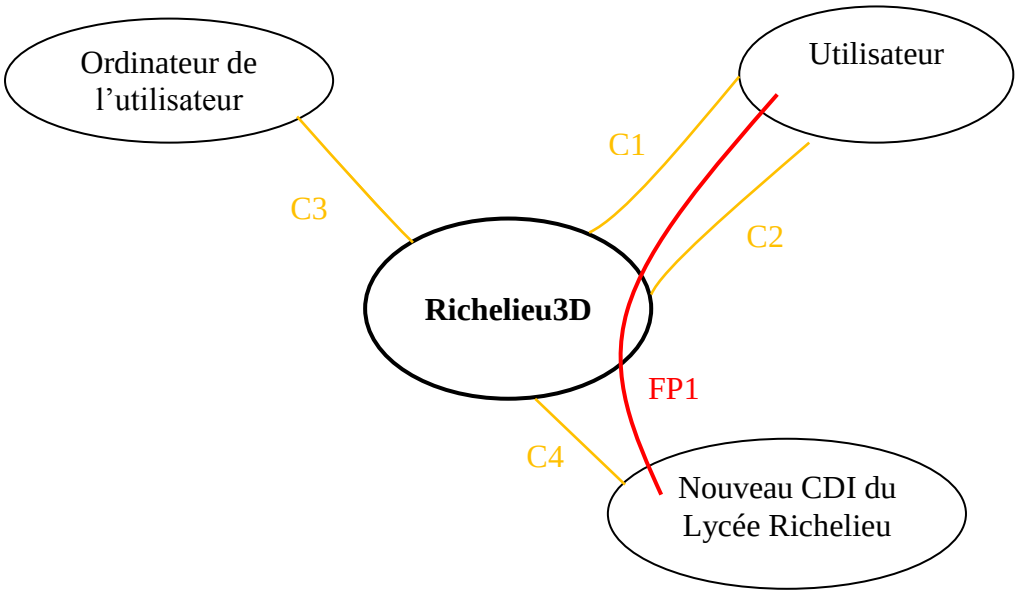
Le besoin sera moindre quand la construction du nouveau lycée sera terminée, mais les futurs lycéens pourront toujours être intéressés. Si l'activité pratiquée dans les locaux du lycée Richelieu venait à changer ou les locaux à être détruits, le besoin disparaîtrait.

2) Etude de faisabilité

Afin d'étudier la faisabilité d'un produit, il faut rechercher les fonctions de services puis formuler et caractériser celles-ci. Pour des raisons de ressources de temps l'étude n'est pas approfondie mais a été effectuée.

a) Recherche des fonctions de service

Graphe des interactions



Caractérisation des fonctions de service

	Fonctions	Critères	Niveaux
FP1	Informer l'utilisateur de l'aspect du CDI du futur lycée Richelieu	- Vraisemblance	- 3 dimensions, textures (couleurs)
C1	Etre facile d'accès pour l'utilisateur	- Contrôles - Clarté - Accès au téléchargement	- Simples et classiques - Explicite - Simple
C2	Plaire à l'utilisateur (ne concerne que la partie « GUI » ou menus, puisque nous ne sommes pas les architectes du CDI…)	- Couleurs - Graphismes	- Harmonieuses, sobre - Haute résolution
C3	Pouvoir se lancer sur l'ordinateur de l'utilisateur	- Ressources nécessaires	- Limitées (1,6GHz, 1Go RAM, Carte graphique OpenGL)
C4	Correspondre au véritable aspect du CDI du futur lycée Richelieu	- Agencement - Couleurs	- Le même que sur les plans existants - Correspondantes à celles du CDI actuel, en plus moderne

Tous les niveaux sont ultra-flexibles, nous sommes en effet le demandeur et en même temps le concepteur. Par souci des utilisateurs, nous nous efforcerons de respecter au maximum ces niveaux prédéfinis.

OpenGL est une librairie très bas niveau interagissant avec la carte graphique. Quasiment toutes les cartes graphiques existantes la supportent.

Conclusion

Après que nos objectifs ont été clairement définis, nous pouvons nous concentrer sur la réalisation du produit.

Mais d'abord il faut s'assurer que le projet est faisable du point de vu moyens. Se demander si un projet est faisable devrait être une des choses que l'on devrait faire dès le choix du sujet -ce que nous avons fait-, cela permet d'éviter d'énormes pertes de temps au cas où le but n'est pas atteignable.

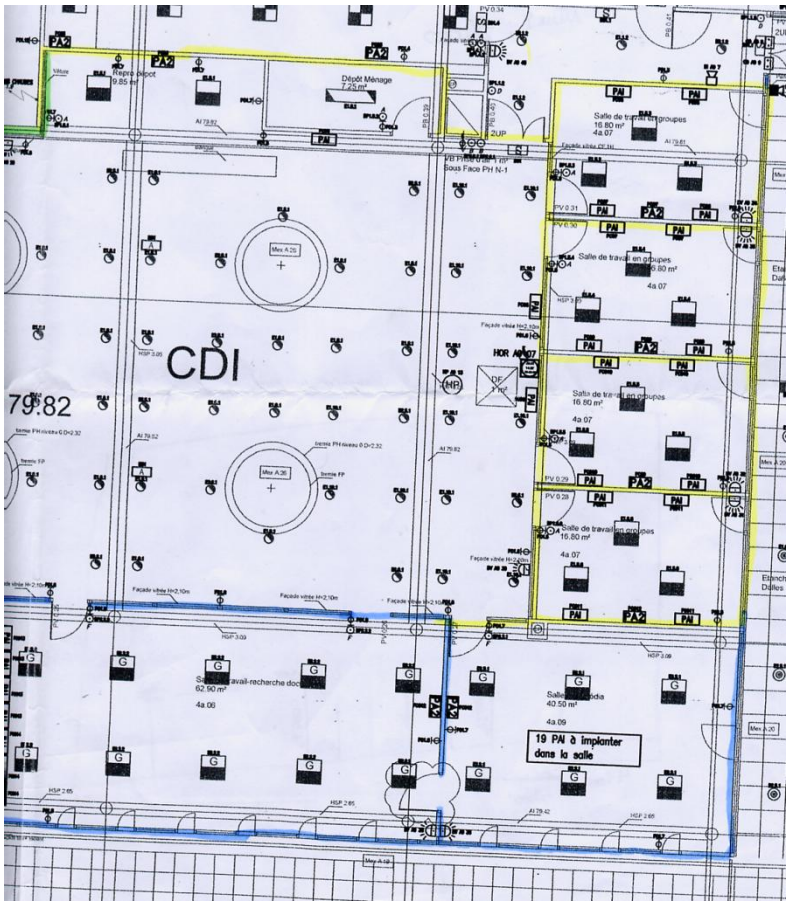
Il s'avère que nous avons tous les outils nous permettant de réaliser sereinement et gratuitement ce projet. Plusieurs logiciels sont à notre disposition : SolidWorks et sa version éducative, Code::Blocks qui est un logiciel open-source… Nous avons de plus des connaissances en modélisation 3D et en programmation, l'application est donc réalisable.

III/ Modélisation 3D

A) Démarche

1) Plans

Afin de modéliser correctement le nouveau CDI, notre professeur nous a proposé d'aller demander les plans du futur lycée. Nous sommes donc allés voir le chef des travaux, M. Philippe LETOUZEY. Malheureusement, ces plans n'ont pas été tout de suite disponibles, et nous avons eu un peu de retard. Nous avons obtenu les plans du mobilier au CDI actuel, ce qui nous a permis de modéliser ces aménagements.



Une partie du plan de l'infrastructure

Le plan définissant l'infrastructure (les murs, l'écartement…) est à l'échelle 1:100 tandis que celui concernant le mobilier est à l'échelle 1:50.

Le premier plan nous permettait donc de voir comment étaient placées les salles, les murs ainsi que des puits de lumières. L'idée des architectes de l'utilisation de puits de lumière nous a paru très bonne. Le deuxième plan détaillait la position de chaque table, chaise, porte, ordinateur, étagère, etc.

2) Pièces

Une pièce est un objet ou un sous-objet lorsque l'on parle de modélisation 3D. Par exemple, un écran d'ordinateur est une pièce, et la tour centrale de l'ordinateur aussi.
Nous avons commencé par modéliser tous les murs et le sol en une seule pièce. Le toit a été fait à part, car nous n'avions pas besoin de vérifier les collisions (la notion de collisions est précisée plus loin) avec celui-ci dans la phase de programmation. Nous avons aussi modélisé un par un chaque meuble: chaise, table, écran d'ordinateur, unité centrale, étagère, livre, porte, et écran de projection…

Nous n'avons pas réalisé d'humain : nous savons nos limites en matière de modélisation et si nous avions fait des professeurs ou des élèves, cela aurait nuit à la vraisemblance…

3) Assemblage

Si on lie une ou plusieurs pièces ensemble, cela forme un assemblage. L'assemblage peut être à grande échelle (une salle entière), mais aussi à petite échelle (celle d'un ordinateur). Reprenons l'exemple précédent: si on assemble un écran et une tour centrale ensemble, on obtiendra un ordinateur. On assemblera ensuite cet assemblage à une table, puis à cette table on assemblera une chaise. Le tout sera assemblé au sol, que l'on considère comme une pièce.

Nous avons donc procédé comme suit :

1. Assemblage d'une table avec les chaises autour afin que nous n'ayons pas à placer à chaque fois les chaises et les tables une par une. La mise en position des chaises et de la table ensemble est définie par un ensemble de contraintes d'assemblage que nous devions placer. (une contrainte définit par exemple que telle arrête doit être parallèle à telle autre etc.)
2. Assemblage de l'assemblage précédent dans le CDI, en mettant des contraintes entre l'assemblage et le sol de la salle, conformément aux mesures sur le plan.

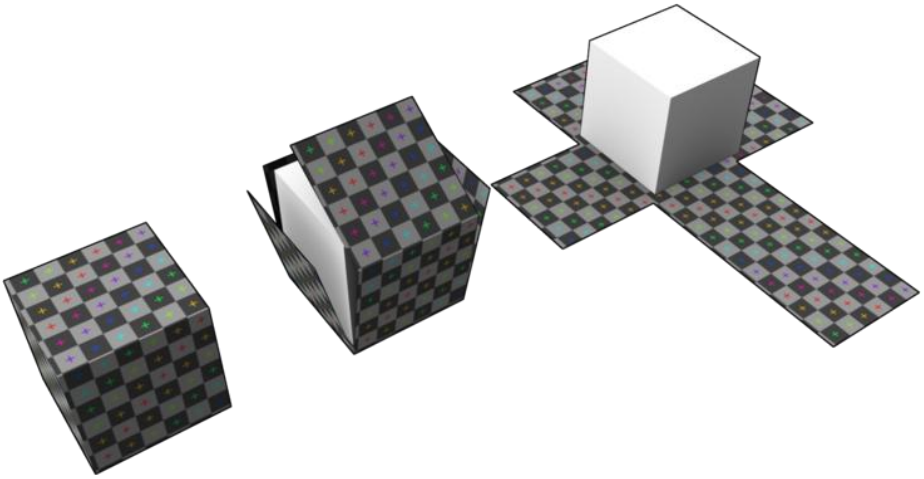
Nous avons fait de même pour les autres pièces, en essayant à chaque fois d'optimiser le temps. Les ordinateurs et leur support ont par exemple été assemblés de façon à ce qu'ils forment un bloc, ce qui nous permet de les assembler au sol encore plus rapidement.

4) Application des textures

Pour que le CDI donne une perception de réalisme et de vraisemblance, nous nous devions de colorer les murs et les objets. On pose donc des textures sur les modèles 3D.

Il y a une technique spéciale pour cela : « l'UV mapping ». Pour ce faire, on décompose l'objet 3D en plusieurs formes 2D, il suffira de mettre des couleurs sur ce fichier 2D pour texturer le modèle. La difficulté est dans la manière dont on décompose l'objet : si on « coupe » de travers le cube ci-dessous, le rendu final deviendra affreux.

On peut d'ailleurs observer une étagère bleue et violette à gauche de la capture d'écran de la visite virtuelle plus haut, cet exemple est dû à une mauvaise utilisation de la technique de l'UV mapping… C'est bien sûr maintenant corrigé.



Le cube représente le modèle 3D, le papier déroulant ce même modèle en 2D.
On « coloriera » ce papier pour donner une texture.

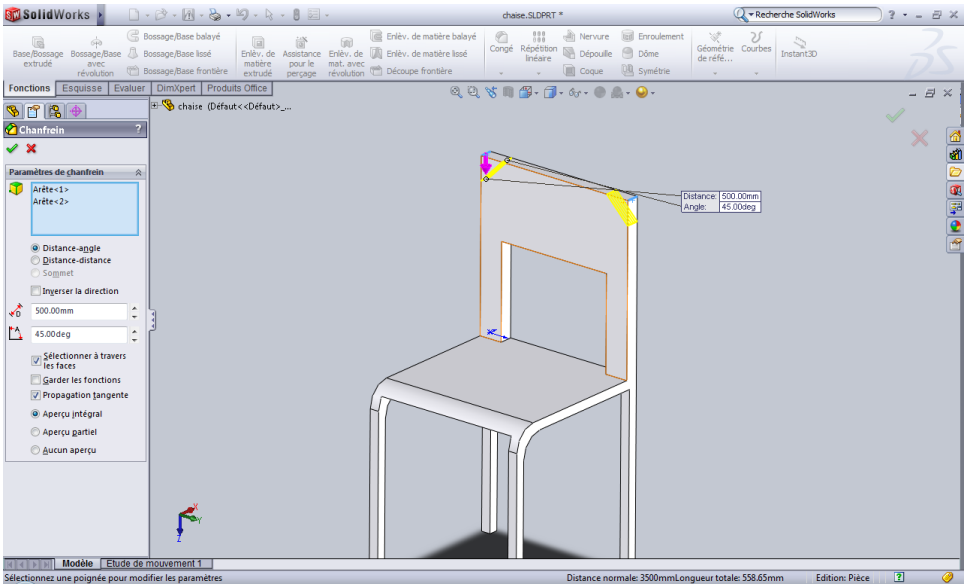
A) Présentation des outils

1) SolidWorks

SolidWorks est un logiciel de CAO (Conception Assistée par Ordinateur) 3D. Il permet donc de faire des modèles 3D.

Il est utilisé par des entreprises comme Michelin et les écoles publiques, c'est d'ailleurs pourquoi nous avons des connaissances sur ce programme depuis le collège. Et au vu de ces connaissances et de la facilité d'utilisation de cette application (surtout pour des objets), nous l'avons tout naturellement choisi.

Son système d'assemblage est très utile dans le sens où si l'on modifie une « pièce-mère » comme une étagère, toutes les étagères vont être changées en même temps dans l'assemblage !

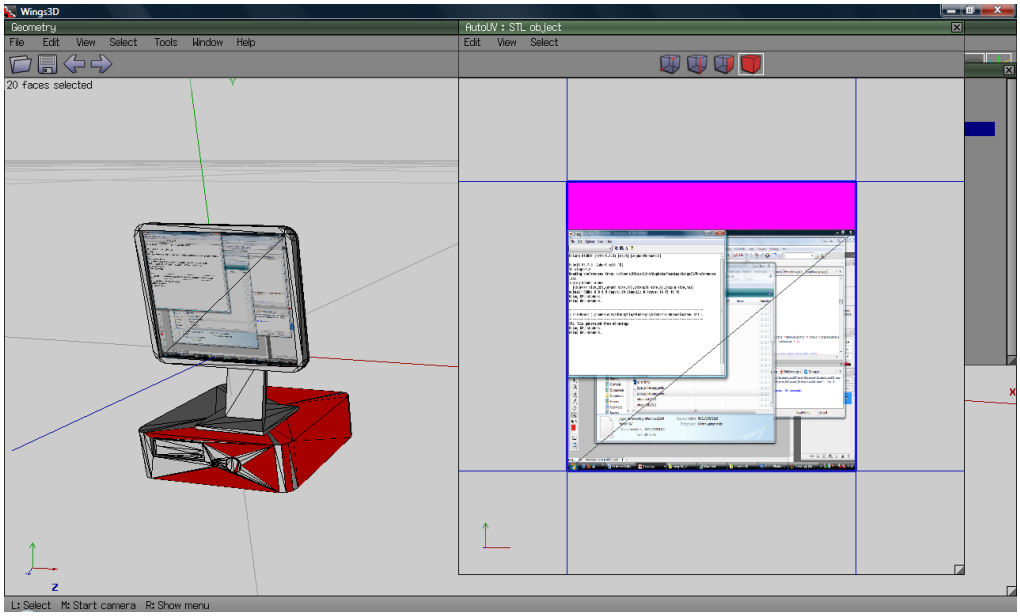


Interface de SW lors d'un ajout de chanfrein

2) Wings3D

Wings3D est un logiciel de modélisation 3D gratuit, il ne nous a pas servi pour modéliser les pièces car son interface est compliquée donc contre-productive. Il nous a cependant permis de **texturer** toutes les pièces, logiciel étant très intuitif pour l'UV mapping. Sans lui, le CDI aurait eu l'air bien triste…

La technique de l'UV mapping a été décrite plus haut.



Nous nous sommes servis de ce logiciel afin de texturer les objets et murs.

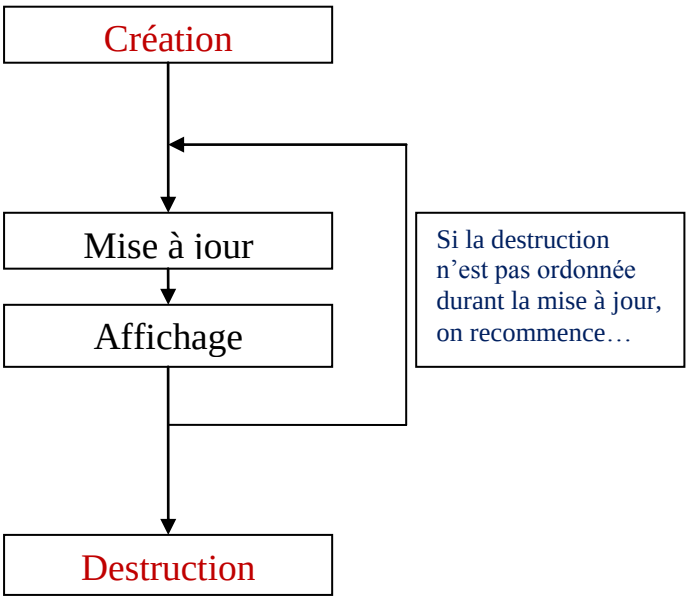
3) eDrawings

La librairie de programmation que nous avons utilisée ne supportant pas les fichiers spécifiques à SolidWorks (.sldprt et .sldasm), nous devions convertir ces fichiers en fichiers compatibles (comme des .stl). Le seul utilitaire gratuit qui nous permettait d'effectuer ce travail correctement était eDrawings, qui est d'ailleurs un produit de la gamme SolidWorks.

IV/ Programmation

A) Démarche

L'application doit pouvoir être exécutée infiniment jusqu'à ce que l'utilisateur décide de quitter. Il faut rafraîchir l'image 60 fois par seconde dans un souci de confort pour la vision de l'utilisateur.
La « vie » de n'importe quel objet (2D ou 3D) peut être schématisée ainsi :



1) La « GUI »

Ou « Graphical User Interface » en anglais. L'environnement graphique comprend pour faire simple tout ce qui est en 2D : les menus, les explications, les cases à cocher…

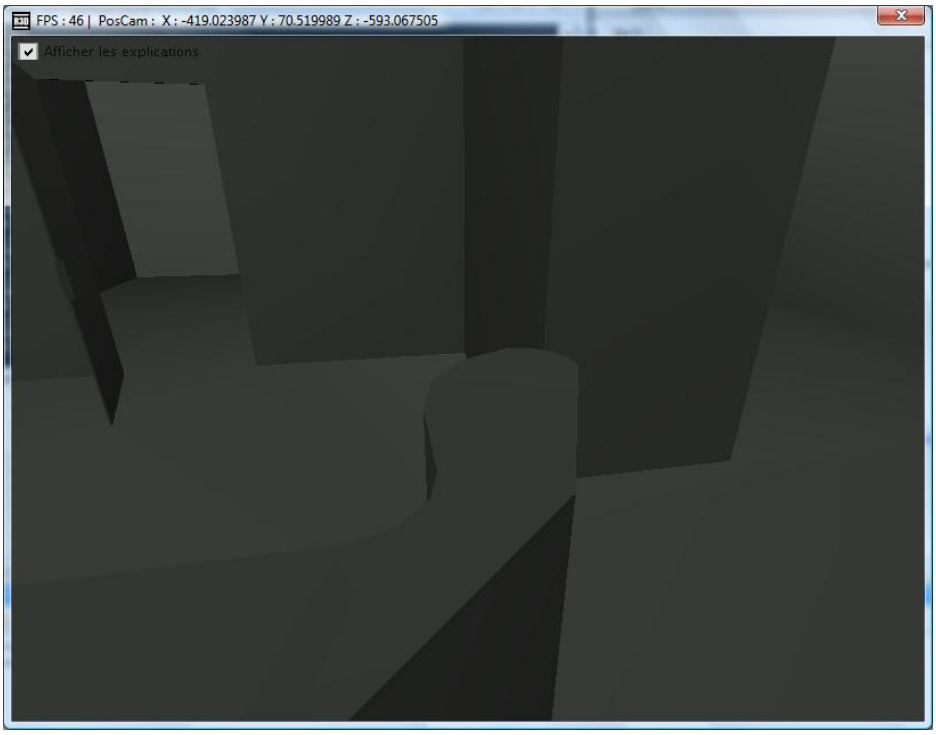
Il comprend aussi le gestionnaire d'évènements. Celui-ci regarde si une touche a été appuyée ou si un clic de souris a été provoqué, en fonction des retours, le programme exécute des instructions. Par exemple, si la touche 'Echap' est appuyée pendant l'exécution de l'application, celle-ci se ferme.

La GUI a pour but d'informer l'utilisateur sur ce qui se passe. C'est ainsi que dès que vous entrez dans une partie du CDI, un petit panneau vous informe sur la nature de la salle visitée, sa superficie, et sa fonction. Afin de savoir dans quelle partie du CDI la caméra se trouve, un algorithme analyse la position en X et Y de la caméra puis fait une comparaison avec un tableau de valeurs qui indique plusieurs informations sur cette zone (nom, superficie, explications).

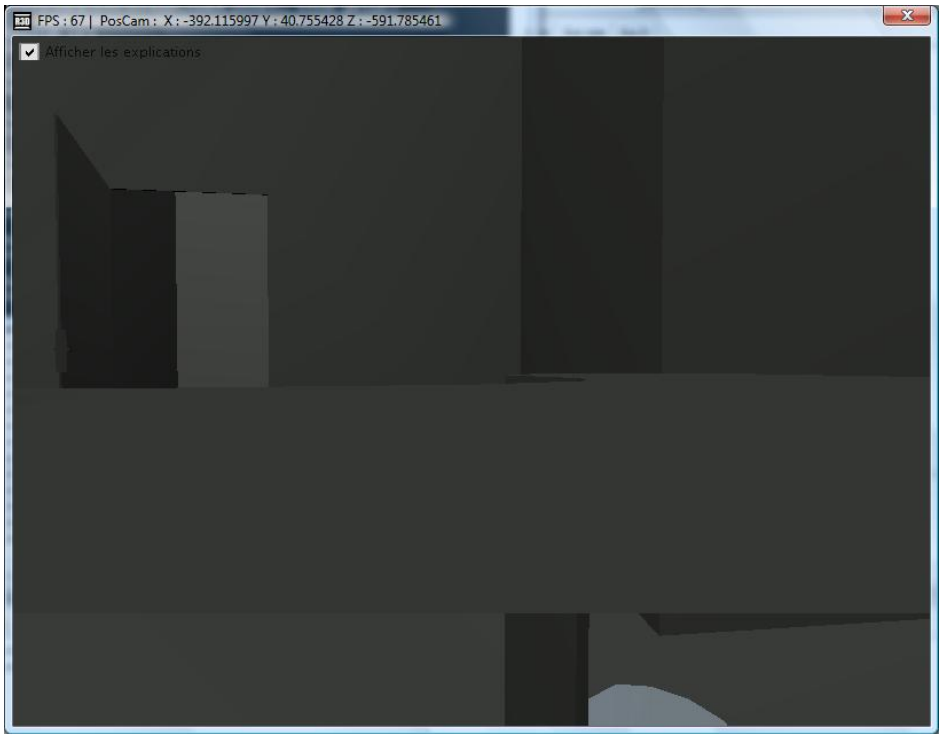
2) La scène

La scène désigne ici tout ce qui est en 3D. Quand l'utilisateur quitte les menus, le programme charge tous les modèles 3D en mémoire et les scannent afin de créer une hiérarchie de polygones que l'on va utiliser dans le but de gérer les collisions.

En effet, s'il n'y avait pas de collisions, le joueur tomberait dans le vide dès son arrivée dans la scène… Et si l'on avait mal élaboré l'algorithme qui scanne les modèles, la caméra pourrait passer par exemple à travers une étagère. Il faut à tout prix un système de collisions dans une visite virtuelle si l'on veut que le lieu soit réaliste !



Avec collisions



Sans collisions

A) Présentation des outils

1) C++

Le C++ est un langage de programmation, souvent désigné comme le successeur du C. A la différence du C qui est de type impératif (suite d'instructions), le C++ est utilisé comme langage de POO (Programmation Orientée Objet).

Dans la POO, pour faire simple, on peut déclarer une structure de variables (ou ensemble de valeurs associées à un nom) que l'on appellera « Objet ». Chaque objet a ses caractéristiques qui sont les attributs et les méthodes.

Un attribut est une variable. Par exemple, «posX» qui définirait la position en X de l'objet. Une méthode est une fonction, une suite d'instructions. Elle pourrait par exemple se nommer « setPosX » et permettrait après déclaration de redéfinir la variable « posX ».

```
toit->setRotation(core::vector3df(270, 0, 0));
toit->setMaterialFlag(video::EMF_LIGHTING, true);
toit->setMaterialFlag(video::EMF_ANISOTROPIC_FILTER,true);
```

A la première ligne, l'objet « toit » fait une rotation sur l'axe des X de 270°.
Objet = 'toit' / Méthode ='setRotation()' / attribut ='core::vector3df(270, 0, 0)'

2) Irrlicht

Une bibliothèque en programmation désigne une sorte de « macro-programme » contenant plusieurs instructions/fonctions que peut réutiliser le codeur. C'est un ensemble de fonctions, les fonctions étant des suites d'instructions données à l'ordinateur. Dans l'extrait de code ci-dessus, la fonction « setMaterialFlag(…) » fait partie de la bibliothèque Irrlicht. Cela permet de ne pas avoir à écrire 60% de code supplémentaire et donc économise du temps précieux.

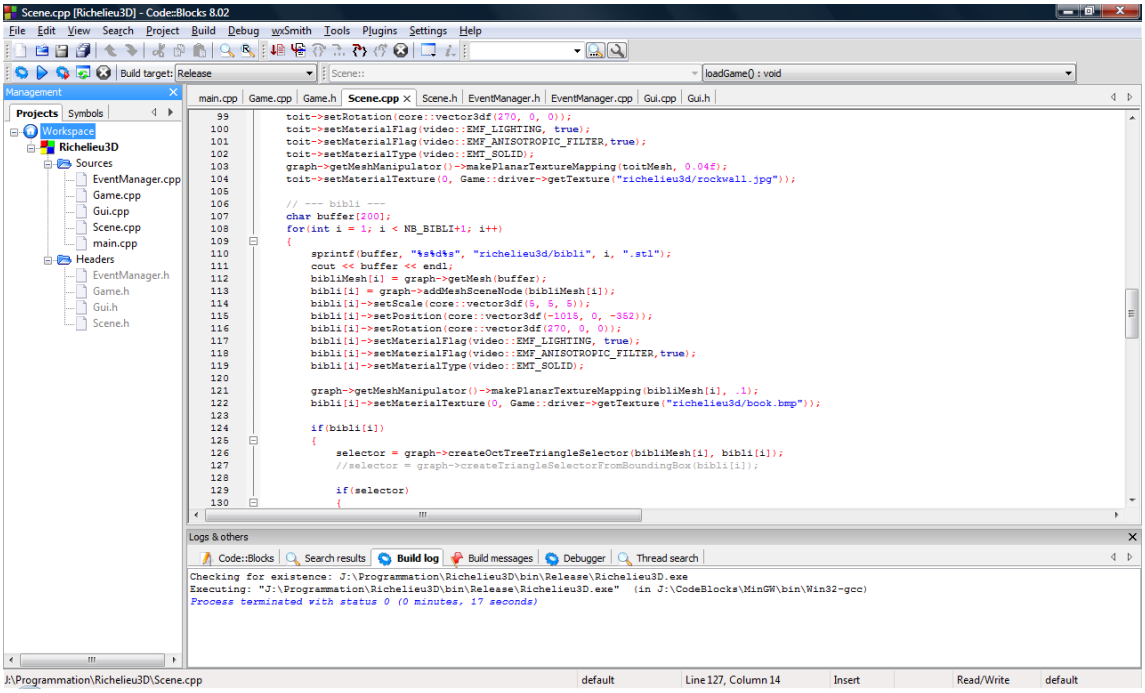
En plus d'être une bibliothèque, Irrlicht est aussi un moteur graphique. Cela signifie qu'il va gérer tout seul ce qui intervient entre la carte graphique et nos modèles 3d. Un moteur graphique permet surtout d'alléger et de simplifier le code.

3) Code::Blocks

Code::Blocks est un logiciel open-source qui rentre dans la catégorie des « environnements de développement intégré ». On utilise plus communément l'abréviation anglaise IDE.

Il permet d'éditer du texte (ici du code) tout en pouvant compiler et lancer le programme dès que le développeur le veut. Le logiciel analyse le code et le colorise dans le but de permettre l'utilisateur de mieux se repérer.

La compilation consiste à transformer en binaire (0 et 1) le code, le fichier binaire résultant sera exécutable par le système d'exploitation.



Gestion d'onglets, complétion du code, rapidité. C::B a tout pour plaire.

V/ Conclusion

A) Quelques chiffres

2150 : Le nombre de polygones rien qu'avec les murs.
469 : Le nombre de contraintes entre chaque étagère.
687.2 m² : C'est la superficie modélisée, sans compter tous les objets.
210 : Le nombre de chaises dans le CDI.

Un millier : Le nombre de lignes de code. Ni trop ni pas assez, c'est optimisé.
10 secondes : Le temps que met le code à se compiler.
60 : C'est le nombre d'images affichées par seconde.
3 secondes : Le temps qu'il faut à l'application pour charger la scène 3D.

B) Améliorations possibles

Toute chose peut s'améliorer, Richelieu3D n'échappe pas à la règle. C'est pourquoi nous serons à l'écoute des bugs possibles et des suggestions que nous feront les utilisateurs. Nous sommes conscients que des améliorations telles que l'ajout de personnes ou de sons seraient bienvenues, mais elles nous prendraient beaucoup de temps pour l'effet obtenu (surtout si l'application n'est pas vendue).

On pourrait aussi faire en sorte que l'utilisateur puisse interagir davantage avec l'environnement virtuel : il serait possible d'ouvrir et fermer les portes, de s'assoir, de courir... On pourrait même faire en sorte que l'application soit reliée à Internet et que l'on puisse parler avec les autres utilisateurs en ligne… Il y a beaucoup de choses à faire, mais toutes ne sont pas facilement réalisables ou en tout cas pas dans le temps dont nous disposons. Cependant, s'il y avait une forte demande des utilisateurs, une suggestion d'amélioration faite pourrait être implémentée dans la visite virtuelle.

C) Téléchargement

Un site web sera à votre disposition à cette adresse :

http://louislagrange.free.fr/Richelieu3D

Le téléchargement de Richelieu3D se fera sur le site web. Si votre document a été imprimé en noir et blanc ou si vous souhaitiez télécharger ce document, vous pourrez le visualiser en couleurs sur ce site.

D) Réponse à la problématique

La réalisation de la visite virtuelle d'un lieu demande donc des ressources de connaissances et de temps, ainsi que de la persévérance. Il faut comme pour tout produit faire un cahier des charges fonctionnel. On peut réaliser le produit de façon complète et personnelle en couplant modélisation 3D et programmation mais il faut soigneusement choisir ses outils. Il faut se procurer les plans du lieu bien évidemment.

VI/ Annexes

A) Lexique

Pathfinding : Algorithme de recherché de chemin.
UV Mapping : C'est une technique permettant d'appliquer des textures sur un modèle 3D.
Bibliothèque (programmation) : C'est un ensemble de fonctions qui permettent de ne pas avoir à réinventer la roue.

B) Liens

- Richelieu3D: http://louislagrange.free.fr/Richelieu3D
- SolidWorks: http://www.solidworks.fr
- Wings3D: http://www.wings3d.com
- eDrawings: http://www.edrawingsviewer.fr
- Irrlicht: http://irrlicht.sourceforge.net
- Code::Blocks: http://www.codeblocks.org