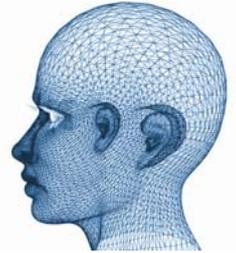


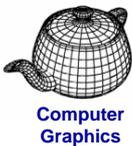
Java e OpenGL



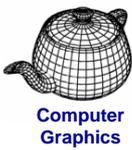
Ing. Luca Gardelli

lgardelli@deis.unibo.it
lgardelli@unibo.it

DEIS - Dipartimento di Elettronica Informatica e Sistemistica
Alma Mater Studiorum--Università di Bologna sede di Cesena
Via Venezia 52, 47023, Cesena (FC) Italy
Ph: +39 0547 339210 Fax: +39 0547 339208

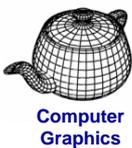


- OpenGL è un API per lo sviluppo di applicazioni grafiche 2D e 3D portabili www.opengl.org
- Dalla sua nascita – 1992 – OpenGL è lo standard più supportato dalle industrie – e.g. Videogiochi, CAD/CAM..
- L'architettura di OpenGL è progettata in modo da essere efficiente ed indipendente dal sistema operativo (portabilità)...
- ..per questo non include nessun comando dipendenti dalla piattaforma, e.g. per la creazione di finestre.



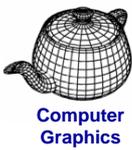
OpenGL e DirectX

- Direct3D è un layer software di astrazione dall'implementazione hardware specifica
- Specifico per i videogame
- Si appoggia ad OpenGL...
- Se l'hardware supporta una certa feature Direct3D la utilizza altrimenti si sfrutta il software



GLU – OpenGL Utility Library

- OpenGL Utility Library (GLU) è un libreria che utilizzando le OpenGL, fornisce delle funzionalità di più alto livello.
- Questa libreria include
 - Operazioni su matrici
 - Primitive di disegno



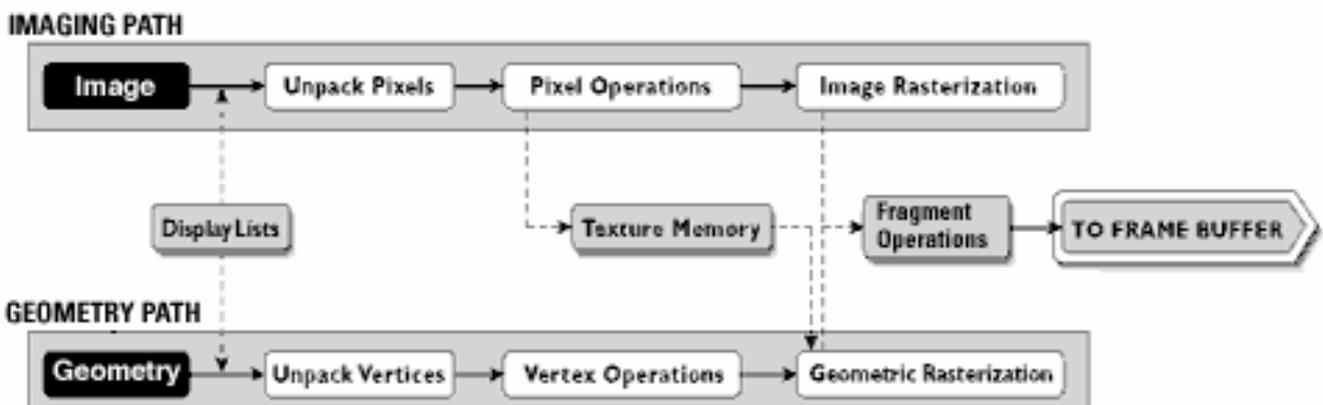
GLX – OpenGL Extension

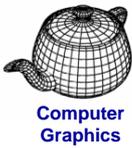
- GLX è una libreria che estende le capacità dei sistemi a finestre per fornire supporto alle OpenGL.
- Esiste una versione di tale libreria per i sistema operativi più diffusi..
 - Microsoft Windows : wgl
 - Unix & Co. : glx
 - Apple : agl
 - IBM OS/2 : pgl



OpenGL pipeline

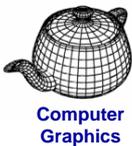
- OpenGL opera sia su immagini che su geometria, componendo i risultati per produrre l'immagine finale





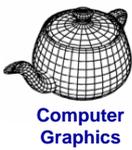
Display list

- Sia che i dati siano pixel o geometria
 - possono essere elaborati subito : *immediate mode*;
 - possono essere memorizzati nella display list: *retained mode*.
- Quando è eseguita la display list, i dati vengono inviati come se fossero stati spediti dall'applicazione



Evaluators

- Tutte le primitive geometriche sono descritte da vertici.
- Le curve e le superfici parametriche possono essere descritte in altri modi (e.g. punti di controllo).
- Gli *evaluators* traducono questa descrizione in un insieme di vertici.



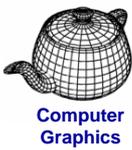
Per-vertex operations

- I vertici vengono convertiti in primitive.
- Alcune operazioni che vengono eseguite in questo stadio sono
 - Trasformazione delle coordinate dalla matrice di visualizzazione
 - Generazione delle coordinate delle texture
 - Calcolo dell'illuminazione



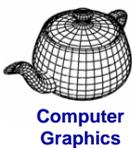
Primitive assembly

- Questo stadio si comporta in modo differente in base alla primitiva (e.g. il clipping)
- Altre elaborazioni riguardano
 - Calcolo della profondità
 - Calcolo dell'antialiasing



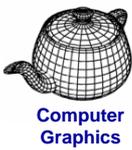
Pixel Operations

- I pixel vengono tradotti nel formato più appropriato.
- I dati sono elaborati (e.g. scalatura) in base ad una mappa di pixel.



Texture assembly

- Questa fase vengono applicate le texture alla geometria per renderla più realistica.
- Alcuni dispositivi potrebbero avere capacità di elaborazione delle texture per ottimizzare le prestazioni.
- Le texture possono essere immagazzinati in una zona di memoria dedicata oppure in memoria centrale (meccanismi di priorità).



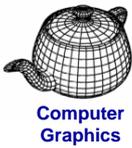
Rasterization

- E' il processo che converte dati geometrici e pixel in frammenti.
- Ogni frammento corrisponde ad un pixel nel framebuffer.



Fragment operations

- Prima di scrivere i frammenti nel framebuffer vengono eseguite alcune operazioni (che possono essere disabilitate).
- Queste operazioni includono
 - Calcolo dei texel
 - Calcolo della nebbia
 - Antialiasing
 - Test di profondità, trasparenza...



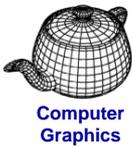
Da OpenGL a JOGL..

- JOGL – Java OpenGL Binding si inserisce in un contesto di tecnologie open source creato dal Game Technology Group di Sun Microsystems.
<https://jogl.dev.java.net/>
- JOGL – eredità di Jungle - è stato sviluppato con l'obiettivo di fornire alle applicazioni scritte in java, l'accesso all'accelerazione grafica hardware
- JOGL incapsula le librerie OpenGL e le ripropone in un'ottica object-oriented
- Il nucleo di JOGL consiste in un “binding” delle funzioni di OpenGL tramite Java Native Interface
- In JOGL sono comprese – oltre ad OpenGL – anche le librerie GLU e GLUT, e alcune estensioni dei produttori di schede grafiche



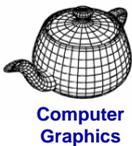
JOGL – per iniziare..

- Per utilizzare JOGL occorre scaricare dal sito web ufficiale
 - la libreria jogl.jar da includere nei progetti
 - il binding per il sistema operativo utilizzato – supportati Windows, Solaris, Linux e MacOSX
 - le librerie native vanno inserite nella cartella di sistema appropriata – e.g. /windows/system32
- Si consiglia anche di fare riferimento anche a
 - Demo di jogl
 - NeHe OpenGL tutorials <http://nehe.gamedev.net>
 - Codice JOGL dei NeHe tutorials
<http://pepijn.fab4.be/nehe/>



JOGL - architettura

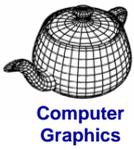
- Ogni applicazione JOGL deve almeno utilizzare Animator, GLEventListener e GLCanvas
- L'[Animator](#) incapsula un Thread e pilota la pipeline OpenGL
- [GLEventListener](#) è un'interfaccia che deve essere implementata dal programmatore in base ad eventi OpenGL standard
- [GLCanvas](#) è la zona client di disegno che sfrutta l'accelerazione grafica



Appunti di

Computer Grafica

Bibliografia e Links



Programmazione

- www.gamedev.net
- www.gamasutra.com
- www.flipcode.com
- www.java.net
- <http://java.sun.com>
- <http://www.jars.com>
- www.opengl.org
- <http://www.xmission.com/~nate/index.html>
- <https://jogl.dev.java.net>