

Università degli Studi di Bologna

Dottorato di Ricerca in Storia e Informatica

XIX Ciclo

Programmazione per la fruizione del progetto Nu.M.E. attraverso Internet

TESI DI DOTTORATO

Relatrice:

Dott. ELENA BONFIGLI

Coordinatrice:

Prof.ssa FRANCESCA BOCCHI

Presentata da:

Dott. TIZIANO DIAMANTI

Anno Accademico 2006- 2007

Alla mia famiglia,

per aver compreso

con amorevole pazienza

le stravaganze dei miei studi

Tiziano

INTRODUZIONE	1
1 IL PROGETTO NU.M.E.	2
2 PRINCIPI DI COMPUTER GRAFICA 3D	6
2.1 Rasterizzazione	7
2.1.1 Viewport.....	8
2.2 Proiezione prospettica.....	8
2.3 Prospettiva parallela	9
2.4 Proiezione prospettica.....	11
2.5 Eliminazione delle linee nascoste.....	13
2.5.1 Z-Buffer.....	14
2.6 Texture Mapping.....	15
2.7 Ray Tracing	16
2.8 Render to texture	17
2.9 Evoluzione dell'hardware.....	19
2.10 Ambienti e metodi di visualizzazione	21
2.10.1 Stereoscopia.....	21
2.11 Sistemi immersivi di visualizzazione.....	23
2.11.1 Cave.....	24
2.12 Le maggiori librerie grafiche di base	25
2.12.1 OpenGL.....	25
2.12.2 Direct 3D	26
2.13 Le librerie basate su Scene Graph.....	26
2.13.1 OpenGL Performer.....	27
2.13.2 Open Scene Graph.....	28
2.14 Utilizzo e vantaggi di una libreria a Scene Graph	29
2.14.1 Caricamento da file di nodi scene graph.....	32
3 APPLICAZIONE DELLA COMPUTER GRAFICA AI BENI CULTURALI	34
3.1 Esigenze pecuriali di questo ambito	34
3.2 L'esperienza Visman.....	37
3.3 Lo Scengraph come organizzazione della scena collegabile alla logica del modello 40	
3.3.1 Scansione dello scene graph.....	47
3.3.2 Picking.....	47
3.4 Tipico flusso di lavoro per produrre un'applicazione grafica 3D per i beni culturali 48	
3.4.1 Modellazione 3D	49
3.4.2 Multigen Creator	49
3.4.3 Autodesk 3D Studio Max con plug-in Flight Studio	51
3.4.4 Verso l'open source nella modellazione: Blender	52
3.4.5 Preparazione dei dati da collegare al modello 3D.....	54
3.4.6 Visualizzatore 3D apposito.....	55
4 ORGANIZZAZIONE DEI DATI	56
4.1 Database Management System.....	56
4.1.1 Il modello di database gerarchico	57
4.1.2 Il modello di database reticolare.....	58
4.1.3 Il modello di database relazionale	59
4.1.4 Database relazionali e archeologia	64
4.2 Database in locale o tramite Server	64
4.3 Open source nei database.....	65
5 NU.M.E. IN INTERNET.....	67
5.1 Applicazione in locale o tramite web.....	67
5.2 Breve storia di Internet.....	67
5.3 Internet attuale	69

5.4	<i>I web browser attualmente più diffusi</i>	69
5.4.1	Mozilla Firefox.....	69
5.4.2	Microsoft Internet Explorer.....	70
5.5	<i>Google Earth</i>	70
5.6	<i>Il VRML</i>	71
5.7	<i>Grafica 3D interattiva sul web: plug-in per browser</i>	71
5.7.1	Plug-in per il progetto Nu.M.E.	72
6	CONCLUSIONI E SVILUPPI FUTURI.....	74

Introduzione

Il progetto Nu.M.E., nuovo museo elettronico della città di Bologna, la città in quattro dimensioni si può considerare il capostipite dei progetti di ricerca storici a cui viene applicata una ricostruzione virtuale 3D al computer.

Sempre più ricercatori utilizzano queste tecnologie per guadagnare maggiori capacità di comprensione, di divulgazione, di studio e di rappresentazione nelle proprie ricerche.

Il framework sviluppato per il progetto di ricerca non si ferma alla sola navigazione di modelli 3D in tempo reale, ma permette il collegamento degli oggetti tridimensionali a database relazionali, permettendo allo studioso di dare alle applicazioni lo spessore di rigore scientifico che spesso manca nelle applicazioni per un pubblico vasto.

Dopo un'analisi della teoria, delle problematiche e delle librerie riguardanti la grafica 3D su computer, la tesi prende in considerazione le evoluzioni dell'informatica negli ultimi anni, sia dal punto di vista dell'hardware che la disponibilità e la diffusione di Internet a banda larga.

Queste evoluzioni hanno permesso al progetto Nu.M.E. di essere fruito dapprima in un teatro virtuale, attraverso quello che allora era un supercalcolatore grafico, poi in un normale PC domestico ed infine di essere portato sul Web.

L'obiettivo della tesi è mostrare i meccanismi che sono alla base del progetto dal punto di vista tecnico ed informatico, ovvero l'analisi di strumenti, tecniche e tecnologie per la fornitura di uno strumento per la ricostruzione di ambienti 3D con contenuti storici e per la sua fruizione su vari dispositivi fisici ed Internet.

1 Il progetto Nu.M.E.

Il progetto NuME ¹ (Nuovo Museo Elettronico), avviato verso la metà degli anni '90, costituisce il primo progetto del genere in Italia, ponendosi come punto di riferimento per molti lavori successivi.

Pensato come una versione aggiornata e moderna del museo della città, sviluppa un'applicazione informatica innovativa, dal momento che:

ricorre alla Realtà Virtuale per epoche storiche fino a quel momento refrattarie all'uso delle tecnologie digitali;

pone a suo fondamento concettuale una rigorosa fedeltà alle fonti storiche;

inserisce la dimensione temporale nella navigazione in ambiente 3D, perseguendo l'idea della città in quattro dimensioni;

è multi piattaforma.

Il punto di partenza è la ricostruzione della città attuale. Per la creazione delle texture viene avviata anche una campagna fotografica del centro storico di Bologna, elaborando quindi le immagini in modo da rimuovere ogni traccia di superfetazione (segnaletica stradale, cartelloni pubblicitari, auto, ecc.). Dal modello virtuale della Bologna attuale ci si spinge quindi indietro nei secoli, cercando di visualizzare per la medesima area le trasformazioni verificatesi nel corso del tempo.

Mentre gli edifici della città appartenenti all'epoca romana sono finiti a sei, sette metri sotto il livello attuale del suolo, quelli di epoca medioevale, quando non sono andati distrutti, sono ancora parte integrante del paesaggio urbano, a

¹ <http://www.storiaeinformatica.it/nume/italiano/ntitolo.html>

F. Bocchi, Medioevo Virtuale, in "Medioevo", n. 11, nov. 1998; M.E. Bonfigli, L. Calori, A. Guidazzoli, NuME: a Virtual Historic Museum of the City of Bologna, Proceedings of SAC2000-ACM Symposium on Applied Computing (J. Carroll, E. Damiani, H. Haddad, D. Oppenheim Eds.), Villa Olmo, Como, Italy, March 2000, vol. 2, pp. 956-961; M.E. Bonfigli, A. Guidazzoli, A WWW Virtual Museum for Improving the Knowledge of the History of a City, in J.A. Barcelo, M. Forte, D. Sanders, Eds., Virtual Reality in Archaeology, ArcheoPress, May 2000; F. Bocchi, M.E. Bonfigli, M. Ghizzoni, R. Smurra, F. Lugli, The 4D Virtual Museum of the City of Bologna, ACM SIGGRAPH99 Conference Abstracts and Applications, Los Angeles (USA), August 1999, pp. 8-11.

partire dalle Due Torri. E' per questo motivo che la Bologna medioevale è stata individuata come punto di riferimento fondamentale per le ricostruzioni del passato. Del resto, nella prima metà del XIII secolo la città contava ben 50.000 abitanti ed era già sede di una rinomata Università.

Aspirando ad essere un prodotto fortemente scientifico, la ricostruzione 3D ha riguardato esclusivamente ciò che è stato possibile definire con precisione tramite le fonti, rifiutando qualsiasi indulgenza verso ricostruzioni più attraenti ma non totalmente suffragate dai dati.

In assenza di fonti iconografiche, la ricostruzione della Bologna medioevale si è basata principalmente sull'interpretazione di prove documentarie. La fonte principale è il Libri Terminorum del 1294, un registro notarile di tutti gli indicatori a livello del terreno che definivano il confine tra spazi pubblici e spazi privati, registrando accuratamente anche i sostegni dei portici. Colonne, sostegni, mura perimetrali, distanze tra gli edifici, tutto è segnato e misurato a partire da alcuni edifici principali, alcuni dei quali ancora esistenti.

Altra fonte di grande rilievo è l'affresco fatto realizzare nel 1575 dal bolognese papa Gregorio XIII nei suoi appartamenti privati a Roma, in quella che ora si chiama sala Bologna nei palazzi Vaticani. In uno spazio di circa 6 m. per 4 m. è riprodotta la città di Bologna all'interno delle mura medioevali. Nell'ambito del progetto NuME è stata portata a termine anche un'accurata campagna fotografica per studiare la mappa in tutti i suoi più minuti particolari, compresa la forma delle finestre e delle arcate dei portici.

Il Libri Terminorum, l'affresco della sala Bologna e numerose altre fonti hanno costituito il punto di partenza ineludibile e sono confluiti nelle ricostruzioni 3D, georeferenziate e posizionate su un modello digitale del terreno, per gli scenari medioevale, ottocentesco e attuale.

Fin dall'inizio il progetto si è caratterizzato per una marcata multidisciplinarietà e per una spiccata sensibilità alle novità informatiche (presentato anche a SIGGRAPH 1999). Realizzato inizialmente in VRML ha subito varie trasformazioni man mano che venivano individuate modalità di

modellazione e navigazione più appropriati. Nel 2000 era già disponibile l'applicazione di NuME per la navigazione stereoscopica nel Teatro Virtuale del CINECA.

Tra i prodotti di Realtà Virtuale italiani realizzati per i Beni Culturali, NuME può essere considerato a tutti gli effetti un progetto antesignano di molti tra gli sviluppi successivi.

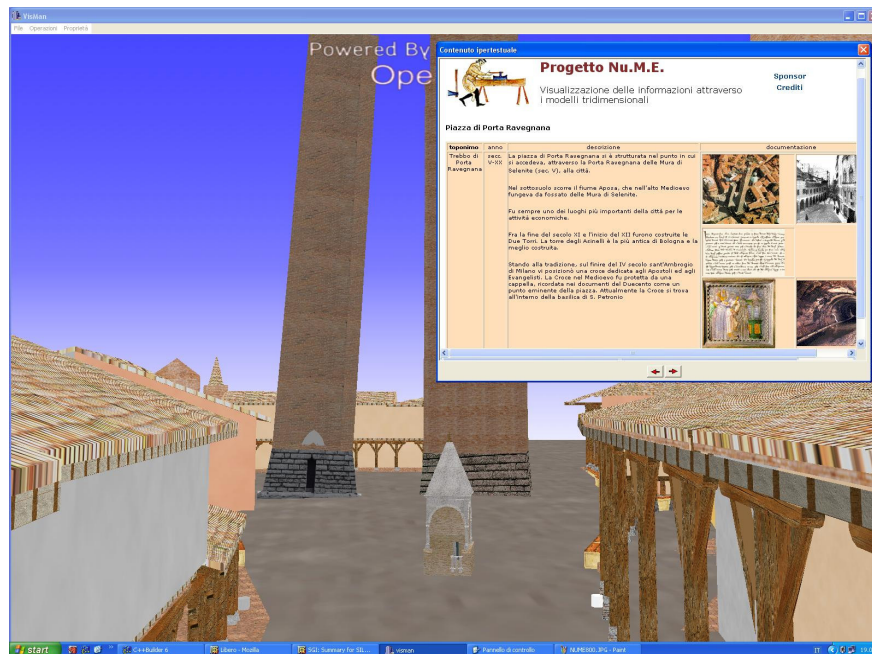


Figura 1: Via Rizzoli, Bologna. Progetto NuMe salto nel tempo, XIII secolo.

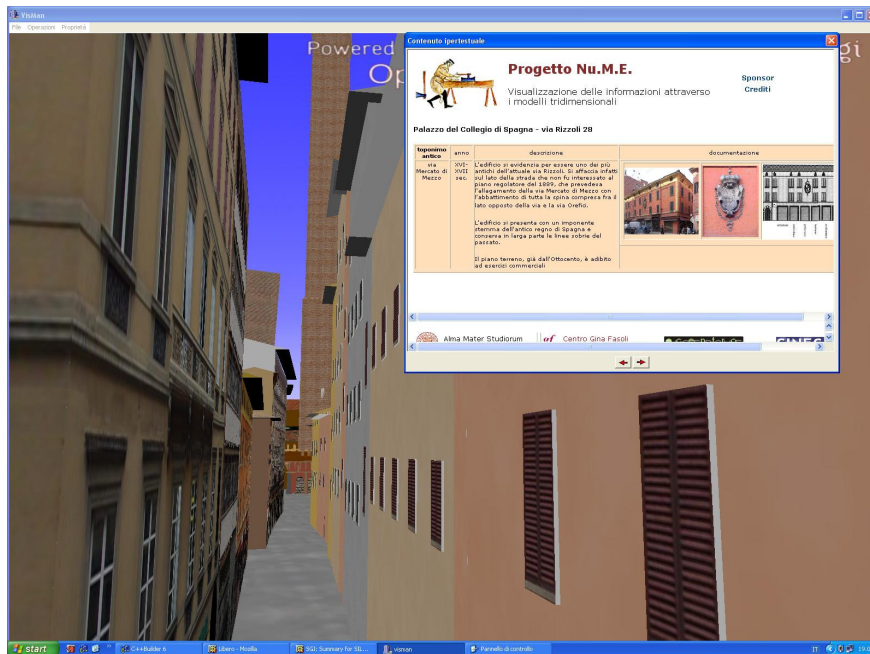


Figura 2: Via Rizzoli, Bologna. Progetto NuMe salto nel tempo, XIX secolo.

Il risultato di tutto questo è un vero e proprio museo virtuale, in grado di offrire al visitatore la possibilità di vedere scenari ricostruiti con oggetti e architetture anche non più visibili ai giorni nostri, e di avere a disposizione informazioni utili per una corretta fruizione dei modelli che si stanno osservando. Il modello risulta così non solo una ricostruzione geometrica ma anche un'interfaccia dalla quale è possibile accedere ad informazioni organizzate in un database complesso, che potrà essere integrato dagli storici anche al termine della realizzazione di un progetto, in modo da risultare sempre aggiornato.

2 Principi di computer grafica 3D

La computer grafica tridimensionale si basa sul concetto che gli oggetti siano rappresentati da vertici nello spazio interconnessi da segmenti a formare dei triangoli. Il triangolo costituisce infatti l'elemento base, il mattone fondamentale per ricostruire qualsiasi oggetto nella realtà virtuale. Punti, linee, segmenti, triangoli e in generale poligoni convessi vengono chiamati primitive grafiche; nella seguente immagine abbiamo riassunte le tipiche primitive grafiche di basso livello:

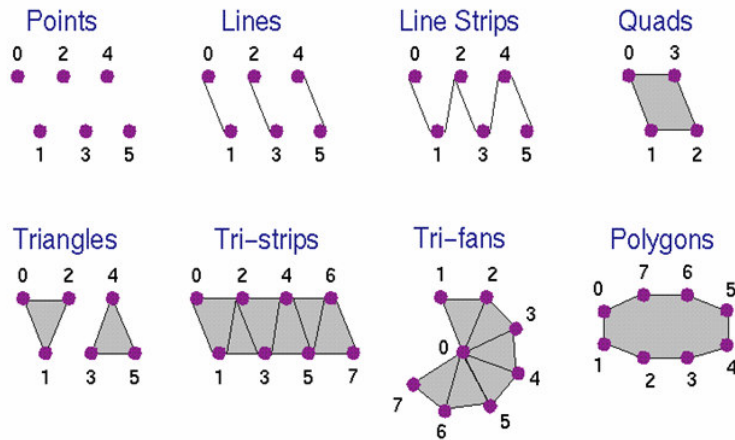


Figura 3: le primitive di OpenGL

Dunque i nostri oggetti della scena virtuale saranno in realtà formati da un insieme di punti e di interconnessioni tra di essi a formare delle primitive grafiche, che sono direttamente disegnabili dalle librerie grafiche e, in ultima analisi dall'hardware grafico.

La seguente immagine mostra come si possono rappresentare elementi architettonici con questa tecnica:

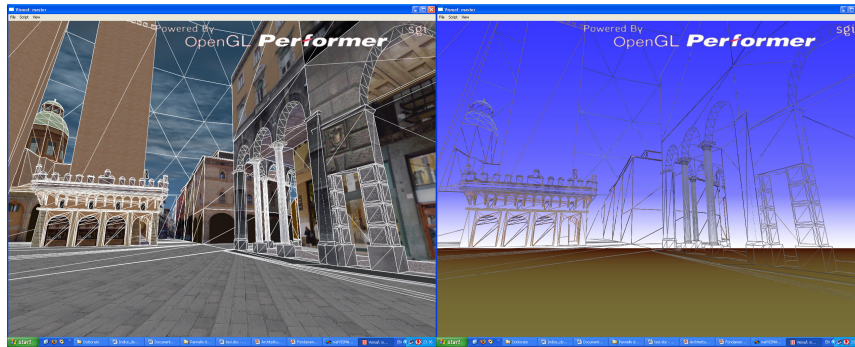


Figura 4: Elementi architettonici in computer grafica 3D

2.1 Rasterizzazione

Come risaputo, le immagini che appaiono sui monitor dei computer sono composte da un numero variabile di punti (detti pixel dall'inglese picture element) che formano una griglia con un numero di righe e colonne variabile (la cosiddetta risoluzione).

Tutti gli elementi matematici (punti, linee, poligoni, ecc.) devono essere disegnati con un'approssimazione tramite i punti all'interno di questa griglia. Questo processo è detto rasterizzazione, dall'inglese raster.

Le seguenti illustrazioni riguardano gli algoritmi di rasterizzazione di una linea e di un triangolo pieno:

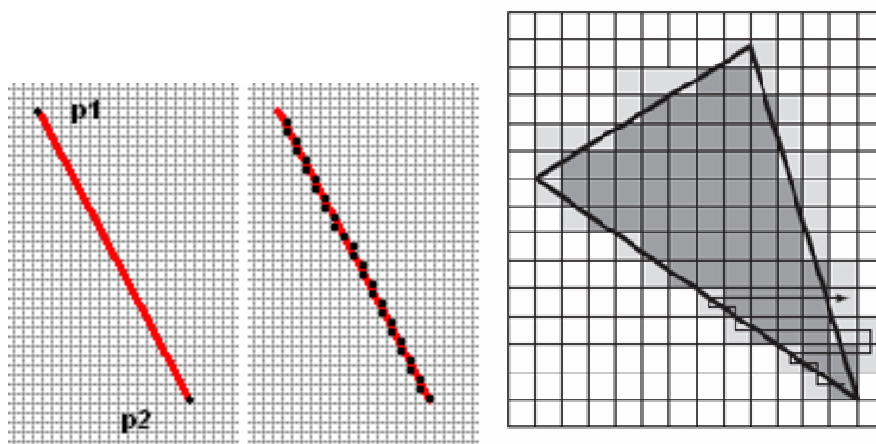


Figura 5: esempi di rasterizzazione, linea e triangolo

Ovviamente, maggiore la risoluzione, migliore l'approssimazione dell'elemento matematico alla sua rappresentazione mediante griglia di punti.

2.1.1 Viewport

La viewport rappresenta la porzione della finestra dove l'immagine del mondo tridimensionale è disegnata. Le sue dimensioni sono misurate nelle coordinate della finestra. Ciò che è disegnato al di fuori della viewport viene scartato; tale processo viene chiamato clipping.

2.2 *Proiezione prospettica*

Il processo che avviene per la computer grafica 3D è assimilabile a quello che avviene nelle retine dei nostri occhi, vale a dire una proiezione in due dimensioni di un mondo, quello reale in questo caso, tridimensionale.

Nella storia dell'arte, i primi studi ed intuizioni sulla prospettiva vengono fatti risalire al periodo del Rinascimento.

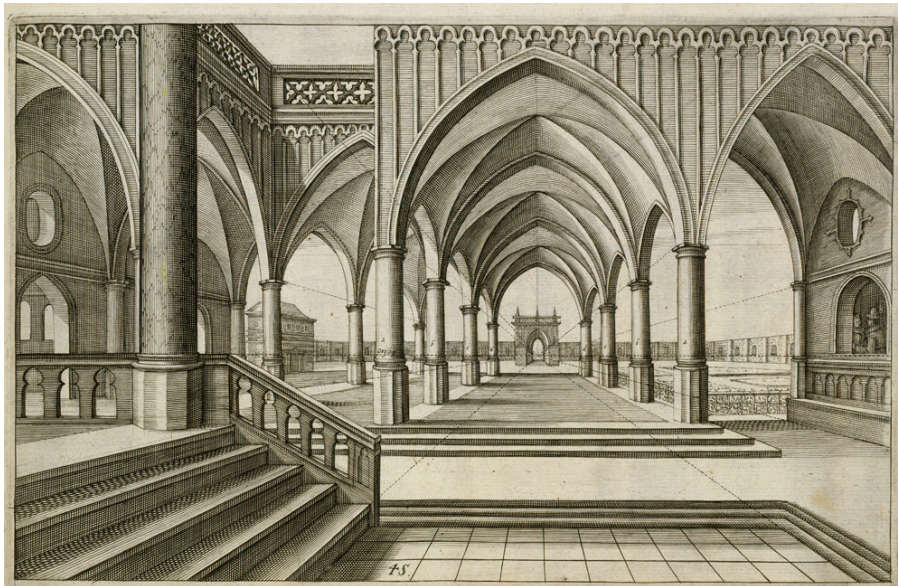


Figura 6: Jan Vredeman de Vries. Perspective

Il mondo virtuale è dunque definito in modo del tutto matematico con una serie di vertici nello spazio 3D e, tramite un processo matematico, la prospettiva viene applicata in modo da rappresentare il mondo 3D su di un piano bidimensionale. I vertici sono immagazzinati nella memoria del computer e ad essi si applicano le formule per il calcolo della prospettiva.

2.3 Prospettiva parallela

Il tipo di prospettiva più semplice dal punto di vista computazionale, anche se non corretto dal punto di vista ottico, è la prospettiva parallela.

In questo caso, di fatto la proiezione avviene sul piano XY dello spazio e si ignora la coordinata Z dei vertici degli oggetti. Il tipo di proiezione è detta parallela perché tutti i segmenti che congiungono i vertici dell'oggetto al piano di proiezione sono paralleli fra di loro.

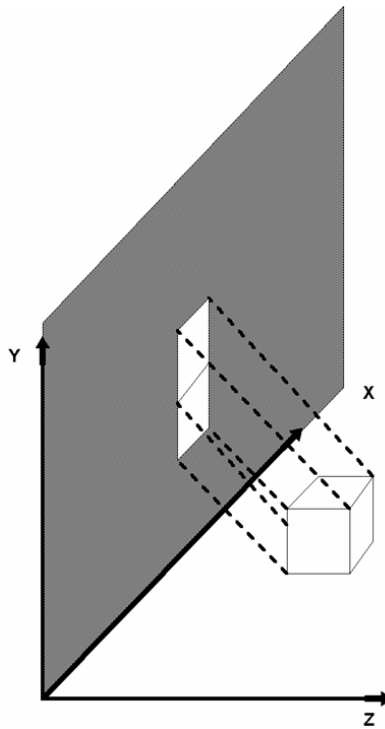


Figura 7: proiezione parallela

In questo caso le equazioni per ottenere i punti sul piano sono:

$$Px = Qx * \cos(b) + Qz * \sin(b)$$

$$Py = Qx * \sin(a) * \sin(b) + Qy * \cos(a) - Qz * \sin(a) * \cos(b)$$

Dove $P (PX , PY)$ è il punto proiettato sullo schermo, $Q (QX, QY, QZ,)$ è il punto di partenza nello spazio, a è l'angolo di rotazione sull'asse X, b è l'angolo di rotazione sull'asse Y.

In queste equazioni, per semplicità, proiettiamo sul piano xy, quindi con $z = 0$.

I risultati di questo tipo di proiezione sono una finta prospettiva, come nei disegni infantili.

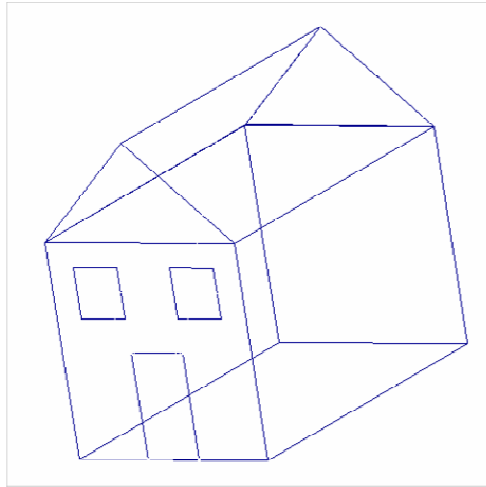


Figura 8: esempio di proiezione parallela

2.4 Proiezione prospettica

La proiezione prospettica o centrale utilizza un punto, il centro prospettico (chiamato anche punto di fuga nel disegno tecnico), che in computer grafica corrisponde anche al punto di vista dell'osservatore, per far convergere i segmenti che uniscono i vertici dell'oggetto. In questo caso i segmenti che congiungono i vertici degli oggetti (nella figura un cubo) con il piano di proiezione non sono più paralleli ma uniscono il centro prospettico con i vertici dell'oggetto.

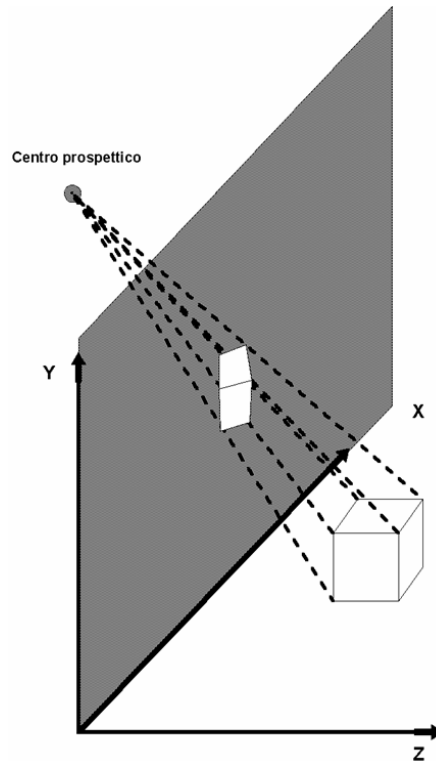


Figura 9: proiezione prospettica

Le intersezioni sul piano di proiezione tra i segmenti che congiungono i vertici dell'oggetto e il centro prospettico costituiscono la proiezione dell'oggetto in prospettiva.

Le equazioni matematiche per ottenere questo risultato sono leggermente più complesse rispetto al caso precedente:

$$P_x = \frac{Z_x * Q_z - Q_x * Z_z}{Q_z - Z_z} \quad P_y = \frac{Z_y * Q_z - Q_y * Z_z}{Q_z - Z_z}$$

Dove $P (P_x, P_y)$ è il punto proiettato sullo schermo

$Q (Q_x, Q_y, Q_z)$ è il punto di partenza nello spazio

$Z (Z_x, Z_y, Z_z)$ è il punto del centro della proiezione (punto di fuga)

I risultati di questo tipo di proiezione sono più realistici, ecco un esempio:

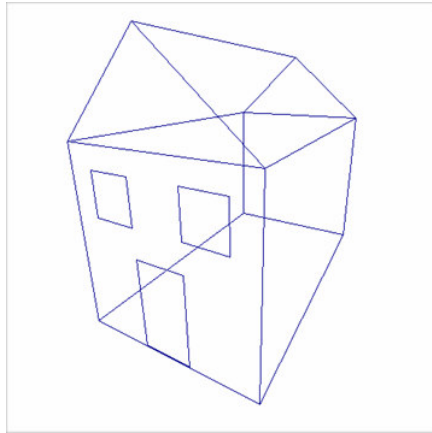


Figura 10: risultato di una proiezione prospettica

2.5 Eliminazione delle linee nascoste

Le visualizzazioni fino ad adesso proposte vengono tipicamente chiamate “wireframe” ovvero fil di ferro dal momento che gli oggetti sono rappresentati tramite i loro bordi e non è considerata l’eliminazione delle facce nascoste ovvero le facce che normalmente non sarebbero visibili da un certo punto di vista; se ad esempio consideriamo la figura 8, i bordi della parete in fondo non dovrebbero essere visibili. In letteratura esistono molti algoritmi per l’eliminazione delle linee nascoste, come ad esempio l’algoritmo del pittore, che si basa sulla tecnica effettivamente utilizzata dai pittori ovvero di disegnare prima gli elementi più lontani dall’osservatore e via via quelli più vicini che vanno a coprire gli elementi già disegnati. Questo algoritmo, seppur corretto da un punto di vista ottico, è inefficiente in quanto vengono disegnati tutti gli elementi, anche quelli che risulteranno coperti da altri.

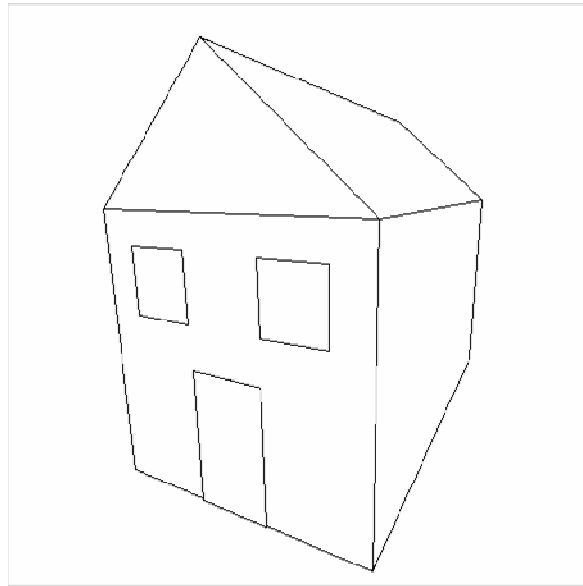


Figura 11: esempio di eliminazione delle linee nascoste

2.5.1 Z-Buffer

Quando si usano poligoni pieni anziché linee, esiste un metodo facilmente implementabile in hardware per risolvere il problema della profondità: lo Z buffer.

Questo buffer ha la stessa dimensione della viewport e memorizza il valore della profondità per ogni pixel che è stato disegnato; dove per profondità si intende la distanza dall'osservatore.

Per ogni pixel, sarà possibile andare a sovrascriverne il colore se e solo se il valore della profondità associato è minore del valore già esistente. In questo modo i poligoni più vicini all'osservatore copriranno i più lontani, nel senso che i pixel che li costituiscono si sovrappongono a quelli che costituiscono i poligoni più lontani.

2.6 Texture Mapping

La tecnica del texture mapping consente di emulare la presenza di molti dettagli senza avere altrettanti poligoni, che peggiorerebbero le prestazioni nella grafica in tempo reale.

Il texture mapping consiste nell'applicare un'immagine bitmap bidimensionale ad un poligono.

Ogni vertice di un poligono ha associate due coordinate di texturing (u,v) comprese tra 0 ed 1, che corrispondono ad un punto specifico dell'immagine.

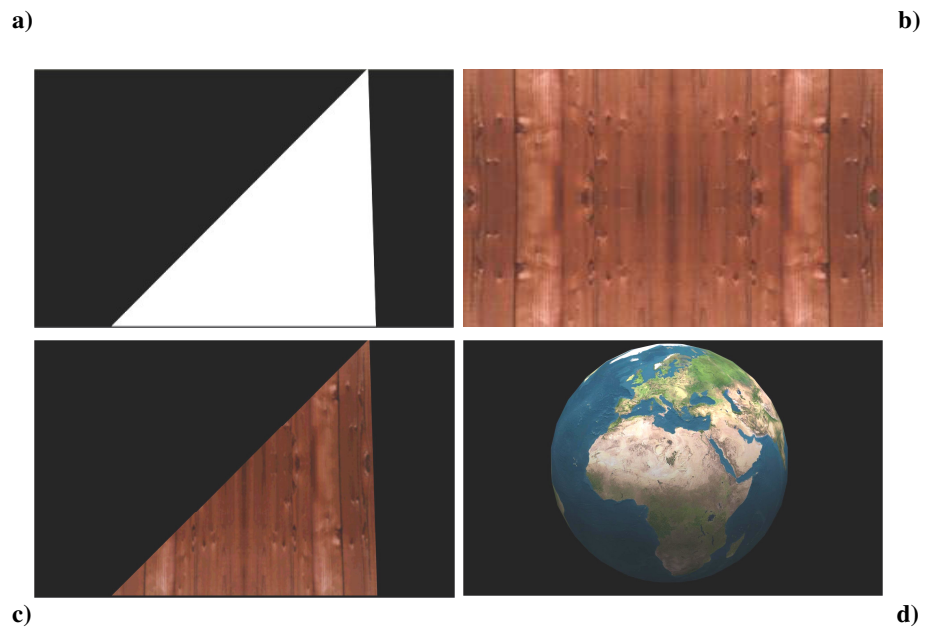


Figura 9: Texture mapping

La Fig.9 illustra la tecnica del texture mapping: la Fig.9 a) contiene un singolo triangolo, al quale viene applicata l'immagine bidimensionale della Fig.9 b); si ottiene così il triangolo con texture mapping della Fig.9 c).

2.7 Ray Tracing

Il ray tracing è una tecnica per la visualizzazione ad alta qualità, utilizzata per la resa finale di immagini. Il nome lascia intuire in quale modo essa lavori: l'algoritmo calcola il percorso dei raggi della luce dalla sorgente fino all'occhio dell'osservatore.

Per evitare di considerare raggi che escano dalla viewport e che quindi non incidono sull'immagine finale, il tracciamento dei raggi avviene in modo inverso rispetto alla realtà: parte dall'occhio a ritroso fino alla fonte luminosa. In questo modo siamo sicuri di calcolare il percorso dei soli raggi che colpiscono l'occhio.

Per ogni pixel della viewport partirà un raggio esplorativo che può incontrare gli oggetti della scena o perdersi all'infinito (in questo caso assumerà il colore dello sfondo).

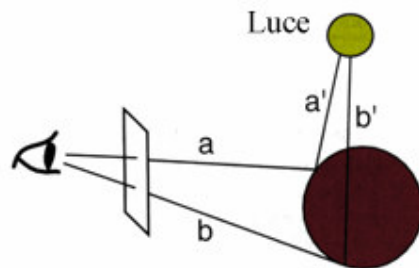


Figura 12: Ray tracing

Nell'illustrazione 10 il raggio a incontra la sfera, quindi viene fatto partire il raggio a' verso la sorgente di luce; non essendoci ostacoli tra la sfera e la luce il raggio a risulterà illuminato. Il raggio b, invece, risulterà in ombra poiché il raggio b' verso la sorgente di luce è ostacolato dalla sfera stessa.

Il colore del pixel sarà determinato dal tipo di oggetti incontrati; se, ad esempio l'oggetto colpito è di tipo riflettente un altro raggio partirà da quel

punto, secondo le leggi di riflessione della fisica; se l'oggetto è trasparente il raggio verrà rifratto secondo la legge della rifrazione, e così via.

Con il tracciamento del percorso dei raggi, fenomeni come riflessioni, rifrazioni e ombre portate, che aumentano considerevolmente la qualità della simulazione, sono facilmente ottenibili.

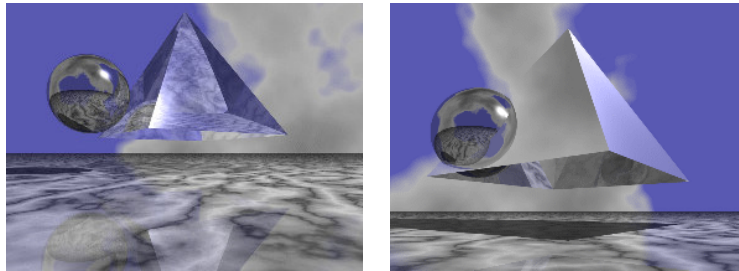


Figura 11: Esempi di immagine in ray tracing

La fig. 11 contiene due esempi di immagini ottenute tramite ray-tracing tratte dal sito internet [Pov00], esse sono state calcolate con il programma open source Pov-Ray: si possono notare le riflessioni, le rifrazioni e le ombre portate.

A causa della elevata complessità, tale algoritmo, non è utilizzabile per animazioni in tempo reale, ma solo per immagini statiche o animazioni precalcolate. Infatti nonostante l'aumento delle prestazioni delle CPU odierne, sono necessari, a seconda della complessità della scena, da alcuni minuti a qualche decina di minuti e oltre, per la generazione del singolo fotogramma, dipendentemente dalla complessità della scena.

2.8 Render to texture

Come detto nel precedente paragrafo, il ray-tracing per la sua complessità algoritmica, si presta maggiormente alla grafica prerenderizzata e non in tempo reale, quindi immagini statiche e filmati precalcolati.

Da pochi anni, però, i maggiori pacchetti commerciali ed open source di modellatori hanno a disposizione la possibilità di effettuare il cosiddetto render-to-texture, ovvero la fase di rendering che è notoriamente dispendiosa in termini di tempo di calcolo viene riversata sulle texture degli oggetti anziché finire solamente in un'immagine. Le texture originali del modello, seppur utilizzate per i materiali del rendering, vengono poi sostituite da nuove texture ottenute per rendering. Il risultato è un modello che può essere utilizzato in grafica real-time, con la qualità dell'illuminazione di un modello ray-tracing.



Figura 132: il progetto Nu.M.E. in render to texture

La figura 12 illustra un esempio di applicazione di render-to-texture: il progetto Nu.M.E., con il centro di Bologna nel 1200; si notino le ombre portate dei portici e dei palazzi sulla strada; inoltre la qualità dell'illuminazione è superiore a quella ottenibile in real-time.

Essendo le texture precalcolate, questo processo non aumenta la complessità del modello che dunque rimane perfettamente navigabile in tempo reale, se lo

era prima dell'applicazione del render-to-texture. L'unica degradazione delle prestazioni potrebbe avvenire se le nuove texture calcolate sono in numero o dimensioni molto grande, tali da occupare più memoria di quella disponibile sulla scheda grafica del PC con il quale si naviga, in questo caso la diminuzione delle prestazioni risulterebbe piuttosto sensibile.

2.9 Evoluzione dell'hardware

L'industria dei PC negli ultimi anni ha visto un'evoluzione per molti versi incredibile delle prestazioni, parallelamente ad un contenimento dei costi, che ha permesso la diffusione e l'utilizzo dell'informatica in tutti i settori dell'attività umana.

Nelle CPU e nei dispositivi di memoria di massa c'è stato un notevole sviluppo, tale da rendere vera nel tempo la famosa legge di Moore (co-fondatore di Intel), formulata nel 1965: le prestazioni dei processori, e il numero di transistor ad esso relativo, raddoppiano ogni 18 mesi.

Anche nell'ambito della computer grafica real-time si è verificato una grande evoluzione, soprattutto dalla fine degli anni Novanta, con la comparsa delle prime schede acceleratrici di grafica 3D per PC.

Prima di allora, esistevano dei mainframe per grafica tipicamente SGI (Silicon Graphics) che erano estremamente costosi e dunque potevano essere acquistati solo da enti di ricerca, simulatori di volo ad usi civili e militari, industrie automobilistiche.

Uno dei modelli di maggiore successo di SGI fu Onyx2, basato su processori MIPS RISC, e con un numero di pipeline grafiche configurabile. La pipeline grafica è il componente dedicato alla grafica 3D.

Silicon Graphics® Onyx2®



Figura 14: SGI Onyx2

Le prime schede grafiche acceleratrici per grafica 3D comparvero sul mercato nel 1997, grazie ad un'azienda, la 3DFX, che riversò il proprio know-how tecnologico guadagnato nel campo dei videogiochi per sale giochi, nel mercato dei PC.

Il primo modello si chiamava Voodoo, poteva essere utilizzato come acceleratore hardware solo a pieno schermo e non in una finestra (il che lascia intuire le finalità prettamente videoludiche del prodotto) e raggiungeva una risoluzione massima di 640 per 480 pixel. Nonostante queste limitazioni, il successo fu talmente elevato che spinse altre aziende a tentare di lanciare un prodotto in questa difficile porzione di mercato.

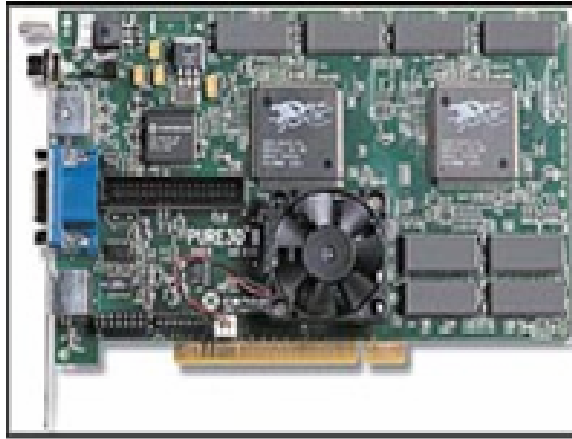


Figura 15: Scheda acceleratrice 3D 3DFX Voodoo

Negli anni successivi al 1997 ci fu una escalation nel lancio di nuovi prodotti che vedeva una nuova generazioni di chip per la grafica ogni 6/12 mesi. Questo ha portato, ad esempio, dai 16 Mbytes di memoria su scheda grafica del 1997 ai 512 Mbytes di alcune schede grafiche attuali; dai 180 milioni di pixel al secondo del 1999, 15 miliardi di alcune schede attuali.

Il maggiore traino dell'industria era, e rimane tuttora, la vendita dei videogiochi per PC. Lo sviluppo dell'hardware e l'abbassamento dei costi consente però una ricaduta positiva per la grafica 3D anche in ambiti più seri, come i beni culturali, la medicina, ecc.

2.10 Ambienti e metodi di visualizzazione

Nei primi anni Novanta, i super-computer grafici erano sovente utilizzati per pilotare sistemi di visualizzazione avanzati, quali ad esempio teatri virtuali e cave, dotati di visione stereoscopica.

2.10.1 Stereoscopia

Lo scopo della visione stereoscopica in generale è quello di fornire ai due occhi, due immagini leggermente diverse fra loro, in modo da simulare la

distanza interoculare (circa 8 cm), che nel mondo reale serve a due occhi per avere due angolazioni diverse dello stesso oggetto della visione, che poi elaborate dal cervello, danno la percezione della profondità, della distanza e una migliore sensazione della prospettiva, non a caso persone che hanno perso la vista in un occhio hanno problemi a distinguere la distanza degli oggetti.

Ci sono varie tecniche per ottenere questo obiettivo, ognuna con pro e contro; si distinguono principalmente per “stereo attivo” e “stereo passivo”.

Nello stereo attivo, gli occhiali hanno lenti a cristalli liquidi che sono in grado di otturarsi (e dunque impedire la visione) e ritornare alla normalità in tempi brevissimi, con frequenze intorno ai 50 Hz.

Gli occhiali sono sincronizzati con il monitor (o lo schermo dove sono proiettate le immagini) tramite cavo o wireless (emitter di segnali infrarossi), dove si alternano le immagini destinate all’occhio destro con quelle destinate all’occhio sinistro.



Figura 16: occhiali per stereo attivo

Dunque se l’immagine sullo schermo sarà destinata all’occhio sinistro ad esempio, l’occhio destro sarà otturato dalla lente, e viceversa.

Il principale vantaggio di questa soluzione per lo stereo è proprio la netta sensazione di profondità percepita, gli svantaggi sono un leggero imbrunimento dei colori (dovuto al continuo inscurimento delle lenti per

l'otturazione) e una maggiore propensione alla nausea per lo sfarfallio dovuto all'alternanza dell'otturazione.

Un altro sistema, cosiddetto stereo passivo, funziona tramite un altro principio, la polarizzazione della luce. Gli occhiali vengono detti passivi perché non otturano gli occhi, ma filtrano le immagini a seconda della loro polarizzazione. In questo caso le immagini destinate all'occhio destro e sinistro vengono proiettate contemporaneamente sullo schermo, saranno le lenti a filtrarle e fare in modo che gli occhi ricevano l'immagine idonea. I vantaggi dello stereo passivo sono: occhiali più leggeri e meno ingombranti (non hanno bisogno dell'elettronica per lo switching dell'otturazione), minore possibilità di nausea; lo svantaggio è costituito dal fatto che questa tecnologia è utilizzabile solo con sistemi di proiezione, non con monitor CRT, in quanto occorre polarizzare il fascio di luce proveniente dalla sorgente.

2.11 Sistemi immersivi di visualizzazione

Un sistema di visualizzazione immersivo, piuttosto diffuso nei centri di calcolo in tutto il mondo è il teatro virtuale; è composto da uno schermo semicilindrico, in fibra di vetro, e copre un'angolazione di circa 120°.



Figura 17: teatro virtuale

Lo schermo è suddiviso in tre settori, dunque tre proiettori, nel caso di stereo attivo, oppure tre paia di proiettori sovrapposti, nel caso di stereo passivo, sono utilizzati per la proiezione.

In questo tipo di installazione, lo spettatore è immerso nell'ambiente virtuale, anche la coda dell'occhio è coperta dalla visuale, essendo lo schermo semi-cilindrico, dunque il senso di immersione è molto buono; è però assente l'interattività, in quanto una persona alla console controlla il movimento virtuale per tutti gli spettatori.

2.11.1 Cave

Il cave è un'installazione a cubo, nel quale lo spettatore entra, munito di occhiali per la stereoscopia, per un'immersione totale nell'ambiente virtuale, in qualsiasi direzione volga il proprio sguardo, tutte le sei facce ricevono la proiezione dell'ambiente virtuale.

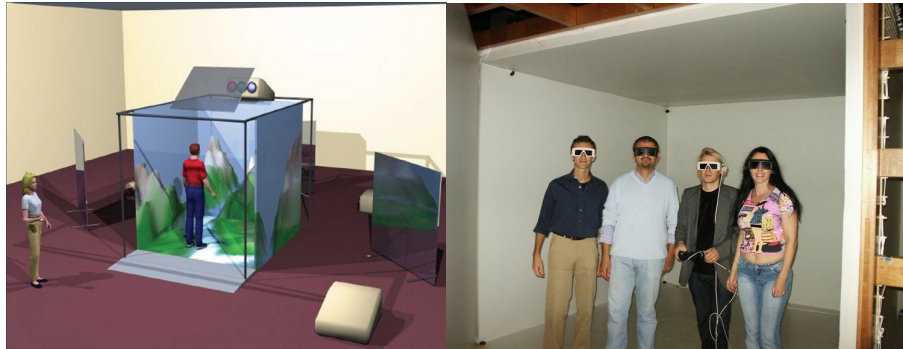


Figura 18: cave

Degli spettatori, solo uno guida la navigazione virtuale tramite degli occhiali dotati di sensori che rilevano la posizione e la direzione dello sguardo. Il sistema supporta anche la stereoscopia.

Il maggiore svantaggio di questo sistema è il notevole costo di installazione e manutenzione.

2.12 Le maggiori librerie grafiche di base

2.12.1 OpenGL

OpenGL² è un'interfaccia all'hardware grafico, che consiste di circa 250 funzioni, le quali permettono di produrre grafica 3D in tempo reale.

Il nome della libreria fa prefigurare gli scopi dei suoi creatori: una libreria grafica aperta: infatti è disponibile su moltissime piattaforme, dai super computer di Silicon Graphics ai PC sia con Windows che con Linux, su MacIntosh, Sun, ecc.

OpenGL è stata realizzata in modo da poter essere hardware indipendente; per raggiungere questo obiettivo non esistono comandi per la gestione delle finestre e degli eventi di input, che dovranno essere gestiti nel particolare sistema operativo e nel particolare hardware in cui si lavora.

OpenGL non contempla un'organizzazione dei modelli tridimensionali, l'intero mondo virtuale è specificato con una serie di primitive (punti, linee, poligoni), ed è forse per questo basso livello di controllo del processo grafico e per la sua proceduralità nell'istanziare i comandi che da molti è considerata l'assembler della computer grafica.

OpenGL è una libreria grafica sviluppata da Silicon Graphics, che ha grandemente beneficiato dell'esperienza maturata con Iris GL, sempre di Silicon Graphics che era stata usata nei primi anni di attività. OpenGL quindi nasce già matura ed è forse questo il motivo principale del suo successo.

Inoltre produttori di hardware con funzionalità particolari possono utilizzare il meccanismo delle estensioni di OpenGL.

Lo sviluppo non è legato alle direttive di Silicon Graphics, in quanto viene deciso da un consiglio di varie aziende, l'ARB (Architecture Review Board), che comprende tra gli altri, Intel, Silicon Graphics, Microsoft, IBM, Hewlett-Packard, Compaq, ecc., che periodicamente si riunisce.

² <http://www.opengl.org>

OpenGL è particolarmente usata in ambiente CAD, come visualizzatore 3D dei progetti.

2.12.2 Direct 3D

Direct 3D è una libreria proprietaria Microsoft e pertanto disponibile solo su sistemi operativi Windows, fa parte dell'insieme di tool multimediali DirectX³.

E' nata soprattutto per il mercato dei videogiochi 3D, per facilitarne la programmazione; infatti prima della sua introduzione gli sviluppatori erano costretti a scrivere codice specifico per i vari tipi di hardware grafico disponibile sul mercato (o perlomeno i più diffusi o performanti).

Con Direct 3D il programmatore scrive codice generico, sarà poi il driver della scheda grafica, fornito direttamente dal costruttore dell'hardware, che avrà il compito di sfruttare al massimo il chip grafico. Se una certa funzionalità non fosse supportata in hardware, verrà, se possibile, emulata via software automaticamente (con ovvio calo di prestazioni).

2.13 Le librerie basate su Scene Graph

A partire da OpenGL, libreria grafica di basso livello, molto vicina all'hardware grafico, sono state sviluppate molte librerie che si basano su un'organizzazione a scene graph; vedremo in particolare OpenGL Performer di SGI per la versione su PC locale di NUME (visualizzabile anche nel teatro virtuale del Cineca) e Open Scene Graph per la versione Web.

La differenza principale tra le librerie grafiche di basso livello come OpenGL e Direct3D e altre librerie basate su Scene Graph è il modo in cui queste ultime strutturano ciò che deve essere visualizzato: in OpenGL tutti gli oggetti vengono disegnati tramite le primitive grafiche, in Performer esiste il concetto di scena che è l'insieme di tutto quanto va a comporre il "mondo virtuale" che

³ <http://msdn.microsoft.com/directx/>

si vuole simulare, quindi oggetti 3D (composti da un numero arbitrario di poligoni), con tutti gli effetti che vogliamo inserire (texture mapping, nebbia, trasparenza, ecc.), sorgenti di luce, nodi di raggruppamento, livelli di dettaglio, ecc.

Tutta la scena deve essere inserita in un apposito albero (Scene graph), che ha un nodo radice, i successivi nodi devono essere figli del nodo radice.

Lo Scene graph può essere modificato in qualsiasi momento della simulazione aggiungendo, togliendo o modificando qualsiasi nodo. Una schematizzazione dello Scene Graph si può osservare nella Fig 17.

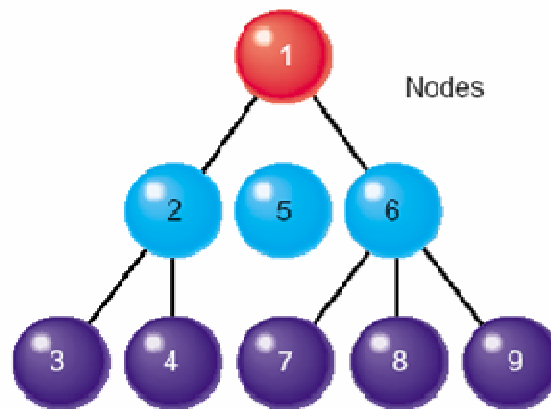


Figura 19: Scene Graph

2.13.1 OpenGL Performer

OpenGL Performer è una libreria grafica, basata su OpenGL, proprietaria di Silicon Graphics, nata principalmente per le workstation grafiche da essa prodotte, ma successivamente introdotta anche per PC in ambiente Linux e Microsoft Windows.

Fu introdotto nel 1993 ed è possibile trovare un articolo di presentazione in [Rol93]; questo articolo riporta la presentazione di Performer al Siggraph del 1993.

L'obiettivo principale della libreria è di massimizzare le prestazioni delle applicazioni real-time e di fornire allo sviluppatore una organizzazione maggiore rispetto all'approccio strettamente procedurale di OpenGL, in modo da facilitare lo sviluppo.

Per il rendering (resa finale) Performer si interfaccia a OpenGL o a Iris GL (sui vecchi modelli Silicon Graphics), con bindings C o C++.

All'interno dell'applicazione Performer si possono anche chiamare funzioni OpenGL se un tale controllo fosse necessario.

La libreria è liberamente scaricabile dal sito web <http://www.sgi.com/software/performer>, in versione free, che è pienamente funzionale, ma il rendering presenterà una scritta "promozionale" nell'angolo in alto a destra. Per eliminare questa scritta, occorre pagare una licenza per ogni PC che si utilizza con la libreria.

2.13.2 Open Scene Graph

Open Scene Graph è anch'essa una libreria basata su Scene Graph, disponibile su moltissime piattaforme IRIX, Linux, Windows, FreeBSD, Mac OSX, Solaris, HP-UX, e persino PlayStation2.

La sostanziale differenza è nella licenza: Open Scene Graph è, come si può immaginare dal nome, completamente open source, ovvero il suo codice sorgente è disponibile in rete e scaricabile interamente, mentre di Performer sono disponibili solo alcune parti del codice, ma non le parti più importanti. Il sito web di riferimento è www.openscenegraph.org

I due fondatori erano ex-dipendenti SGI, hanno iniziato a scrivere Open Scene Graph come porting free di Performer, dunque le due librerie si somigliano moltissimo dal punto di vista tecnico, con la sola eccezione che Open Scene Graph è programmabile solo ad oggetti, dunque solo con il linguaggio C++ e non C.

Uno dei possibili utilizzi di questa libreria, essendo completamente open source è la programmazione di un plug-in per i maggiori browser web; dunque

si apre la possibilità di un utilizzo della grafica 3D su Internet che permette sicuramente una maggiore diffusione dei contenuti informativi dei progetti legati alle ricostruzioni virtuali.

Open Scene Graph può contare su un vasto numero di programmatori iscritti alla propria mailing-list, quindi si tratta di un progetto open source molto vivace e portato avanti con dedizione da una comunità di sviluppatori volontari.

2.14 Utilizzo e vantaggi di una libreria a Scene Graph

Lo Scene Graph può essere visto come un albero informatico (ricordiamo che un albero in informatica è una struttura dati che al contrario degli alberi “reali”, ha il nodo radice in alto e i nodi foglia in basso), che ha una radice e una serie di nodi di raggruppamento o di comportamento che agiscono sui nodi che contengono la geometria vera e propria, che si trovano in basso, ovvero non hanno nodi figli.

Ci sono molti tipi di nodo, che intervengono sui nodi che contengono geometria; in generale vale la regola dell’ereditarietà dei nodi, ovvero le proprietà di un qualsiasi nodo si applicano ai nodi figli e ai figli di questi fino ad arrivare ai nodi foglia.

Vediamo brevemente i più comuni tipi di nodi:

- Nodi di tipo gruppo: questi nodi servono per raggruppare altri nodi, possono avere un numero arbitrario di figli, di qualsiasi tipo. Si possono assegnare dei nomi ai nodi gruppo, questa proprietà è stata largamente utilizzata dal visualizzatore del progetto Nume, Visman, per riconoscere nodi (che originariamente erano di tipo gruppo) con nomi particolari ed attivare azioni collegate al nodo, come ad esempio collegamento a database, ecc. Torneremo su questo punto più avanti nei prossimi capitoli.
- Nodo Switch: il nodo switch è simile al nodo gruppo nel senso che può avere un numero illimitato di nodi figli, ma al contrario del nodo

gruppo, uno solo di essi sarà abilitato, e dunque visibile, in un determinato momento. Questo nodo è molto utile per far scarire e contemporaneamente far apparire sulla scena virtuale un oggetto.



Figura 20: esempio di utilizzo di un nodo switch nella scena

- Nodi con trasformazioni geometriche: questi nodi (che assumono nomi leggermente diversi tra Performer e Open Scene Graph, ma di fatto si equivalgono) implementano tre tipi di trasformazioni: traslazioni, rotazioni e scale nello spazio. Queste trasformazioni si applicano a tutti i nodi figli e per ereditarietà a tutta la discendenza.
- Nodi LOD (Level of Detail): questi nodi si comportano in modo simile agli Switch, ovvero possono avere un numero indefinito di figli ed uno solo attivo in un determinato istante; la differenza è che il cambiamento non è manuale ma bensì automatico, in base alla distanza dell'osservatore e alle prestazioni (fotogrammi al secondo) dell'applicazione in quel momento. Questo nodo serve per avere maggiori prestazioni nella grafica in tempo reale: si possono avere diverse versioni di uno stesso oggetto, con più o meno dettagli. Quando l'osservatore sarà sufficientemente vicino da poter apprezzare i maggiori dettagli, sarà visualizzato il nodo figlio più dettagliato, viceversa quando l'osservatore si allontanerà dall'oggetto, sarà mostrata la versione con minor numero di dettagli per aumentare le prestazioni dell'applicazione.

- Nodi Sequence: anche questi nodi derivano dai nodi Switch e sono utili per costruire delle animazioni. I nodi figli del nodo Sequenze costituiranno gli elementi dell'animazione (ad esempio un uomo che cammina) e solo uno di essi è visualizzato alla volta; il nodo sarà sostituito dal nodo figlio successivo con un tempo variabile, che deve essere impostato dall'applicazione. Al completamento della sequenza, essa può fermarsi oppure ripartire dall'inizio.
- Nodi Billboard: questi nodi sono stati implementati per simulare la presenza di oggetti di alta complessità mediante l'utilizzo di un solo quadrilatero e una texture applicata ad esso. Il quadrilatero viene automaticamente ruotato dall'applicazione in modo da voltarsi sempre verso il punto di vista dell'osservatore, in qualsiasi punto esso si trovi. Viene spesso utilizzato per inserire elementi complessi come alberi, persone, statue nella scena.

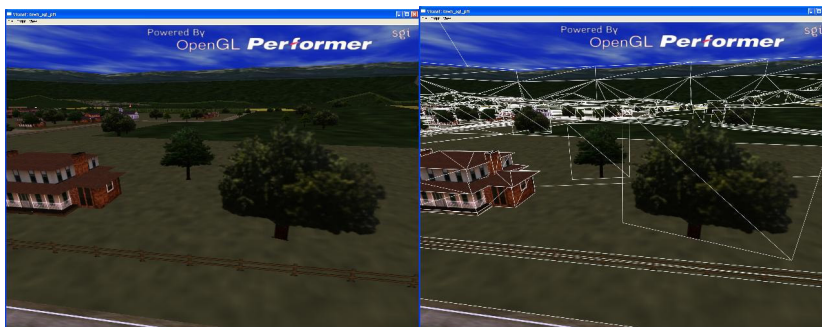


Figura 21: esempio di utilizzo di nodi billboards per alberi

- Nodi Image Based Render: questi nodi sono un'estensione dei precedenti. Dunque sono composti da un unico quadrilatero, però anziché avere un'unica texture, ne possono avere un numero indefinito; le texture applicate vengono cambiate a seconda dell'angolo con il quale l'osservatore guarda il nodo IBR. In questo modo è possibile simulare in modo più efficace oggetti non simmetrici, come in particolare statue e persone, di cui si possono immettere texture ottenute da campagna fotografica da varie angolazioni.

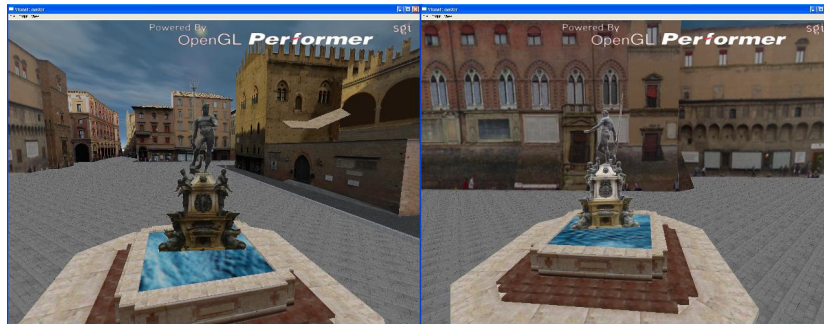


Figura 22: esempio di utilizzo di nodi IBR nella scena

2.14.1 Caricamento da file di nodi scene graph

Uno dei maggiori vantaggi dell'utilizzo di librerie scene graph rispetto a librerie di basso livello come OpenGL o DirectX consiste nel poter caricare da file oggetti 3D. Tipicamente sono supportati i maggiori formati commerciali disponibili come 3DS (3D Studio Max), OpenFlight (Multigen Creator ed altri), dxf (Autocad ed altri), VRML, ecc.

Il file caricato andrà ad integrarsi sullo scene graph e potrà essere immediatamente visualizzato.

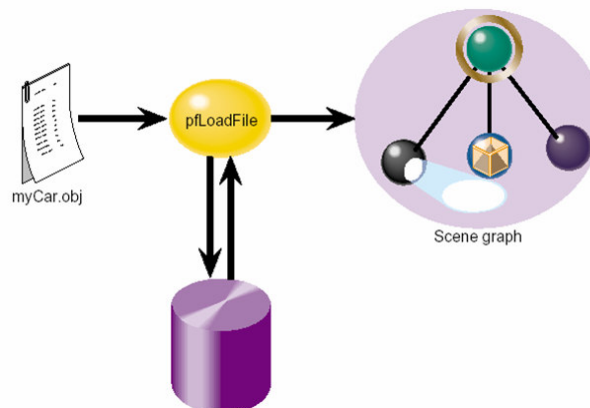


Figura 23: caricamento di un file sullo scene graph

Se il file conteneva una struttura gerarchica in stile scene graph, essa viene mantenuta nel caricamento. Dunque si può impostare una gerarchia ad hoc con un modellatore 3D in grado di editare uno scene graph (parleremo di questo nel prossimo capitolo) e fare in modo che il visualizzatore carichi uno scene graph particolare, con nodi che assumono, per convenzione, dei significati specifici.

3 Applicazione della computer grafica ai beni culturali

3.1 Esigenze peculiari di questo ambito

L'applicazione della grafica 3D ai beni culturali ed alla storia presenta delle esigenze particolari, che si differenziano ad esempio dalla visualizzazione scientifica classica, per la quale sono nati i primi sistemi di visualizzazione di cui abbiamo discusso nel capitolo precedente.

Il concetto stesso di virtualità implica la disponibilità di visualizzazioni tridimensionali e di sistemi interattivi finalizzati alla creazione di ambienti immersivi generati in tempo reale dal calcolatore⁴. Oggi è possibile fare esperienza di sistemi virtuali anche per generare, navigare ed esplorare ambienti ricostruiti d'interesse storico-culturale; in particolare nel campo dell'archeologia l'esperienza è oramai decennale⁵.

Con l'uso di queste nuove tecnologie si possono visualizzare e presentare “ricostruzioni” di ambienti del passato, comprendenti edifici, paesaggi ed artefatti⁶.

Da questo punto di vista le applicazioni di Realtà Virtuale⁷ sono strettamente collegate a quelle di Visualizzazione Scientifica⁸ perché creano modalità

⁴ Lanier J., Interview in “Whole Earth Review”, 1988.

⁵ Reilly P., *Towards a Virtual Archaeology*, in K. Lockyear and S. Rahtz, *CAA90: Computer Applications and Quantitative Methods in Archaeology 1990*, BAR International Series, 565, Tempvs Reparatum, Oxford, 1991, pp. 133–139. Individua le linee guida di quest'ambito di ricerca: la sua visione iniziale si articola sulla registrazione di scavi e sulle possibilità offerte dalla ripetizione virtuale degli scavi stessi grazie all'impiego di tecnologie come gli ipertesti, la multimedialità e la modellazione tridimensionale di solidi.

⁶ Ryan N., *Documenting and Validating Virtual Archaeology*, in CAA 2002, “The Digital Heritage of Archaeology” - Proceedings, 2-6 April, Heraklion, Crete, Greece, 2002.

⁷ Vince J., *Virtual Reality Systems*, Addison-Wesley, Wokingham, 1995; Card S. K., Mackinlay J., Shneiderman B., *Readings in Information Visualization – Using Vision to Think*, Morgan Kaufman Publishers, 1999.

⁸ McCormick B.H., De Fanti T.A., Brown M.D., *Visualization in Scientific Computing Computer Graphics*, ACM SIGGRAPH, Vol. 21, n. 6, 1987; Marmo R., Valle M., Zannoni C., *Introduzione alla Visualizzazione Scientifica*. Editrice Il Rostro, 2005.

originali di navigazione ed interrogazione di mondi visibili, invisibili, ipotetici ed immaginari.

È necessaria, al fine di ottenere un risultato che sia visivamente appagante, ma anche storicamente valido, una stretta collaborazione tra informatici, storici e modellatori 3D, all'insegna della massima interdisciplinarietà.

Il potenziale di questo tipo di sinergia è notevole, se le professionalità in campo superano i preconcetti, si può arrivare ad un nuovo media per la comunicazione della storia e dei beni culturali, che può come corollario generare nuovi tipi di professionalità per archeologi e divulgatori per un'archeologia virtuale, da affiancare alle modalità di sistematizzazione e divulgazione tradizionali del sapere scientifico in ambito storico e archeologico, come il classico saggio in formato cartaceo, per integrarlo con nuove modalità, nuovi spazi di conoscenza.

Le visualizzazioni scientifiche collegate a ricostruzioni di ambienti virtuali possono ampliare le nostre capacità percettive e consentire l'interazione tra modelli numerici e dati empirici.

Le applicazioni grafiche rendono più agevole la comprensione di costruzioni numeriche complesse attraverso una sintesi mediata dalle immagini, creando modalità originali per navigare in Internet e per interrogare mondi visibili, invisibili, ipotetici e immaginari.

L'insieme di questi elementi è alla base della realizzazione di Ambienti Virtuali che sono, in effetti, sistemi complessi.

Per la realizzazione di ambienti virtuali è indispensabile:

- saper costruire uno scenario digitale interattivo a più livelli di dettaglio (Multilod, multi level of details⁹);

⁹ Luebke D., Watson B., Cohen J. D., Reddy M., Varshney A., *Level of Detail for 3D Graphics*, The Morgan Kaufmann Series in Computer Graphics, San Francisco, California; London, Morgan Kaufmann, 2003.

- ottenere la validazione e la certificazione per le relative fonti o informazioni digitali;
- ottenere la validazione e la certificazione per inserire eventuali simulazioni numeriche ;
- disegnare un'interfaccia interattiva in “collaborazione” con i potenziali fruitori lettori.

Prodotti fondati su Ambienti Virtuali interattivi non solo mantengono nella loro integrità le informazioni disponibili nell'ambito della comunità scientifica, ma le conservano efficacemente e le mettono infine a disposizione in una modalità più amichevole.

Per questo, gli ambienti virtuali sono una valida alternativa sia alla necessità di semplificare molte comunicazioni, sia ad una certa passività imposta da molti media.

Tutto ciò consente la navigazione all'interno d'ambienti sintetici (Virtual Environments - VE). Un'agevole interazione e navigazione si ottiene con una modellazione ottimizzata per il real time disegnando al calcolatore la geometria dello scenario da ricostruire. In generale i singoli oggetti sono costituiti da insiemi di poligoni, occorre poi definire colori, materiali ed infine disegnare digitalmente le texture, le immagini cioè, che rivestono le geometrie dei modelli al fine ottenere un effetto fotorealistico. I modelli vanno poi prodotti secondo diversi livelli di dettaglio al fine di ottimizzare la visualizzazione in tempo reale e per consentire di fare emergere la complessità degli scenari.

E la differenza da tutti gli altri media è qui. Gli ambienti virtuali interattivi sono spazi non sequenziali come il cinema ma un medium non lineare l'autore VR può decidere un percorso guidato o lasciare libero l'utente di navigare, muoversi, immergersi dentro lo scenario ricostruito fino al punto di approfondimento desiderato o possibile. Dovrà decidere quindi il livello di interazione con lo scenario, le modalità possibili di navigazione, tour

automatici, punti di vista particolari nonché la gestione di diversi dispositivi d'input.

L'esperienza di lavoro ormai più che decennale ha messo in luce l'importanza che va riposta nell'organizzazione e gestione del patrimonio dei mondi digitali creati e di tutti i dati ad essi connessi. Occorre rendere visibili i riferimenti relativi agli autori dei modelli, al patrimonio delle fonti utilizzate per le ricostruzioni ed illustrare le procedure di validazione dei modelli stessi. È necessario definire le convenzioni grafiche utilizzate per indicare ciò che non può essere modellato per mancanza di informazioni. Il tutto deve esaltare lo spessore dei contenuti e travalicare l'aspetto "spettacolare-televisivo" dello scenario ricostruito.

3.2 L'esperienza Visman

Non esistendo un software commerciale esplicitamente pensato per applicazioni sui beni culturali, che consenta non solo la navigazione dei modelli 3D, ma anche la possibilità di estrapolare informazioni dagli oggetti che compongono la scena, CINECA con il supporto iniziale del consorzio Spinner¹⁰ ha sviluppato un framework ad hoc per colmare questa lacuna..

Una notevole innovazione della fruizione di ricostruzioni grafiche consiste infatti nell'introduzione del collegamento tra oggetti 3D e link a database relazionali (Access, Oracle, SQL Server, Sqlite ecc.) in modo da aprire una nuova frontiera di conoscenza tramite l'accesso a tutte le informazioni disponibili e catalogate per ogni particolare oggetto ricostruito in 3D.

Queste informazioni possono, a questo punto, essere dei tipi più svariati, come testo, immagini, filmati, suoni, ecc.

Per un software di visualizzazione applicato ai beni culturali e al paesaggio si delinea la necessità di avere questo tipo di informazioni, altrimenti la

¹⁰ <http://www.spinner.it/>

ricostruzione risulterebbe priva di contenuti divulgativi, potendo contare sulla sola ricostruzione 3D dei modelli.

Inizialmente fu scelta la libreria grafica OpenGL Performer, poiché al Cineca, dove era installato un sistema SGI collegato al teatro virtuale, Performer era già incluso nei pacchetti standard installati insieme al sistema operativo. Nel 2000 Performer fu convertito su PC dapprima con sistema operativo Linux, poi nel 2001 su sistema operativo Windows, aprendo dunque l'utilizzo di questa libreria al mondo dei PC, oltre ai super-computer grafici di produzione SGI con sistema operativo IRIX (lo Unix di SGI).

Performer non è stato pubblicato da SGI su PC con licenza open source, ma con una licenza commerciale, comunque sufficientemente flessibile: si può scaricare gratuitamente da Internet la versione free che è completamente funzionale, ma presenta un logo pubblicitario nell'angolo in alto a destra. Per eliminare la scritta, occorre acquistare una licenza per ogni PC dove si utilizza il software. Di alcune parti del codice è disponibile il sorgente dal sito Internet, ma si tratta di parti del codice secondarie, non del nucleo della libreria.

Utilizzando il linguaggio di programmazione C++, ed organizzando il codice in classi, è stato possibile isolare funzionalmente il codice, in modo che le classi che implementano le funzionalità del software fossero del tutto modulari ed intercambiabili.

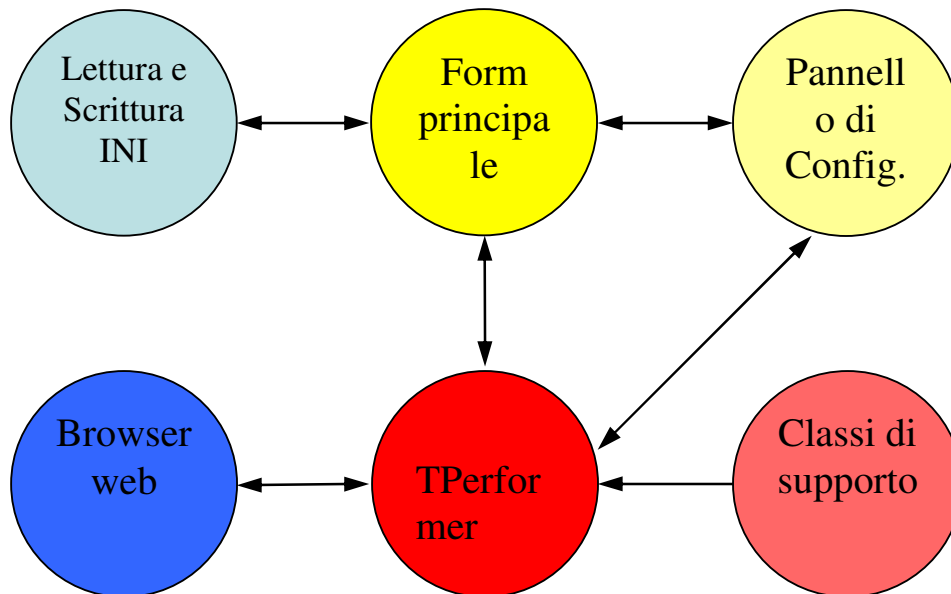


Figura 24: schema delle classi C++ di Visman

La prima versione era programmata con l'ambiente di sviluppo Borland C++ Builder, una seconda versione, che utilizza la libreria open source wxWidgets¹¹ per la GUI (interfaccia grafica), fu realizzata cambiando solo le classi che si occupavano di GUI e di lettura/scrittura dei files di configurazione, senza cambiare le classi che si occupavano di 3D e di scene graph. Questo porting si rese necessario per avere una versione Linux di Visman, affinché potesse girare sulla macchina SGI Altix del teatro virtuale del Cineca.

Dal punto di vista utente e non tecnico, Visman ha le seguenti caratteristiche:

- Navigazione intuitiva, “simile ad un videogioco”
- Gestione delle collisioni con il mondo 3D
- Connessione tra oggetti 3D della scena e database relazionali
 - Apertura file multimediali: Html, filmati, immagini
 - Visualizzazione dei dati con form personalizzate

¹¹ www.wxwidgets.org

- Hint di colore diverso per l'identificazione dei diversi tipi di nodi
- Lettura e scrittura della configurazione da file INI

Riguardo al primo punto, la maggioranza dei navigatori disponibili sul mercato offriva delle possibilità di movimento tipiche del progettista e non del “turista storico”; cioè i tipici movimenti offerti da Performer (ed anche da Open Scene Graph) sono il movimento Trackball, che consente di ruotare un oggetto, avvicinarsi o allontanarsi; oppure altri tipi di movimento sono Fly, per spostarsi in modalità “volo d’uccello” sopra al proprio mondo 3D ed infine Drive per guidare nella scena virtuale.

Considerando questi tipi di movimento innaturali, una delle prime funzionalità sulle quali si è concentrato lo sforzo fu di replicare il movimento in prima persona di un visitatore che cammina virtualmente all’interno di una scena 3D, con l’attivazione delle collisioni col mondo 3D, rendendo impossibile ad esempio attraversare muri ed oggetti, e facendo rimanere la distanza da terra costante, anche in presenza di dislivelli e scalinate.

Questo tipo di navigazione permette una migliore immersione dell’utente nell’ambiente virtuale, consentendo una immedesimazione nella scena 3D.

3.3 Lo Scengraph come organizzazione della scena collegabile alla logica del modello

Oltre alla navigazione e alla possibilità di leggere/scrivere la configurazione su file (punto di vista iniziale, colore del cielo, presenza ed intensità di nebbia, ecc.), una delle principali caratteristiche di Visman è la scansione dello scene graph, per riconoscere nodi “speciali” e la possibilità di picking degli oggetti (Vedi paragrafo 3.3.2).

L'accorgimento generale per questo tipo di operazioni è il "etichettare" i nodi dello scene graph attraverso dei nomi particolari che siano contrassegnati da speciali prefissi.

Il nodo tipicamente utilizzato per queste operazioni è un semplice nodo di tipo gruppo, al quale viene assegnato un nome particolare che risponde alla forma generica:

PREFISSO_numero-incrementale

Dove prefisso identifica il tipo di nodo speciale, che può essere un nodo di tipo collegamento a database, un nodo di tipo switch, Image Based Render, Pview ecc.

Vediamo con maggiore dettaglio i nodi particolari che l'applicazione Visman è in grado di riconoscere:

- Nodo di tipo collegamento a database: in questo caso il prefisso è variabile, per permettere potenzialmente il collegamento a database diversi; il prefisso viene letto insieme al resto della configurazione da un file (che è in formato INI). Esempi di questo tipo di nodi sono DB_30, oppure T_1140. Questo nodo viene attivato attraverso il picking (di cui si parlerà più diffusamente in seguito), ovvero il click dell'utente su un oggetto tridimensionale, che viene riconosciuto dall'applicazione; se il nodo ha un nome il cui prefisso corrisponde a quello assegnato per il collegamento a database, viene estratto il numero incrementale e viene effettuata la query sul database; se un record corrispondente viene trovato, possono essere letti i restanti campi che possono contenere informazioni sull'oggetto 3D cliccato dall'utente, come ad esempio testo, immagini, siti web, ecc. che si possono mostrare all'utente come chiarimenti ed approfondimenti alla ricostruzione 3D.

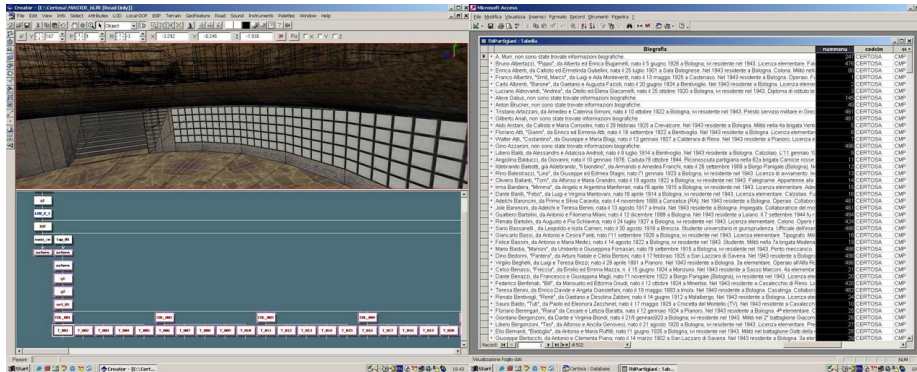


Figura 25: l'immagine sulla sinistra mostra lo scene graph dell'oggetto e i prefissi con i numeri incrementali dati agli oggetti; la figura sulla destra mostra la colonna della tabella corrispondente ai numeri impostati per gli oggetti 3D

- Nodo di tipo switch: il nodo nello scene graph iniziale è in realtà un semplice nodo gruppo, che viene riconosciuto durante la fase di scansione dello scene graph dal suo nome, “SWT_numero incrementale” (vedasi paragrafo 3.3.1); una volta riconosciuto, l'applicazione crea un vero nodo switch, nel quale vengono copiati tutti i nodi figli del nodo originario, che, finita questa operazione, viene cancellato. Per controllare il nodo switch viene creata un'apposita GUI (interfaccia grafica) in un pannello di controllo, la quale può avere delle etichette apposite per il progetto, che vengono lette dal file di configurazione durante la lettura del modello. I modelli che fanno parte del nodo switch vengono cambiati con un effetto di scalatura sull'asse Z, con una compressione del modello che deve essere tolto dalla scena e un'espansione del modello che deve essere inserito nella scena.

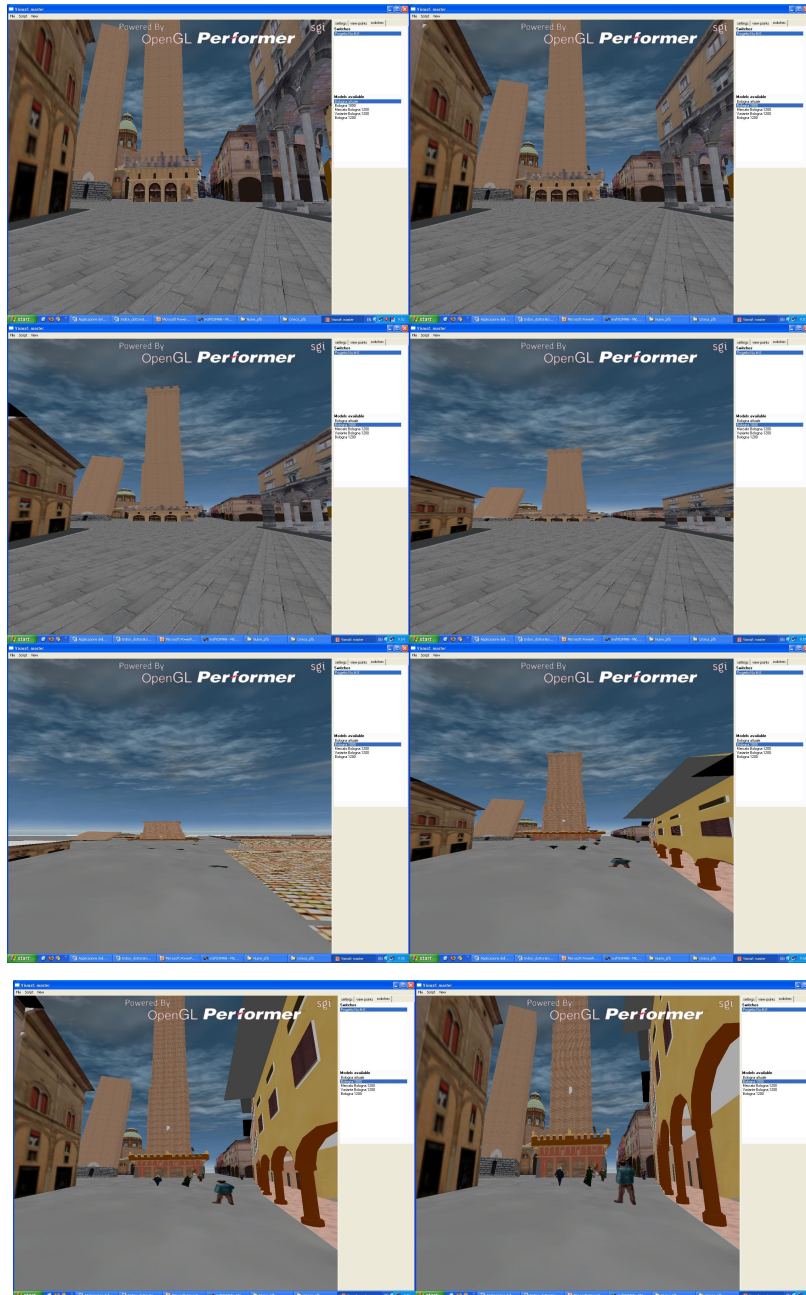


Figura 26: Sequenza di immagini che mostra l'effetto di scalatura sull'asse Z per gli switch e il relativo pannello di controllo

- Nodo di tipo story: il nodo story a differenza del nodo switch è stato pensato non per visualizzare differenze nel tempo, ma per comunicare

visivamente aggiunte, tipicamente in ambito architettonico, che si sono verificate nel tempo. Anche questo nodo sullo scene graph originario è un nodo gruppo, con un nome particolare “STORY_numero incrementale”, che una volta riconosciuto dall’applicazione durante la scansione viene svuotato di tutti i nodi figli, che vengono spostati in un altro nodo gruppo di appoggio (il quale non è attaccato allo scene graph principale, dunque i suoi figli non vengono disegnati sullo schermo); viene anche in questo caso creata un’apposita interfaccia grafica, uno slider a scorrimento; mano a mano che l’utente sposta lo slider, i nodi dal gruppo di appoggio vengono spostati nel nodo originario, in modo da essere disegnati dalla libreria grafica. Se l’utente muove lo slider verso l’inizio, accadrà il processo inverso, ovvero i nodi vengono tolti dal gruppo visibile e tornano invisibili, nel gruppo di appoggio.

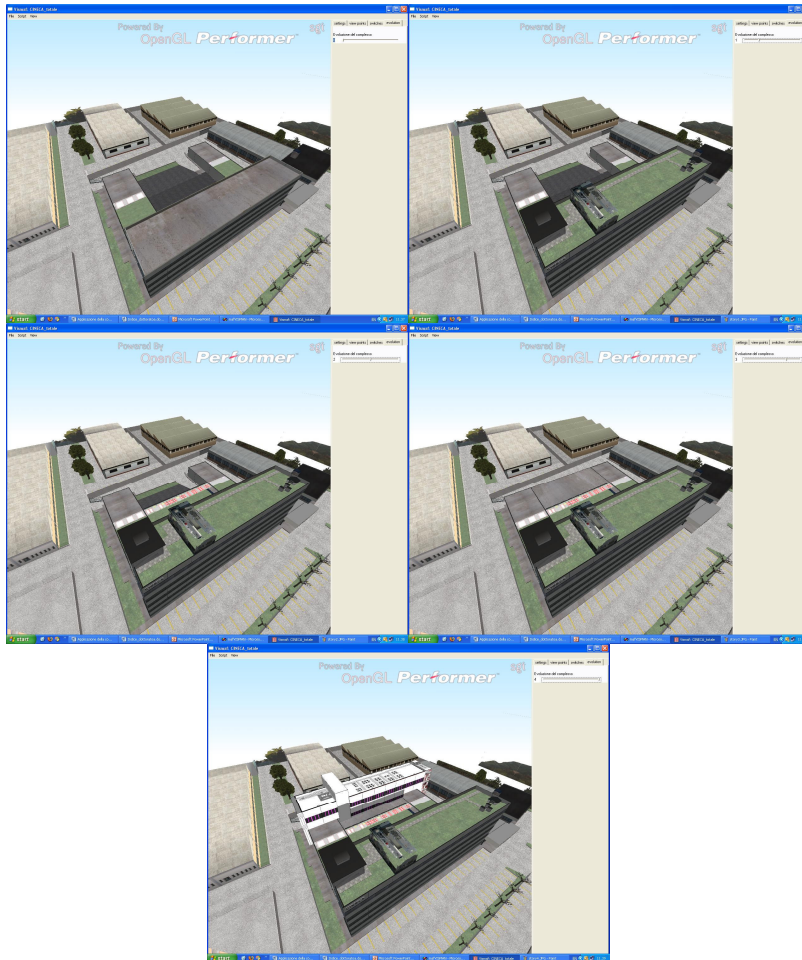


Figura 27: sequenza di immagini che illustra il nodo story e la sua interfaccia grafica

- Nodo di tipo Image Based Render: abbiamo già parlato del nodo Image Based Render nel paragrafo 2.8, esso è utile per simulare nella scena oggetti di notevole complessità utilizzando però soltanto un quadrilatero e un numero variabile di texture. Questi oggetti però non sono salvati nel formato nativo della libreria, dunque fu necessario anche per questo tipo di nodo creare un apposito prefisso, “IBR_nomefile_numerofiles_numero incrementale“, dove è la radice dei nomi di file delle texture da caricare, che l’applicazione si aspetterà in una sotto-directory di nome “IBR” nello stesso percorso del file del

modello caricato; *numfiles* rappresenta il numero delle texture da caricare, queste cambieranno ogni $360/\text{numfiles}$ gradi, cioè ad esempio se si caricano 8 texture, ad esempio 8 foto di un statua prese attorno ad essa, l'applicazione cambierà texture quando l'osservatore si sposta di 45 gradi. Anche per questo nodo, basta inserire nel file un nodo di tipo gruppo con questo nome particolare, Visman durante la scansione dello scene graph, provvederà alla creazione del nodo IBR e al caricamento delle texture.

- **Nodo di tipo PView:** il nodo PView (che sta per Point of View) è stato ideato per navigare a livello del territorio ed avere la possibilità di inserire approfondimenti nei quali addentrarsi. È un po' come atterrare in un sito di cui si dispone di maggiori dettagli nei modelli 3D e di collegamenti a dati, memorizzati in database.

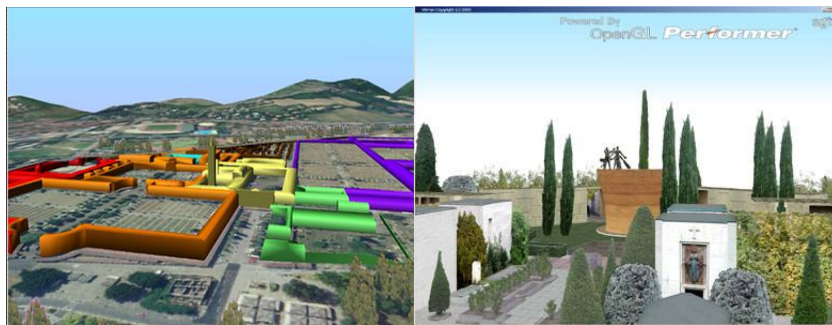


Figura 28: esempio di nodo PView, dal livello del terreno si giunge all'approfondimento

Questo approccio permette un'immediata localizzazione e una contestualizzazione del progetto sul territorio che aiutano notevolmente lo spettatore nella comprensione. I livelli di approfondimento sono teoricamente infiniti, limitati solo dalla memoria disponibile sulla macchina utilizzata per la navigazione. In qualsiasi momento si può tornare al livello superiore.

3.3.1 Scansione dello scene graph

Visman, una volta caricato il modello da file, esegue una scansione dello stesso, andando a percorrere tutti i nodi che lo compongono dalla radice dello scene graph, fino ai nodi foglia.

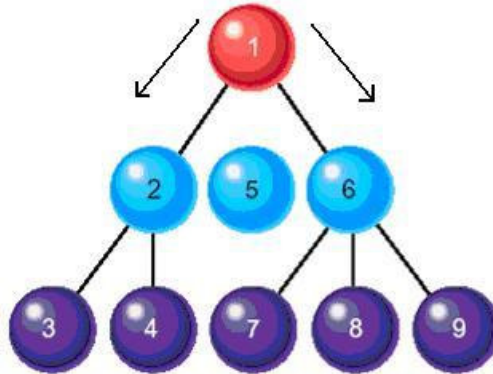


Figura 29: scansione dello scene graph

Se uno dei nodi descritti in precedenza sono incontrati durante la scansione, Visman esegue determinate operazioni, che variano da nodo a nodo, e modifica anche lo scene graph in alcuni casi, in modo da rendere possibili le azioni che si prospettano dai nodi.

3.3.2 Picking

Il picking è l'operazione concettualmente contraria allo scanning dello scene graph, ovvero dai nodi foglia si risale alla radice. Tipicamente questo processo è scatenato dall'utente dell'applicazione, che con il tasto sinistro del mouse clicca su di un oggetto tridimensionale durante la sua navigazione, per ottenere possibilmente maggiori dettagli o approfondimenti sull'oggetto 3D che vede in quel momento davanti a se, nella rappresentazione virtuale.

In questo caso (ovvero picking), Visman non va alla ricerca di qualsiasi nodo speciale, ma solamente dei nodi di tipo collegamento a database e PView. Se

incontra il nodo radice senza aver incontrato nessun nodo di questo tipo, nulla accade, evidentemente il nodo cliccato non era collegato.

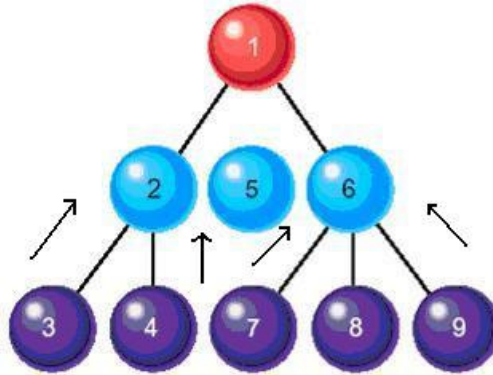


Figura 30: picking

I nodi foglia possono rappresentare una parte della geometria di un oggetto, come ad esempio alcuni poligoni, anche se concettualmente, se l'utente clicca su uno di questi poligoni, intendeva conoscere gli approfondimenti dell'oggetto. Per questo si effettua una ricerca "all'indietro" sullo scene graph, per vedere se un determinato nodo (eventualmente un nodo di sola geometria) derivi da un nodo di raggruppamento che sia collegato a database o PView. Se nella risalita dello scene graph, si incontra uno di questi nodi, allora la relativa azione viene intrapresa, se invece si giunge al nodo radice, allora il nodo foglia cliccato dall'utente non era discendente di nessun nodo particolare e dunque nessuna azione viene compiuta.

3.4 Tipico flusso di lavoro per produrre un'applicazione grafica 3D per i beni culturali

La creazione delle applicazione per il progetto Nu.M.E., sia in locale che via web, ha permesso l'individuazione di un percorso tipico per lo sviluppo dei modelli, dei contenuti ad essi associati e della loro fruizione, in modo che il tutto sia storicamente e filologicamente verificato.

3.4.1 Modellazione 3D

Affinché il modello 3D risultante dalla modellazione sia coerente con i documenti storici, occorre che la modellazione sia un processo interdisciplinare e che ci sia una continua collaborazione tra gli storici, che hanno studiato e che conoscono bene i documenti riguardanti i siti che si vorrebbero ricostruire, con i modellatori 3D che sulla base di questi studi, ricostruiscono virtualmente gli ambienti storici. Questo processo è indispensabile se la ricostruzione finale vuole avere un valore storico e non solo scenografico, assumendo i connotati di una divulgazione storica, supportata dalle più recenti tecnologie informatiche relative alla grafica 3D.

Parleremo ora di alcuni modellatori utilizzati spesso per le ricostruzioni storiche.

3.4.2 Multigen Creator

Multigen Creator è un modellatore nato per applicazioni di grafica 3D in tempo reale, dunque non ha la possibilità di fare rendering di tipo ray-tracing e render-to-texture. Creator fu sviluppato inizialmente per esigenze di tipo militare (simulazioni di armamenti, equipaggiamenti militari, scenari di battaglia), e in un secondo momento si diffuse come modellatore specifico per la grafica interattiva in tempo reale. Il software è piuttosto costoso, dunque la sua maggiore diffusione si riscontra nei centri di calcolo o nei centri di ricerca che si erano dotati di sistemi di visualizzazione avanzati come quelli presentati nel capitolo precedente.

Una delle caratteristiche degne di nota di questo modellatore è il forte supporto ad un gerarchia di nodi con la quale è possibile creare i propri modelli. Questo consente al modellatore non soltanto di modellare la geometria e di aggiungere le texture agli oggetti, ma anche di preparare lo scene graph per ottenere dal visualizzatore dei comportamenti particolari (in

convenzione con il visualizzatore) seguendo alcuni semplici accorgimenti, ovvero raggruppando la geometria e dando a questo nodi dei nomi particolari.

Inoltre il modellatore è specifico per ambienti real-time, dunque favorisce la modellazione utilizzando pochi poligoni, che è indispensabile per ottenere buone prestazioni durante la navigazione in tempo reale.

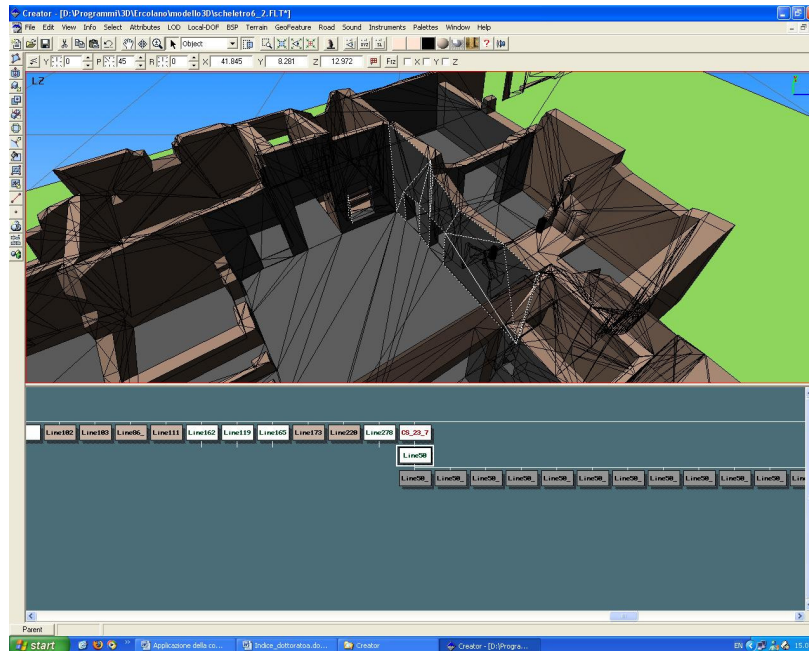


Figura 31: Multigen Creator: modellazione e scene graph

Il formato utilizzato dal modellatore è Open-Flight, di cui sono state pubblicate le specifiche, diventando così un formato aperto; questo formato conserva lo scene graph impostato nel modellatore, che dunque può essere caricato da file dal visualizzatore. Questa caratteristica è fondamentale per poter passare al visualizzatore dei modelli con scene graph preparati ad hoc per ottenere i comportamenti che abbiamo descritto in precedenza.

3.4.3 Autodesk 3D Studio Max con plug-in Flight Studio

3D Studio (successivamente rinominato 3D Studio Max) è stato uno dei primi modellatori disponibili per PC, all'inizio degli anni novanta; prima di allora, tipicamente era necessario utilizzare work stations con sistema operativo Unix, oppure super computer grafici. È stato sviluppato da Autodesk, famosa per un altro prodotto di successo, Autocad¹² un CAD bidimensionale molto utilizzato in ambito di ingegneria civile.

3D Studio Max è al contrario di Multigen Creator, un modellatore rivolto soprattutto al rendering in ray-tracing, dunque di alta qualità, con i tipici tempi di attesa per il rendering che occorre rispettare.

Questo modellatore è utilizzato anche in produzioni cinematografiche, per inserire effetti speciali all'interno dei film, a testimonianza della qualità raggiunta dal prodotto.

Una delle caratteristiche più interessanti è la possibilità di espandere il pacchetto con i cosiddetti "plug-in" (letteralmente dall'inglese inserimento di uno spinotto), che sono dei veri e propri sotto-programmi all'interno del pacchetto principale. Vi sono in commercio tantissimi plug-in, per gli usi più disparati, in quanto vengono utilizzati dall'industria cinematografica, quello che interessa di più ai nostri scopi è la plug-in Flight Studio, sviluppato da Bluerock¹³.

Questa plug-in introduce non solo la possibilità di importare ed esportare modelli nel formato Open Flight, ma anche l'organizzazione gerarchica a scene graph all'interno di 3D Studio Max, che altrimenti è lacunosa.

¹² <http://www.autodesk.it/>

¹³ <http://www.bluerock.com/products.html>

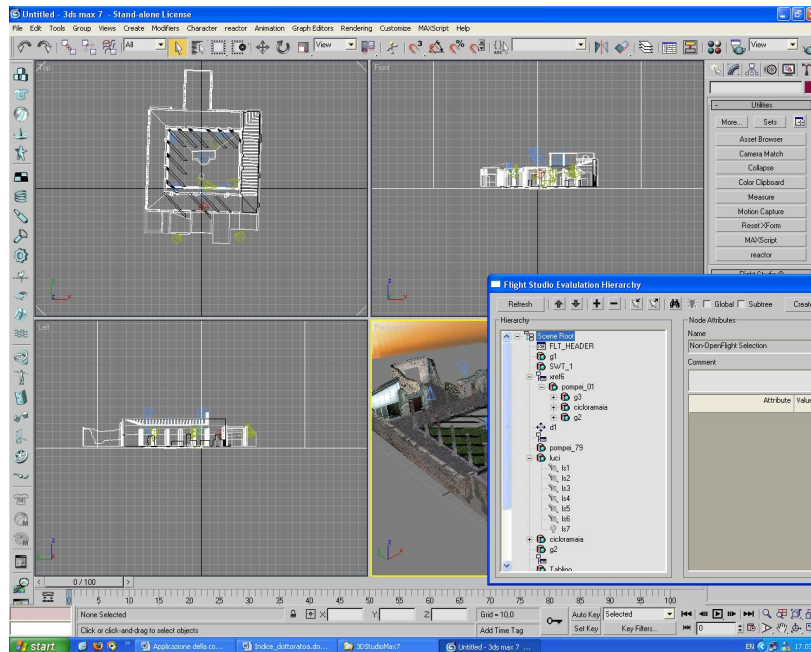


Figura 32: 3D Studio Max e la plug-in Flight Studio

I due principali vantaggi di 3D Studio Max rispetto a Multigen Creator sono innanzitutto il costo molto minore e la possibilità di creare con poco sforzo modelli con render-to-texture, che dalla versione 6 del software è supportata nativamente, questo rende i modelli molto più piacevoli e realistici, grazie ai calcoli di ombreggiatura, altrimenti assenti.

Esiste infine una plug-in per 3D Studio Max, chiamata OSG-Exporter¹⁴, che permette di esportare i modelli nel formato Open Scene Graph.

3.4.4 Verso l'open source nella modellazione: Blender

Un modellatore che negli ultimi anni ha sempre più fatto parlare di sé è Blender¹⁵. La principale caratteristica di questo modellatore è sicuramente di essere pubblicato con licenza open source, dunque liberamente scaricabile dal sito Internet, sia l'eseguibile che il codice sorgente.

¹⁴ <http://osgexp.vr-c.dk/>

¹⁵ <http://www.blender.org/>

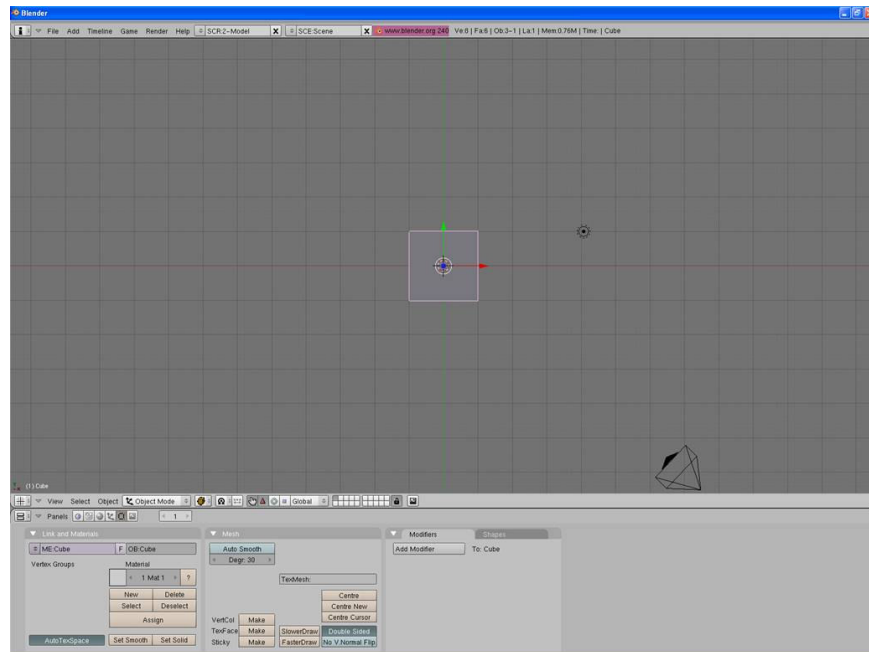


Figura 33: il modellatore open source Blender

Blender è disponibile per molte piattaforme, ovvero Windows, Linux e MacOS, questo anche grazie al fatto che la sua interfaccia grafica è stata programmata interamente in OpenGL, dunque il porting su sistemi operativi che supportano OpenGL è piuttosto semplificato.

Se negli anni scorsi, alcune caratteristiche lo ponevano in secondo piano rispetto ai modellatori commerciali, gli sviluppi più recenti lo rendono sempre più interessante come valida alternativa a costo zero ai più blasonati software di modellazione a pagamento.

Nel 2006 è stato presentato un cortometraggio, Elephants Dream ¹⁶, interamente realizzato con Blender, per dimostrare che anche con un modellatore open-source è possibile realizzare filmati in computer grafica di alto livello; il filmato è a sua volta open-source, nel dvd ci sono anche i modelli delle scene utilizzate per il rendering finale del filmato, che possono

¹⁶ <http://orange.blender.org/>

essere riprese, modificate e montate di nuovo per creare una versione personalizzata di Elephants Dream.

Anche per Blender esiste la stessa plug-in, OSG-Exporter¹⁷, che permette di esportare i modelli nel formato Open Scene Graph. È dunque possibile immaginare che in un immediato futuro sia possibile costituire una catena completamente open-source dalla modellazione, alla visualizzazione, all'utilizzo di database open-source, per la fruizione di modelli e dati ad essi collegati.

3.4.5 Preparazione dei dati da collegare al modello 3D

Oltre alla modellazione, occorre prestare attenzione, durante la preparazione di un'applicazione di realtà virtuale in ambito storico, ai dati che si vogliono collegare agli oggetti 3D modellati.

Il risultato che si vuole ottenere per i dati collegati al 3D deve guidare l'organizzazione dei dati stessi nel database: nel caso di semplici collegamenti a fonti multimediali, come immagini, audio, schede HTML, basterà assegnare un numero incrementale ai postfissi degli oggetti 3D che si vogliono collegare al db e avere una colonna (chiave primaria) nella tabella di accesso al database con gli stessi numeri identificativi; l'applicazione, durante la navigazione, potrà effettuare una query sul database a partire dal nodo individuato attraverso il picking (vedi paragrafo 3.3.2) e mostrare il campo di dati relativo. Un'altra soluzione, più complessa sarebbe quella di allestire un server web che interroga un server database, al fine di comporre delle pagine web facendo complesse interrogazioni al database; in quest'ultimo caso, l'applicazione effettua una richiesta http tramite Internet andando a comporre la stringa di indirizzo http con il numero identificativo dell'oggetto 3D piccato, quindi il server web, effettuate le query sul db comporrà la pagina web completa.

¹⁷ <http://projects.blender.org/projects/osgexport/>

3.4.6 Visualizzatore 3D apposito

L'ultimo componente del processo, che unisce tutti i dati (modelli 3D storicamente validati, dati storici provenienti dalle fonti, ecc.), è il visualizzatore. Come abbiamo scritto precedentemente (paragrafo 3.2), le esigenze delle ricostruzioni in ambito storico ci ha portato a scrivere un software ad hoc, poiché non esiste in commercio un navigatore che abbia la flessibilità e le caratteristiche tali da soddisfare le richieste.

Attraverso il navigatore è possibile creare una nuova applicazione per ogni progetto; esso deve avere sufficiente configurabilità da non dover modificare il codice sorgente ogni qual volta che si vogliono cambiare dei parametri di visualizzazione, come ad esempio il punto di vista iniziale (diverso da modello a modello), la velocità di navigazione, i colori di sfondo e così via.

Il visualizzatore mette insieme tutto il lavoro preparatorio sia degli studiosi che dei modellatori: in un unico ambiente virtuale, si fruiscono le ricostruzioni e tutti i dati storici che i responsabili del progetto hanno deciso fossero utili per un arricchimento della navigazione; l'utente finale percepisce tramite l'accuratezza della ricostruzione e la ricchezza dei contenuti collegati la profondità e la complessità del progetto. Lo scopo di questo tipo di progetti non è di sostituire la letteratura con dei progetti multimediali, ma anzi, quello di stimolare la ricerca di approfondimenti tramite degli strumenti, che validati dal punto di vista storico, siano di facile utilizzo e che affascino immediatamente tramite il coinvolgimento delle ricostruzioni tridimensionali e dei contenuti.

4 Organizzazione dei dati

Abbiamo spesso fatto riferimento all'organizzazione dei dati da collegare ai modelli tridimensionali, in questo capitolo approfondiremo l'argomento dei database e della loro gestione.

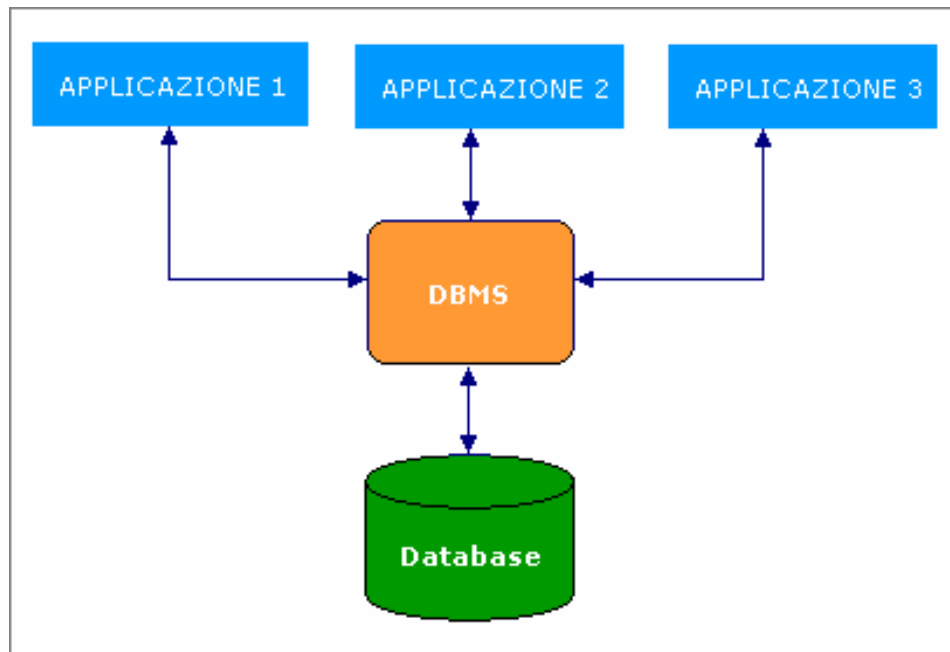


Figura 34: Struttura di un DBMS

4.1 Database Management System

Per usare le informazioni contenute in un database, occorre un Database Management System (DBMS). Un DBMS è costituito da un insieme assai complesso di programmi software che controlla l'organizzazione, la memorizzazione e il reperimento dei dati (campi, record e archivi) in un database. Inoltre controlla anche la sicurezza e l'integrità del database.

Il DBMS accetta richieste di dati da parte del programma applicativo e istruisce il sistema operativo per il trasferimento dei dati appropriati.

I tre tipi di organizzazione più comuni sono il modello gerarchico, il modello reticolare e il modello relazionale. Un sistema di gestione del database può

fornire uno, due o anche tutti e tre questi metodi. Sono usate anche le liste invertite e altri metodi.

La scelta della struttura più adatta dipende dal tipo di applicazione, dalla frequenza delle transazioni e dal numero di interrogazioni che saranno effettuate.

4.1.1 Il modello di database gerarchico

Si può fissare la data di nascita di questo modello alla fine degli anni '60, quando IBM sviluppa e introduce sul mercato IMS, il primo database gerarchico, ma anche il primo DBMS in assoluto.

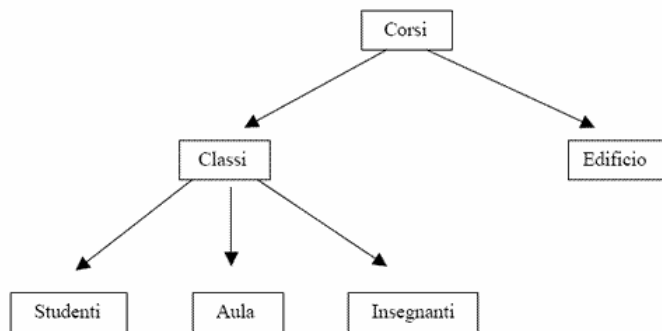


Figura 35: Gli archivi sono composti da record chiamati segmenti. I segmenti sono in rapporto gerarchico tra loro attraverso legami di tipo padre-figlio

Un database gerarchico è un insieme di archivi. Gli archivi sono composti da record chiamati segmenti. I segmenti sono in rapporto gerarchico tra loro attraverso legami di tipo padre-figlio.

La struttura ad albero che caratterizza il modello gerarchico si basa sulla possibilità di individuare un segmento principale, il padre o la radice, dal quale dipendono n segmenti figli, che a loro volta si trasformano in padri per altri figli e così via. A questi, in virtù della totale dipendenza dal padre, è possibile fare riferimento solo attraverso il passaggio dal nodo principale. Non è possibile dal figlio risalire al padre.

Questa architettura mal si adatta ad una gestione moderna e dinamica delle basi di dati.

4.1.2 Il modello di database reticolare

La logica dalla quale parte il database di tipo reticolare è organizzata intorno a record e puntatori. Il modello reticolare è stato recepito da un organismo di unificazione e, per questo motivo, ne esiste una versione standard, detta CODASYL¹⁸. Questo modello, a differenza del modello relazionale che si basa su gruppi di record, considera un record alla volta. Esso è molto vicino alla struttura fisica di memorizzazione dei record; infatti, per poter "navigare" all'interno del database si devono utilizzare i puntatori, proprio come avviene nella memoria fisica di un computer.

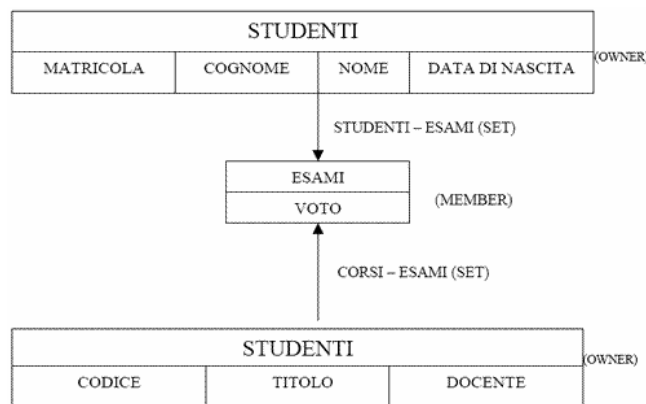


Figura 36: Correlazioni sono espresse attraverso record particolari, chiamati record di collegamento (member), che formano delle catene tra le varie parti del sistema

Il database reticolare può essere visto come un'estensione del modello gerarchico, al quale sono stati apportati importanti miglioramenti.

In una struttura gerarchica un segmento figlio può avere solo un segmento padre; non è così nel modello reticolare: ogni record può avere un numero qualsiasi di record subordinati e di record precedenti e le correlazioni sono

¹⁸ J. S. Knowles and D. M. R. Bell, *The Codasyl Model*, in *Databases - Role and Structure*, ed. P. M. Stocker, P. M. D. Gray, and M. P. Atkinson, CUP, 1984.

espresse attraverso record particolari, chiamati record di collegamento (member), che formano delle catene tra le varie parti del sistema.

Un database reticolare è composto da due insiemi di dati, un insieme di record e un insieme di legami: i tipi record sono costituiti da campi tra i quali ci deve essere anche un campo chiave, mentre i legami sono realizzati memorizzando le coppie di chiavi nelle entità associate.

Non esiste quindi una gerarchia predefinita tra le entità. Come detto, la differenza principale rispetto al modello gerarchico consiste nel fatto che un record figlio può avere un numero qualsiasi di padri: in questo modo sono evitate situazioni di ripetizione di dati uguali.

La gestione delle informazioni nei database costruiti secondo il modello reticolare è naturalmente più complessa, in quanto deve essere utilizzata una struttura a grafo: risulta pertanto più difficile l'implementazione e la costruzione del software applicativo.

4.1.3 Il modello di database relazionale

Il termine Relational Database Management System (RDBMS), o sistema relazionale per la gestione di basi di dati, indica un database management system basato sul modello relazionale, ed è stato introdotto da Edgar F. Codd.

Codd ha introdotto il termine nel testo del seminario "A Relational Model of Data for Large Shared Data Banks" (un modello relazionale di dati per gestire grandi banche dati condivise)¹⁹. In questo scritto ed in altri successivi, Codd definì cosa intendeva per relazionale fissando dodici punti molto restrittivi²⁰.

Tuttavia, molte delle prime implementazioni del modello relazionale non erano conformi a tali regole, per questo il termine fu gradualmente cambiato fino a descrivere una più ampia classe di sistemi di basi di dati.

¹⁹ Codd, E. F., *A relational model of data for large shared data banks*, *Communications of the ACM*, Vol. 13, No. 6, June 1970, pp. 377-387.

²⁰ Codd E.F., *Is Your DBMS Really Relational?*, in "ComputerWorld", 14 ottobre e 21 ottobre 1985.

I requisiti minimi per cui quei sistemi venissero chiamati RDBMS erano:

- deve presentare i dati all'utente sotto forma di relazioni (una presentazione a tabelle può soddisfare questa proprietà).
- deve fornire operatori relazionali per manipolare i dati in forma tabellare.

Non c'è un accordo unanime su ciò che può chiamarsi un DBMS "relazionale", e le definizioni più popolari sono piuttosto imprecise. Alcuni sostengono che perfino presentare una visione dei dati sotto forma di righe e colonne sia sufficiente a qualificare un RDBMS.

Molti teorici dei modelli di dati sostengono invece che se un database non implementa tutte le regole di Codd, non può dirsi relazionale. Questa definizione lascia fuori molti dei database in commercio, non considerandoli "veramente-relazionali" (TRDBMS), ma piuttosto "pseudo-relazionali" (PRDBMS).

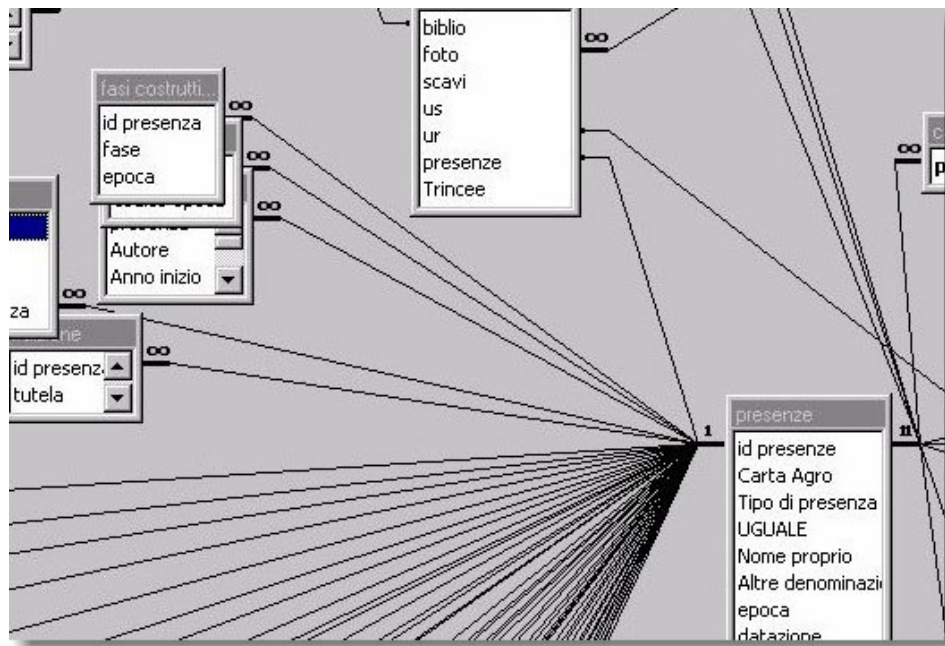


Figura 37: Relazioni tra tabelle del database che cataloga i dati degli scavi del Comune di Roma

I critici di questa posizione sostengono che è eccessivamente rigida, notando che il solo uso dell'SQL (Structured Query Language) sarebbe di per sé sufficiente a violare le regole di Codd.

La creazione di tabelle in realtà si basa tutto sul linguaggio SQL. All'interno di questo linguaggio sono contenuti comandi per la creazione di database e tabelle, per la definizione delle chiavi e dei tipi di dati.

La storia di SQL inizia nel 1974 con la definizione da parte di Donald Chamberlin e di altre persone che lavoravano presso i laboratori di ricerca dell'IBM di un linguaggio per la specificazione delle caratteristiche dei database che adottavano il modello relazionale.

Questo linguaggio si chiamava SEQUEL (Structured English Query Language) e fu implementato in un prototipo chiamato SEQUEL-XRM fra il 1974 e il 1975. Le sperimentazioni con tale prototipo portarono fra il 1976 ed il 1977 ad una revisione del linguaggio (SEQUEL/2), che in seguito cambiò nome per motivi legali, diventando SQL.

Il prototipo (System R) basato su questo linguaggio venne adottato ed utilizzato internamente da IBM e da alcuni sui clienti scelti. Grazie al successo di questo sistema, che non era ancora commercializzato, anche altre compagnie iniziarono a sviluppare i loro prodotti relazionali basati su SQL.

Il modello dominante oggi è quello relazionale, normalmente utilizzato con il linguaggio di interrogazione SQL. Molti RDBMS supportano le API (Application Program Interface) dell'Open Database Connectivity (ODBC) o Java Database Connectivity (JDBC, lo standard per Java), che forniscono ai programmatori strumenti standardizzati per l'accesso ai database.

Come si è visto, il RDBMS è un modello di database che consiste di diverse tabelle separate che sono correlate l'una all'altra attraverso campi chiave. Si può accedere alle informazioni memorizzate, grazie alle relazioni stabilite tra queste.

Ad esempio, un database relazionale vede il collegamento fra una tabella che contiene l'anagrafico degli impiegati in una società e la tabella delle retribuzioni, tramite un codice univoco che identifica il singolo impiegato.²¹

Il modello Relazionale è un modello logico di dati la cui architettura è suddivisa in tre livelli nel modo seguente:

- Livello concettuale
è denominato database logico ed è costituito dalla rappresentazione astratta del database, in quanto indipendente dalla implementazione fisica. Questo livello si pone tra il livello interno e quello esterno.
- Livello interno
viene detto database fisico ed è costituito dalla implementazione del database logico e considera i tipi di dato, i formati, le strutture di memorizzazione ed i metodi di accesso, esso rappresenta la forma in cui il database viene memorizzato e richiamato.
- Livello esterno
concerne le viste, intese come porzioni del database logico, che gli utenti hanno sul database.

Quando si usa un RDBMS i sistemi informativi possono essere adeguati molto facilmente al cambiamento delle richieste informative dell'organizzazione. Possono essere aggiunte al database nuove categorie di dati senza dover stravolgere il sistema esistente.

Il sistema di sicurezza dei dati impedisce agli utenti non autorizzati di visualizzare o aggiornare il database. Mediante l'uso di password agli utenti è permesso l'accesso all'intero database o ad un suo sottoinsieme: in questo secondo caso si parla di subschema.

Per esempio un database di impiegati può contenere tutti i dati riguardanti un singolo soggetto, ma un gruppo di utenti può essere autorizzato a vedere solamente i dati riguardanti lo stipendio, mentre altri utenti possono essere

²¹ Atzeni P., Ceri S., Paraboschi S., Torlone R., Basi di dati - Modelli e linguaggi di interrogazione, McGraw-Hill, 2002.

autorizzati a vedere solamente le informazioni che riguardano la sua storia lavorativa e la situazione sanitaria.

Un RDBMS può mantenere l'integrità del database non consentendo a più utenti di modificare lo stesso record contemporaneamente (blocco del record). Il database può impedire l'immissione di due record duplicati; per esempio può essere impedita l'immissione nel database di due clienti con lo stesso numero identificativo (campi chiave). L'insieme di regole che determinano l'integrità e la consistenza di una base di dati prendono il nome di *Vincoli di integrità referenziale*.

I linguaggi di interrogazione del database mediante *query* (interrogazioni) e i generatori di *report* permettono agli utenti di interrogare in maniera interattiva il database e di analizzarne i dati.

Se un RDBMS fornisce un modo per aggiornare ed immettere nuovi dati nel database, oltre che per interrogarlo, questa capacità permette di gestire database personali. Comunque queste funzionalità non danno la possibilità di mantenere traccia delle revisioni e non forniscono gli strumenti necessari alla gestione di una organizzazione multi-utente. Questi controlli sono disponibili solamente quando un insieme di programmi applicativi sono appositamente costruiti per gestire e coordinare ciascuna funzione di immissione o modifica dei dati.

La progettazione del database (*database design*) è il processo decisionale su come organizzare questi dati in tipi di record e su come ciascun tipo di record si relaziona con gli altri. Un RDBMS dovrebbe rispecchiare la struttura dei dati dell'organizzazione e gestire in maniera efficiente le varie transazioni.

Alcune regole di base da rispettare sono:

- Creare una tabella separata per ciascun insieme logico di dati.
- Eliminare i campi ridondanti, creando una nuova tabella.
- I campi di una tabella devono dipendere da una e una sola chiave primaria (*dipendenza funzionale*).

- La chiave primaria deve garantire l'unicità di ciascun record.

4.1.4 Database relazionali e archeologia

La disponibilità di computer e programmi sempre più potenti permette di organizzare, trattare e analizzare dati sempre più complessi, correlando immagini a più database, anche differenti come struttura e costruzione, messi a loro volta in relazione.

I database relazionali si dimostrano particolarmente adatti al perseguimento di simili compiti. Infatti, non necessitano di una fonte uniforme e di "buon carattere"²² - cioè adatta all'elaborazione elettronica, ma è possibile partire anche da un oggetto archeologico), con le sue coordinate, e ad esso collegare tutti i dati storici che provengono anche da fonti disomogenee, di tipo profondamente diverso.

4.2 Database in locale o tramite Server

Per le applicazioni di grafica 3D interattive, relative alla storia, sono possibili due approcci per il collegamento degli oggetti tridimensionali ai dati contenuti nel database. Il primo, più semplice dal punto di vista tecnico e implementativo è quello di avere tutto il database in un unico file in locale (ovvero sullo stesso computer utilizzato per il visualizzatore della grafica) insieme ai modelli 3D. In questo caso le interrogazioni al database coincidono con degli accessi al file. Le librerie di utilizzo del database devono essere linkate insieme all'eseguibile del visualizzatore, cosicché è lo stesso visualizzatore ad effettuare le query ed a ottenere i dati contenuti nei record; ed è anche il visualizzatore a dover mostrare i dati in un'apposita finestra o porzione di vista, una volta che li abbia letti dal database. Ovviamente

²² Bocchi F., *Nuove metodologie per la storia delle città: La città in quattro dimensioni*, in *Medieval Metropolises, Proceedings of the Congress of Atlas Working Group – Bologna 8-10 maggio 1997*, Grafis, 1999.

qualsiasi tipo di elaborazione dei dati è possibile prima di mostrare i dati o anche dopo averli visualizzati.

Il secondo tipo di approccio consiste nell'istallare e nel tenere in esecuzione un server database, che a sua volta è interrogato da un server web, il quale compone le pagine web a partire dai dati contenuti nei record e li invia al browser che ha fatto richiesta della pagina. In questo secondo caso il visualizzatore deve soltanto aprire un web browser ed effettuare la richiesta all'indirizzo web apposito, per innescare il meccanismo di query e di visualizzazione dei risultati. L'indirizzo web, in questo caso, contiene in qualche parte, tipicamente alla fine, il numero identificativo dell'oggetto 3D interrogato dall'utente tramite il pulsante sinistro del mouse. Quest'ultima architettura per l'accesso ai dati del database consente una maggiore complessità del sistema; inoltre vi è un indubbio vantaggio di centralizzazione dei dati, che consente ad esempio a più di uno storico di immettere o comunque lavorare sul database contemporaneamente; inoltre si risolve in modo definitivo il problema della replicazione dei dati, essendo la gestione centralizzata. Gli svantaggi sono: per accedere ai dati, il pc su cui è installato il visualizzatore deve essere collegato ad Internet per poter accedere ai dati ed inoltre l'istallazione e la gestione dei due server (db e web) richiede una struttura hardware che il database in locale non richiede.

4.3 Open source nei database

Ancor prima che nella grafica, il software open source si è diffuso nell'ambito dei database. Sebbene tra i database con architettura server/client il software di maggiore diffusione resti Oracle²³, anche grazie a prestazioni eccellenti anche quando i record superano la soglia del milione, da vari anni si sta sempre più diffondendo un database open-source chiamato MySQL²⁴. Questo database,

²³ <http://www.oracle.com>

²⁴ <http://www.mysql.org>

come per molti altri software open-source, è disponibili per i maggiori sistemi operativi, sia per singolo pc che per cluster di server; anche per MySQL, il download del software eseguibile e del codice sorgente è gratuito e libero, la fonte di reddito per i mantentori del software viene, come da molte altre aziende che rilasciano software in modalità open-source, dalla consulenza effettuata sul proprio prodotto.

Un database open-source che ultimamente sta riscuotendo un certo successo è SQLite²⁵; come abbiamo mostrato nel precedente paragrafo, in questo caso l'utilizzo del db è immediato, non occorre installare ne configurare nessun server, in quanto un intero database risiede in un unico file. Questo database è stato utilizzato nell'ultima versione di Visman, insieme ad Open Scene Graph per la grafica 3D, realizzando così un visualizzatore che fa uso di sole librerie open-source. Visman dunque non ha più bisogno di alcuna licenza software per essere installato ed utilizzato, sia nella versione in locale, come eseguibile da lanciare in un pc, che nella versione internet, come plug-in da installare al proprio web browser, di cui parleremo nel prossimo capitolo.

²⁵ <http://www.sqlite.org/>

5 Nu.M.E. in Internet

5.1 Applicazione in locale o tramite web

Nel capitolo (DIRE QUALE) abbiamo visto come l'hardware relativo alla parte grafica dei PC si sia notevolmente evoluto nel corso di questi ultimi anni. Parallelamente a questa evoluzione è comparso un fenomeno per molti versi ancora più dirompente nel mondo dell'informatica e nella società, ovvero Internet. Anche in questo caso l'offerta di connessione sia casalinga che nelle infrastrutture pubbliche e aziendali ha subito una enorme accelerazione, passando da sistemi analogici attraverso la rete telefonica a sistemi digitali (dapprima ISDN, poi ADSL) e attraverso fibra ottica.

La diffusione di Internet a banda larga e i relativi bassi costi raggiunti al giorno d'oggi permettono un utilizzo della rete per mezzi non ancora sperimentati dall'industria e dal mercato, che si sono focalizzati, come era prevedibile, su ambiti più remunerativi quali informazione, home banking, finanza, ecc.

È possibile oggi pensare di portare la grafica 3D su internet, per scopi divulgativi e didattici, anche in ambiti non tradizionalmente legati alla grafica virtuale come la storia e l'archeologia; ambiti di cui il progetto Nu.M.E. è stato un precursore.

5.2 Breve storia di Internet

Le origini di Internet si trovano in ARPANET, una rete di computer costituita nel settembre del 1969 negli USA da ARPA, l'Advanced Research Projects Agency. ARPA fu creata nel 1958 dal Dipartimento della Difesa degli Stati Uniti durante la guerra fredda, con l'obiettivo di recuperare lo svantaggio tecnologico nei confronti dell'Urss, che aveva lanciato il satellite Sputnik nel 1957. Lo scopo era di realizzare una rete di comunicazione in grado di

resistere ad un attacco nucleare, quindi in grado di funzionare anche nel caso in cui alcuni o molti nodi facenti parte della rete avessero cessato di funzionare causa bombardamento atomico.

Il primo collegamento telefonico da computer a computer avvenne nel 1969, fra l'Università della California di Los Angeles e lo Stanford Research Institute, che furono così i primi due nodi di Internet: la prima applicazione che abbia mai funzionato su internet fu una sessione Telnet.

Nel 1973 fu istituito il progetto del protocollo di controllo trasmissione (TCP), standard indispensabile per la comunicazione tra reti di computer. Nel 1978 fu aggiunto un protocollo tra rete e rete (IP), mettendo a punto il definitivo protocollo su cui ancor oggi opera Internet, il TCP/IP.

Il concetto centrale dietro ad internet protocol è l'idea di spezzettare le informazioni e di inviarle attraverso dei pacchetti numerati, in modo che attraverso la numerazione, il destinatario dell'informazione possa essere in grado di ricostruire l'informazione originaria. I pacchetti possono essere instradati in modi diversi tra il mittente ed il destinatario, per via della numerazione, e questo consentiva di avere una minore vulnerabilità ad attacchi nemici, in quanto per interrompere la comunicazione tra mittente e destinatario occorrerebbe distruggere tutti i nodi tra i due.

Il dispiegamento delle potenzialità di Internet e la sua progressiva diffusione popolare sono però frutto dello sviluppo del WWW, il World Wide Web, un sistema per la condivisione di informazioni in ipertesto del 1990 sviluppato da Tim Berners-Lee presso il CERN (Centro Europeo per la ricerca nucleare), il quale nel 1990 mise a punto il protocollo HTTP ed una prima specifica del linguaggio HTML, sulla base dei quali sono stati realizzati un primo programma browser/editor ipertestuale per il WWW, utilizzato all'interno del CERN nel 1991. Nel 1993 uscì la release Mosaic, che combinava una capacità grafica avanzata e diverse tecnologie d'interfaccia multimediali. Da Mosaic, che poi prese il nome di Netscape Communication, fu sviluppato il primo browser commerciale, Netscape Navigator, nel 1994, che successivamente è

stato reso disponibile online. Microsoft scopre Internet nel 1995 e fa uscire Internet Explorer insieme con il proprio sistema operativo Windows 95. Nel 1998, ormai per pura concorrenza con Microsoft, Netscape rilascia Navigator in open source.

5.3 Internet attuale

Dai primi collegamenti attuati tra università americane, dopo circa quaranta anni la diffusione di Internet è stata travolgente, al punto di cambiare le società del mondo con nuove possibilità, impensabili solo pochi decenni or sono.

Non solo la diffusione di Internet è estremamente capillare, ma anche i collegamenti a banda larga, ovvero ADSL e fibra ottica, che permettono velocità fino 20 Mbit/s si stanno diffondendo anche grazie a finanziamenti governativi; nel 2006 solo il 10% della popolazione italiana non era ancora raggiunta da Internet a banda larga. Questo scenario, con i progressi tecnologici degli ultimi anni, permette di poter pensare di portare progetti di grafica 3D direttamente sul web, per avere la massima visibilità e raggiungere un pubblico molto più vasto di quanto fosse possibile in passato.

5.4 I web browser attualmente più diffusi

5.4.1 Mozilla Firefox

Uno dei browser più diffusi in questo momento è Mozilla Firefox²⁶, un software pubblicato in open source che riprende l'eredità di Netscape Navigator e quindi di Mosaic, il primissimo browser in assoluto.

L'azienda dietro a Firefox è una fondazione senza scopi di lucro, la Mozilla Foundation, che ha sede in California.

²⁶ <http://en-us.www.mozilla.com/en-US/firefox/>

Mozilla Firefox, insieme a Open Office, suite di programmi open source per le tipiche operazioni d'ufficio, ha contribuito notevolmente alla diffusione di Linux come sistema operativo per uso desktop, che precedentemente era utilizzato quasi esclusivamente per Server web o database.

5.4.2 Microsoft Internet Explorer

Microsoft lanciò Internet Explorer insieme a Windows 95 appunto nel 1995. Questo browser è giustamente stato considerato strategico da Microsoft per la diffusione di Windows come sistema operativo, quindi molta attenzione è stata riservata al suo sviluppo. Rimane, anche per il fatto che Windows è il sistema operativo più usato per ambienti desktop, il browser più utilizzato attualmente.

5.5 Google Earth

Un'applicazione che recentemente ha fatto molto discutere di sé è sicuramente Google Earth, anzitutto perché è stata lanciata da Google, azienda leader in Internet per motori di ricerca e come collezionista di pubblicità via web, inoltre perché è stata la prima applicazione legata alla grafica 3D lanciata da Google.

L'applicazione consiste in un client, liberamente scaricabile dal sito web²⁷ che si collega con un server Google, e funge da esploratore del mondo. Il punto di partenza è la Terra vista in lontananza, come ad un'astronave in orbita, con la possibilità di avvicinarsi e di vedere con maggiore dettaglio tutte le località del mondo. Le mappe a maggiore risoluzione vengono costantemente scaricate dal client, e vengono aggiornate mano a mano che i dati arrivano, dipendentemente dalla banda disponibile.

²⁷ <http://earth.google.it/>

Oltre alle mappe georeferenziate, sono visibili punti d'interesse turistici, ristoranti, negozi di vario genere, e nell'ultima versione anche oggetti modellati.

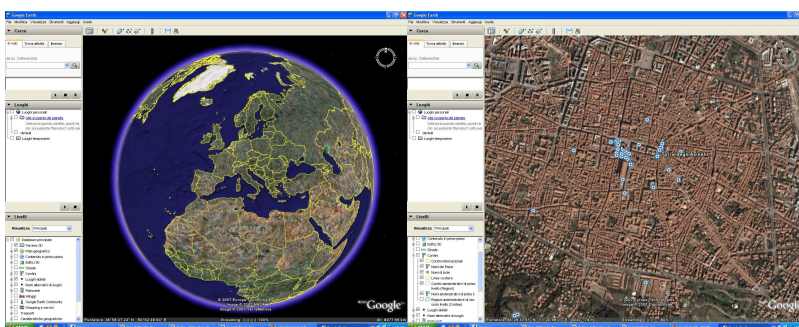


Figura 38: Google Earth

5.6 II VRML

La prima implementazione di grafica 3D per Internet fu attraverso un linguaggio appositamente studiato, il VRML che sta per Virtual Reality Modeling Language.

Il VRML è un linguaggio simile ad HTML, ma consente di definire vertici, spigoli, colore della superficie, texture, brillantezza, trasparenza ecc.

Non essendo supportato nativamente dai web-browser, è necessario installare appositi plug-in per permettere la navigazione.

Il VRML ha avuto un certo successo nella seconda metà degli anni novanta, oggi si aspetta una revisione del linguaggio verso una nuova specifica chiamata X3D, di cui però non sono disponibili ancora dei player.

5.7 Grafica 3D interattiva sul web: plug-in per browser

I moderni browser web permettono l'installazione di plug-ins, che come abbiamo visto per i modellatori 3D, sono dei veri e propri sotto-programmi

all'interno del software principale e implementano delle funzionalità altrimenti non disponibili per il software originale, in questo caso il web browser.

Alcuni esempi di plug-ins molto famosi e oramai pressoché indispensabili anche per la navigazione Internet di persone comuni, sono Adobe Acrobat Reader²⁸ per la lettura dei files in formato pdf, e Macromedia Flash²⁹ utilizzato per animazioni, applicazioni interattive bidimensionali, filmati, ecc.

L'area dei plug-ing tipicamente viene delimitata in un riquadro della pagina web, dove si interagisce nella modalità prevista dal plug-in.

5.7.1 Plug-in per il progetto Nu.M.E.

Per il progetto Nu.M.E. è stato realizzato un plug-in per i maggiori web browser esistenti in questo momento, ovvero Internet Explorer e Mozilla Firefox; l'idea di realizzare un programma client è stata scartata poiché il plug-in, seppure più difficile da realizzare tecnicamente, consente di inserire la parte di grafica 3D all'interno di un contesto HTML che può essere molto utile per comunicare il progetto di cui la ricostruzione virtuale fa parte.

Il plug-in risulta più impegnativo da realizzare perché occorre considerare le interazioni con i web browser, che invece sono escluse con un'applicazione stand-alone come può essere un client.

Per realizzare il plug-in, è stato necessario anzitutto fare il porting dell'applicazione Visman da OpenGL Performer a Open Scene Graph, per vari motivi: il primo consiste nel fatto che Performer, anche nella versione free effettua un controllo di una licenza sul disco, inoltre Silicon Graphics da circa un anno a questa parte è in seri problemi finanziari, tali che è stata tagliata la parte dell'azienda che si occupava di computer grafica sia dal punto di vista hardware che software, dunque lo sviluppo di Performer risulta bloccato; al contrario la comunità di sviluppatori di Open Scene Graph risulta essere molto vivace e gli sviluppi sono costanti e frequenti. Da non sottovalutare infine il

²⁸ <http://www.adobe.com/products/acrobat/>

²⁹ <http://www.adobe.com/it/products/flash/flashpro/>

fatto che Open Scene Graph è open-source dunque libera completamente il problema della acquisto di licenze per il software.

Nel porting, si sono mantenute tutte le caratteristiche di cui si è parlato nel capitolo 3.3, solo nell'aspetto della programmazione, abbiamo approfittato del porting per effettuare anche una ristrutturazione del codice a favore di una maggiore modularità, sfruttando il meccanismo della programmazione ad oggetti del C++.

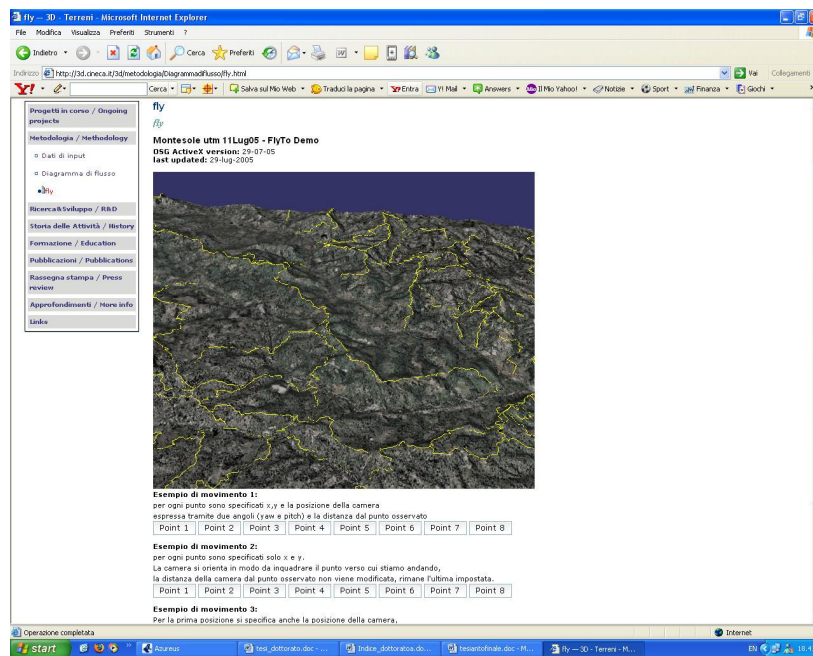


Figura 39: plug-in all'interno di Internet Explorer

6 Conclusioni e sviluppi futuri

La realizzazione del plug-in per Internet, apre al progetto Nu.M.E. infinite possibilità di divulgazione sia per la parte di grafica 3D, discussa nella presente tesi, che per possibili collegamenti ipertestuali che i moderni servizi web consentono.

In questa ottica, dunque, la pubblicazione del progetto Nu.M.E. attraverso il plug-in Visman costituisce soltanto il primo passo per la divulgazione di informazioni di carattere storico riguardanti il centro di Bologna nel Medioevo ed in altri periodi storici.

È possibile pensare, tramite il lavoro di eventuali altri ricercatori, di inserire il plug-in all'interno di un framework web per il reperimento e la pubblicazione di informazioni testuali, di immagini, ecc. provenienti da database relazionali, che si colleghino ed integrino la grafica 3D, come un'evoluzione della trasmissione del sapere umano, dalla carta stampata ai mezzi moderni che la tecnologia moderna consente.

La collaborazione tra specialisti di conoscenze diverse, la cosiddetta interdisciplinarietà ha qui la massima possibilità di esprimersi, perché i nuovi mezzi, resi utilizzabili da esperti in informatica e modellazione tridimensionale, sono disponibili a studiosi di materie storiche e ricercatori, affinché la ricerca abbia maggiore coinvolgimento ed efficacia cognitiva.

Inizia insomma una nuova fase per la ricerca legata ai beni culturali, di cui la presente tesi rappresenta solo l'inizio.

Glossario

API	È l'acronimo di Application Programming Interface, indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per un determinato compito. Le API permettono di evitare ai programmatori di scrivere tutte le funzioni dal nulla.
Campo	Il campo è l'unità di base per l'immissione dei dati in un record.
CAVE	(Cave Automatic Virtual Environment) ambiente di grafica immersiva in cui è possibile proiettare grafica 3D su tutte le pareti di una stanza accompagnato da un sistema <i>tracking</i> per l'interazione del pubblico.
Chiave primaria	Una chiave primaria è un campo di una tabella che contiene valori non ripetuti che identificano in maniera univoca ciascun record.
Database design	È il processo decisionale su come organizzare questi dati in tipi di record e su come ciascun tipo di record si relaziona con gli altri.
Database gerarchico	Un database gerarchico è un insieme di archivi. Gli archivi sono composti da record chiamati segmenti. I segmenti sono in rapporto gerarchico tra loro attraverso legami di tipo padre-figlio.
Database relazionale	Il database relazionale è un modello di database che consiste di diverse tabelle separate che sono correlate l'una all'altra attraverso campi chiave. Si può accedere alle informazioni memorizzate, grazie alle relazioni stabilite tra queste.
Database reticolare	La logica su cui si basa questo modello è basata su record e puntatori.
Dipendenza funzionale	Significa che i campi di una tabella devono

dipendere da una e una sola chiave primaria.

Framework	Nella produzione del software, il framework è una struttura di supporto su cui un software può essere organizzato e progettato. Alla base di un framework c'è sempre una serie di librerie di codice utilizzabili con uno o più linguaggi di programmazione.
Gamepad	È un dispositivo elettronico di input con diversi tasti.
GPS - Sistema di navigazione satellitare (Global Positioning System)	Sistema di rilevamento al suolo di coordinate geografiche e altimetria, mediante un ricevitore, software dedicato e una costellazione di satelliti.
Grafica raster	Termine raster è usato per indicare la tipologia di dato impiegata nella rappresentazione cartografica digitale. Con i dati raster il territorio viene riprodotto attraverso una matrice di pixel di forma quadrata o rettangolare. A ciascun pixel è associato un attributo che definisce le caratteristiche dell'elemento rappresentato.
Grafica vettoriale	Nella grafica vettoriale le informazioni vengono rappresentate attraverso un insieme di primitive matematiche che sono punti, linee, curve e poligoni opportunamente colorati. I principali vantaggi della grafica vettoriale rispetto alla grafica raster sono la qualità, la maggiore compressione dei dati e la più facile gestione delle eventuali modifiche.
GUI (graphical user interface)	L'interfaccia grafica, è un paradigma di sviluppo che mira a consentire all'utente di interagire col calcolatore manipolando graficamente degli oggetti.
HMD (Head mounted display)	Dispositivo hardware, la Head Mounted Device (Strumento montato in testa) usa una sorta di elmetto o di grossi occhiali per posizionare piccoli video displays di fronte a ciascun occhio, con una speciale ottica per mettere a fuoco e allungare il campo visivo percepito. La maggior parte degli HMD usa due displays e può fornire immagini

stereoscopiche. Altri usano un singolo display più grande per fornire una risoluzione più alta, ma senza la visione stereoscopica.

IMAX	Da Image Maximum, è un sistema di proiezione della pellicola cinematografica che ha la capacità di mostrare immagini con una grandezza ed una risoluzione molto superiore rispetto ai sistemi di proiezione classici. Le dimensioni standard di uno schermo IMAX sono 22 metri di lunghezza e 16 di altezza.
JDBC	È l'acronimo di Java DataBase Connectivity è un middleware per database che consente l'accesso alle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DBMS utilizzato.
Layer	Livelli informativi diversi appartenenti ad un unico contesto.
Metadato	Informazione che descrive un insieme di dati.
Multi piattaforma	Può essere detto di un linguaggio di programmazione, di un'applicazione software o di un dispositivo hardware che funziona su più di un sistema operativo (piattaforma).
ODBC	È l'acronimo di Open Database Connectivity è una API standard per la connessione ai DBMS. Questa API è indipendente dai linguaggi di programmazione dai sistemi di database e dal sistema operativo. È stata creata dall'SQL Access Group.
Open Source	Il termine inglese indica un software rilasciato con un tipo di licenza per la quale, il codice sorgente, è lasciato alla disponibilità di eventuali sviluppatori.
Pixel	Con il termine pixel (contrazione della locuzione inglese picture element) si indica ciascuno degli elementi puntiformi che compongono la rappresentazione di un'immagine raster in formato digitale.

Porting	Operazione di conversione di codice, da un linguaggio di programmazione ad un altro oppure da una libreria verso un'altra.
Query	Operazione che estrapola dati da un database.
Real time (Calcolo in tempo reale)	Il termine viene usato per quelle applicazioni dove le specifiche relative ai tempi di risposta sono preponderanti rispetto alla complessità dell'algoritmo da utilizzare.
Record	Tipo di dato composto da elementi eterogenei detti campi.
Rendering 3D	Identifica il processo di "resa" ovvero di generazione di un'immagine a partire da una descrizione matematica di una scena tridimensionale.
Report	E' la stampa dei dati, completa o parziale dei dati, di un database.
SceneGraph	Sistema capace di raggruppare e gestire gli oggetti composti da poligoni presenti nella scena.
Shutter glasses	Occhiali a cristalli liquidi usati per oscurare alternativamente gli occhi in sincronizzazione con il display. Quando il cervello riceve le immagini in successione sufficientemente rapida, fonde le immagini in una scena singola e ne percepisce la profondità.
SQL	È l'acronimo di Structured Query Language, è un linguaggio creato per l'accesso a informazioni memorizzate nei database.
Tabella	Insieme di record. Più tabelle formano un database relazionale.
Texture	Le immagini che rivestono le geometrie dei modelli al fine ottenere un effetto fotorealistico.
Vincoli di integrità	Sono l'insieme di regole che determinano l'integrità

referenziale

e la consistenza di una base di dati.

Virtual set

Si tratta di uno studio vuoto con una colorazione uniforme blu o verde. La scelta del colore non è casuale, in quanto il virtual set sfrutta una tecnica televisiva, conosciuta già da molti anni con il nome *chroma key*. Il salto di qualità, reso possibile dal virtual set rispetto al suo predecessore (*chromakey*), è costituito dal fatto che lo sfondo risulta indipendente rispetto agli attori posti sulla scena. Le fughe prospettiche e la profondità di campo sono ricalcolate automaticamente da un sistema informatico in funzione delle diverse inquadrature televisive che possono essere fisse o in movimento.

Bibliografia

Cok K., Commike A., Kuehne B., True T., "Developing Efficient Graphics Software", in: *Siggraph*, Vol.10, Los Angeles, Siggraph Course Notes, 1999, 1-78.

Eckel G., *IRIS Performer, Programmer's Guide*, Mountain View, Silicon Graphics Press, 1997.

Falby J., Zyda M., Pratt D., Mackey R., "Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation", in: *Computer & Graphics* 17(1), 1993, 173-182.

Foley, James D. *Computer Graphics : Principles and Practice*, New York, Addison-Wesley, 1990.

Funkhouser T., Sequin C., "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments", in: *Siggraph*, Vol.10, Los Angeles, Siggraph Course Notes, 1999.

Gouraud H., "Continuous shading of surfaces", in: *IEEE Transactions on Computers*, Giugno, 1971, 223-228.

Heckbert P., Garland M., "Multiresolution Modeling for Fast Rendering", in: *Proceedings of Graphics Interface '94*, 1994, 1-8.

Koh E., *IRIS Performer Programming Student Handbook*, Mountain View, Silicon Graphic Press, 1998.

McAllister D., *Stereo Computer Graphics and other true 3D Technologies*, Princeton, Princeton University Press, 1993.

Okoshi T., *Three-dimensional imaging techniques*, New York, New York Academic Press.

Piegl L., Tiller W., *The NURBS Book*, seconda edizione, Springer-Verlag, 1997.

Plenge A., *AMIGA grafica 3D e animazione*, Gruppo Editoriale Jackson, 1989.

Phong B., "Illumination for computer generated pictures", in: *CACM*, Giugno, 1975, 311-317.

Rogers D., *Procedural Elements of Computer Graphics*, McGraw-Hill, 1985.

Rohlf J., Helman J, “IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics”, in: *Computer Graphics Proceedings*, Annual Conference Series, ACM SIGGRAPH, 1993, 381-394.

Rossignac J., Borrel P., “Multi-resolution 3D approximation for rendering complex scenes”, in: *Modelling in Computer Graphics*, Springer-Verlag, 1993, 455-465.

Simrad Neptune Postprocessing system operator manual, base version, Simrad Norgeas Documentation Department, 1995.

Woo M., Neider J., Davis T., Shreider D., *OpenGL Programming Guide*, Third Edition, New York, Addison Wesley, 1999.

R. Scateni, P. Cignoni, C. Montani, R. Scopino, *Fondamenti di computer grafica tridimensionale interattiva*, Milano, McGraw-Hill