Immagine che contiene testo, segnale

Descrizione generata automaticamente**OpenGL**

Open Graphics Library

È una piattaforma cross-Platform e cross-Language, con api per molti linguaggi.

È un API astratta, può essere implementata sia in hardware che software.

Gestito dal consorzio Khronos.

OpenGL si occupa solo di RENDERING

OpenGL è una specifica su come creare un API grafica.

OpenGL non è uno standard ISO

OpenGL è installato nel sistema

Usare OpenGL bisogna inizializzare un **contesto:** un contesto è un contenitore dove sono allocati dai driver per immagazzinare lo stato delle API e le risorse grafiche usate dalla finestra dall’applicazione.

* glGetString(): stampa informazioni della GPU
  + - * + GL\_VERSION
  + GL\_VENDOR
  + GL\_RENDERER
  + GL\_EXTENSIONS

**FreeGLUT**

È una libreria con semplifica l’uso di OpenGL

Feature list:

* Event-based (callbacks).
* Supports user-input through mouse and keyboard.
* Provides a menu/sub-menu system.
* Supports printing text to the OpenGL window.
* Provides a series of built-in, dynamically generated 3D objects.
* Available on Windows, Linux, MacOS, etc.

Typical usage:

* Initialize the library.
* Create a window for graphic output.
* Register callback functions.
* Enter the main loop.

**Main Buffers**

Sono delle aree di memoria che contengono informazioni legate ai pixel renderizzati su schermo.

**Framebuffer**: contiene le immagini da inviare allo schermo

**Back buffer**: è come il framebuffer, ma non viene visualizzato e viene usato per evitare problemi di refresh

Questa area viene usata per disegnare e successivamente si passa al framebuffer, altrimenti si vedrebbe flickering dell’immagine, aggiornamento in tempo reale del rendering

**Depth buffer o Z buffer:** immagazzina il valore z per ogni pixel

**Framebuffer**

I colori hanno diverse codifiche

Immagine che contiene testo

Descrizione generata automaticamente

Alla memoria immagazzinata nel framebuffer si accede dal dispositivo per aggiornare periodicamente i pixel renderizzati sullo schermo: refresh monitor 60-120/144Hz

Per evitare artefatti visivi (sfarfallio e tearing dello schermo), l'aggiornamento della memoria video è sincronizzato con la frequenza di aggiornamento del display.

Il v-sync sincronizza il framebuffer con il monitor

Un'altra soluzione è usare il double buffering (front e back buffer)

Si renderizza tutto il contenuto sul backbuffer ed una volta terminato il rendering della scena openGL copia il contenuto del back nel front(framebuffer). In pratica non vengono copiati i dati da una memoria all’altra, ma si swoppano i puntatori facendo un Ping-Pong tra i due buffer, ciò favorisce le prestazioni.

Immagine che contiene testo

Descrizione generata automaticamente

glColor3f(r,g,b) è una macchina a stati, se n on si specifica prima del vertice si tiene il precedente (glColor imposta il colore)

glVertex3f(x,y,z) imposta un vertice

bisogna metterei la sequenza della funzioni tra glBegin(primitiva da renderizzare) e glEnd()

Primitive

Immagine che contiene testo

Descrizione generata automaticamente

Gl\_triangle\_fan crea triangoli prendendo l’ultimo vertice del triangolo precedente, il primo vertice inserito in tutto il processo e il nuovo vertice, utille se si vuole un centro comune (solidi)

le coordinate dei vertici sono in object coordinates, non bisogna moltiplicarli per le metrici, ci pensa openGL.

glViewport(originx, originy, width, height) imposta la dimensione della finestra, questo va richiamato ogni volta che la finestra viene ridimensionata.

Immagine che contiene testo

Descrizione generata automaticamente

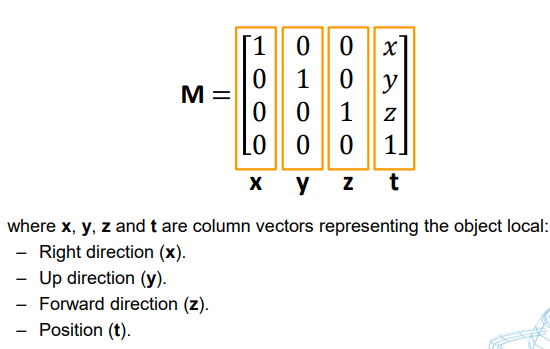


Immagine che contiene testo

Descrizione generata automaticamente

Questo metodo è utile per creare la matrice della camera (ritorna l’inversa, “da accertare”)

Il parametro up definisce la rotazione della camera nel mondo tipicamente (0,1,0)