



EPS

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Eng. Tècn. Informàtica de Gestió. Pla 2001

Títol:

Entorn de treball per a visualitzar escenes tridimensionals

Document:

Resum

Alumne: Salvi Pi Bonany

Director/Tutor: Mateu Sbert Casasayas

Departament: Informàtica i Matemàtica Aplicada

Àrea: Llenguatges i Sistemes Informàtics

Convocatòria (mes/any): 01/08

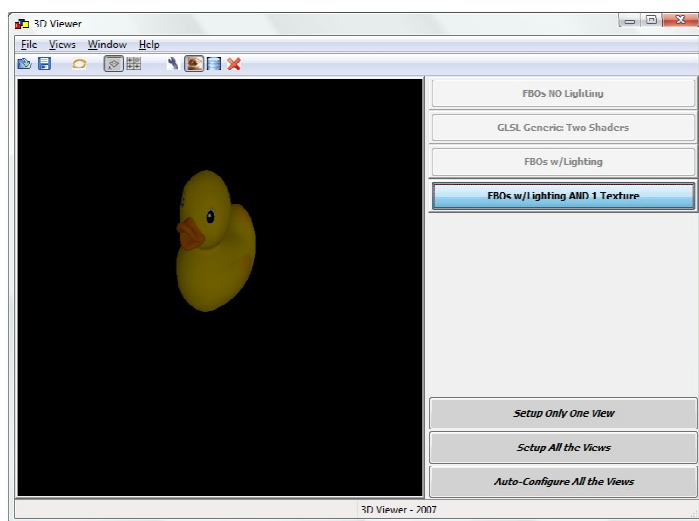
resum de la memòria

El camp dels gràfics en tres dimensions és cada vegada més interessant i prometedor, i un dels grans impulsors del desenvolupament accelerat de la indústria dels gràfics per computador han estat precisament les consoles de joc personals, que incorporen tots els avenços en aquesta matèria per millorar el realisme i la vistositat dels seus jocs. També els avenços en gràfics permeten desenvolupaments que ajuden en CAD, realitat virtual, etc.

La intenció d'aquest projecte és proporcionar un **entorn de treball** per tal de **visualitzar models tridimensionals en temps real**. Aquest projecte perdria part del seu sentit si només pretengués això, ja que existeixen, tan al mercat com en codi lliure, molts programes que realitzarien aquesta tasca a la perfecció. Els objectius que es marca aquest projecte són en resum aquests:

- a) Proporcionar una interfície gràfica per poder visualitzar interactivament una escena, modificant-ne els seus elements, com llums, càmeres i mètodes de visualització entre altres.
- b) Aconseguir un disseny que faci el projecte altament revisable i reutilitzable en el futur, i serveixi per tant de plataforma per provar altres projectes, com per exemple diferents tipus de càlculs d'il·luminació.

A l'hora de dissenyar l'aplicació s'han hagut de superar una sèrie de problemes i necessitats, així com prendre les decisions de l'entorn i les eines de programació a utilitzar. Es pretenia que l'aplicació utilitzés el potencial de les targetes gràfiques d'última generació, que tenen, com s'ha explicat, capacitat de programació. Alhora es volia conservar una certa independència de l'aplicació respecte el sistema operatiu, així que es va escollir desenvolupar la interfície amb les biblioteques lliures multiplataforma anomenades **wxWidgets**. Així doncs, pel que fa al **primer objectiu**, el d'aconseguir una interfície gràfica per visualitzar l'escena, s'han usat per aconseguir-ho directrius que han donat una presència semblant a la que tenen la majoria de programes sota **Windows** usant per això la biblioteca esmentada, que té la propietat d'adaptar-se al *feel & look* del sistema.



És el **segon objectiu** el que diferencia el projecte de qualsevol altre visor de models tridimensionals, i que ens permetrà adaptar l'aplicació als estàndards i desenvolupaments de la informàtica gràfica. Aquest objectiu d'aconseguir aquesta llibertat de programació s'ha estructurat en el que s'anomena *View*. Així un *View* és el mòdul de codi encarregat de definir quina tècnica volem utilitzar per pintar els elements d'una escena.

OpenGL és una especificació estàndard que defineix una API (Application Programming Interface, conjunt de definicions de funcions per als programadors) multilinguatge i multiplataforma per a escriure aplicacions que produeixen gràfics 3D. L'aplicació resultant d'aquest projecte s'ha basat en aquesta API, ja que la seva escalabilitat ha permès que no s'hagi estancat el seu desenvolupament i inclogui ràpidament els últims desenvolupaments de la indústria. La creació d'extensions, una sèrie d'afegits sobre les funcionalitats bàsiques, per tal d'aprofitar les creixents evolucions tecnològiques contribueix a aquest objectiu. Podem ressenyar per exemple la inclusió dels GLSL (un llenguatge de *shaders* propi) com a estàndard en la versió 2.0 d'OpenGL, que ens permeten programar les targes gràfiques d'última generació.

Així doncs, basant-nos en OpenGL, un *View* podrà usar ordres d'aquesta biblioteca per mostrar-nos un determinat model tridimensional. Ara bé, un *View* per exemple pot només usar ordres bàsiques de OpenGL per pintar la geometria, sense tenir en compte la il·luminació. També se'n poden definir de més complexos, que incloguin la il·luminació, les textures, el pas per un o varis *shaders*, ... Aquest aspecte queda a l'entera disposició que en faci el programador de *Views*, que serà qui definirà quins elements s'han de requerir per usar aquell *View*, i quines tècniques usarem per pintar-los.

El segon objectiu marcat pel projecte també suposa escollir un estàndard principal per carregar diferents models tridimensionals. COLLADA (**COLL**aborative **D**esign **A**ctivity) és un estàndard obert que estableix un esquema digital per a les aplicacions 3D interactives i és interessant perquè està recolzat per la indústria 3D, dissenyadors i desenvolupadors entre d'altres. Per això i també per les eines i biblioteques obertes que han sortit a la seva empara, el fa una elecció òptima per incloure un primer format de models tridimensionals.

En resum, obtindrem per una banda l'ús normal que qualsevol **usuari** pot esperar de l'aplicació per mostrar un model en tres dimensions, i per l'altra l'ús del **programador** que investiga noves funcionalitats. Les funcionalitats bàsiques de l'**usuari** de l'aplicació són: carregar l'escena (models 3D, llums, càmeres, etc.), modificar els elements d'aquesta i visualitzar l'escena en si. El **programador** per la seva banda, pot esperar un entorn operable i força desenvolupat que li permetrà centrar-se en el desenvolupament i prova de noves tècniques de visualització. Aquestes tècniques les podrà provar creant un nou mòdul de visualització o *View*, despreocupant-se de la resta (carregar el model, mostrar els elements, ...)

A continuació es presenten alguns resultats obtinguts usant diferents Views inclosos a l'aplicació resultant.

Aquests Views representen diferents tècniques senzilles de visualització usant OpenGL.

El primer de la fila de la dreta és producte d'una visualització que no té en compte cap element que no sigui només la geometria.

El segon de la fila té en compte els llums i el material del model a l'hora de mostrar-lo.

El tercer, finalment, fa ús de les textures per tal de donar més realisme als models tridimensionals.

En els dos casos posteriors es fa ús de l'extensió d'OpenGL anomenada GLSL (GL Shading Language) que permet programar les targetes gràfiques per obtenir altres efectes de visualització. Això s'aconsegueix definint els *shaders*, que són els petits codis que substitueixen la funcionalitat de la tarja a l'hora de pintar. El de l'esquerra defineix un colorejat tipus dibuix d'animació (*toon*) i el de la dreta aplica un patró (de maons en aquest cas) a l'hora de fer la visualització.

Com podem veure el món dels gràfics, i més amb la introducció d'eines com GLSL i les actuals targetes gràfiques, té unes possibilitats immenses pel que fa no només a la il·luminació, sinó a altres àmbits, que permeten introduir realisme i efectes diversos a les representacions dels models tridimensionals.

