

FONDAMENTI DI COMPUTER GRAPHICS LM

LAB 2 - NAVIGAZIONE INTERATTIVA IN SCENA CON MODELLI GEOMETRICI 3D

Questa esercitazione può essere eseguita sia in ambiente Windows che Linux. Dopo aver scaricato i file necessari, compilare ed eseguire il programma fornito.

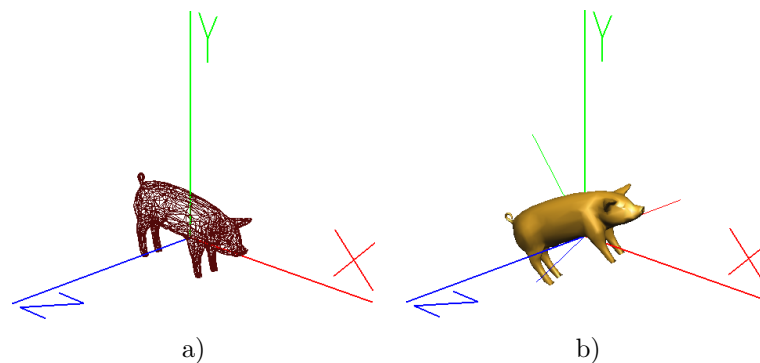


Figure 1: a) Uno screenshot dell'applicazione model_viewer; b) il rendering in model_viewer del modello pig.m in modalità smooth dopo aver sviluppato il calcolo delle normali ai vertici, attivato lo shading smooth, cambiato il materiale e ruotato il modello.

1. model_viewer - Caricamento e visualizzazione modelli geometrici

L'applicazione model_viewer carica e visualizza sia quadriche predefinite in OpenGL GLU library, sia oggetti mesh poligonali memorizzati in file con estensione .m. La directory data contiene vari file di modelli a mesh poligonali.

Il programma model_viewer.c fornito presenta un menu pop up (a tendina) che si apre al click del bottone destro del mouse sulla finestra grafica e fornisce un ambiente interattivo di caricamento e visualizzazione di modelli geometrici 3D, sia primitive GLU che mesh poligonali in formato .m. Estendere il programma secondo le seguenti specifiche:

- (a) Caricamento e visualizzazione modelli geometrici di tipo mesh in formato .m
- (b) Visualizzazione superfici quadriche dalla libreria GLU (es. Sfere, cilindri, tori)
- (c) Verifica della gestione della visualizzazione dei modelli poligonali a mesh tramite display list.
- (d) Calcolo e memorizzazione delle normali ai vertici per i modelli mesh poligonali. Visualizzazione con normali ai vertici in modalità smooth (`glShadeModel(GL_SMOOTH)`).

2. model_viewer - Controllo interattivo della scena

Alla pressione del tasto destro del mouse compare un menu pop up. Si può interagire con la scena selezionando gli item del menu e successivamente, tramite input da tastiera, modificare i valori associati. Le funzionalità che il sistema interattivo deve fornire sono elencate nella tabella 1, alcune

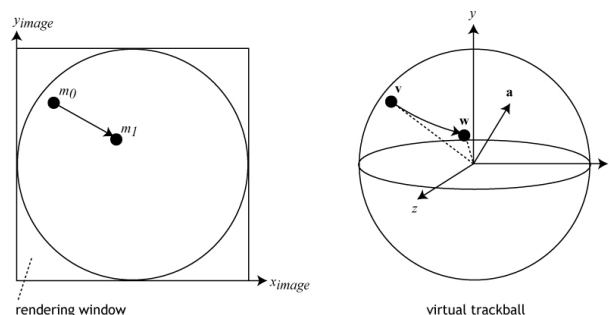


Figure 2: La trackball virtuale traduce il movimento del mouse in una matrice di rotazione

sono già realizzate (prendete visione di come) e altre invece sono lasciate da realizzare (indicate in grassetto). La trackball è già implementata, bisogna solo renderla **SEMPRE** attiva.

Aggiungere, ove necessario, le voci nel menu pop up. *Nota: A lettere piccole corrisponde un aumento del valore parametro, a lettere grandi una diminuzione.*

3. Manipolazione dello stack delle matrici di trasformazione

Modificare l'applicazione model_viewer sviluppando i seguenti punti:

- posizionare almeno tre diversi modelli geometrici nella scena sfruttando le funzioni OpenGL: `glPushMatrix()`, `glPopMatrix()`.
- permettere la traslazione e rotazione dei **singoli** oggetti rispetto ai sistemi di riferimento WCS, OCS (World e Object Coordinate Systems). I singoli oggetti devono essere selezionabili tramite i tasti 1, 2, 3, etc. . Le trasformazioni di traslazione o rotazione devono essere applicate tramite i tasti 'x', 'X', 'y', 'Y', 'z', 'Z'. La selezione del tipo di trasformazione (traslazione / rotazione) e del sistema di riferimento rispetto al quale eseguire tale trasformazione deve essere selezionabile tramite i tasti:
 - 'o' traslazione rispetto all'OCS
 - 'O' rotazione rispetto all'OCS
 - 'w' traslazione rispetto al WCS
 - 'W' rotazione rispetto al WCS

4. **OPZIONALE:** Manipolazione dello stack delle matrici

Modificare l'applicazione model_viewer per permettere la traslazione e rotazione dei **singoli** oggetti rispetto al sistema di riferimento VCS (View Coordinate System) seguendo le modalità del punto 3. Il tipo di trasformazione (traslazione / rotazione) deve essere selezionabile tramite i tasti:

- 'v' traslazione rispetto al VCS
- 'V' rotazione rispetto al VCS (con assi di rotazione aventi direzione del VCS e passanti per l'origine del VCS oppure direzione del VCS e passanti per l'origine dell'OCS)

²La trackball virtuale permette di ruotare un oggetto (o meglio girare attorno ad un oggetto) interattivamente utilizzando il mouse. La figura 2 illustra intuitivamente come tradurre il movimento del mouse in un asse ed un angolo di rotazione. m_0 ed m_1 sono due posizioni consecutive del mouse e definiscono due punti v e w sulla semisfera 3D virtuale. Il loro prodotto vettoriale definisce l'asse di rotazione $a = v \times w$, mentre l'angolo di rotazione può essere calcolato dal loro prodotto scalare.

²Muovere la camera virtuale lungo un percorso in modalità *look at* (mediante curva chiusa di Bézier) in modo che la camera si muova lungo un percorso con centro di interesse l'oggetto in scena

<i>Funzionalità</i>	<i>Descrizione</i>	<i>Tasti</i>	<i>Default</i>
Change eye point	modifica le coordinate del punto di vista	x,X,y,Y,z,Z	(8.8, 4.9, 9.0)
Change ref. point	modifica le coordinate del punto di riferimento del sistema	x,X,y,Y,z,Z	(0.0, 0.0, 0.0)
Change up vector	modifica le coordinate del vettore "alto" della camera	x,X,y,Y,z,Z	(0.0, 1.0, 0.0)
Change light position	modifica le coordinate del vettore posizione della fonte luminosa	x,X,y,Y,z,Z	(5.0, 5.0, 5.0)
Rotate model	ruota il modello	x,X,y,Y,z,Z	(0.0, 0.0, 0.0)
Zoom in/out	modifica l'angolo (field of view) al vertice della piramide di vista	f,F	20.00
Proiezione	modifica il tipo di proiezione e quindi il volume di vista	-	prospettica
Culling	abilita/ disabilita il back face culling	-	off
Wireframe	cambia modalita' di rendering dei poligoni	-	GL_LINE
Shading	Cambia modalita' di shading	-	GL_FLAT
Materials	Cambia il materiale associato agli oggetti	-	red.plastic
Trackball	modalità trackball virtuale per il controllo della camera. Vedi nota ¹	-	-
Camera Motion	movimento della camera lungo un percorso. Vedi nota ²	-	-
Print system status	stampa sul terminale i dati correnti del sistema	-	-
Reset	riporta il sistema allo stato iniziale	-	-
Quit	chiude il programma	Esc	-

Table 1: Tabella delle funzionalità sviluppate e da sviluppare (indicate in grassetto) nell'applicazione model_viewer