

Esercitazione 3 - Navigazione interattiva in scena con modelli geometrici 3D

Lorenzo Gasparini
Corso di fondamenti di Computer Graphics

2 settembre 2019

1 Descrizione

L'applicazione fornisce all'utente un ambiente interattivo 3D per visualizzare modelli geometrici 3D e effettuare trasformazioni base quali traslazione, rotazione e scaling rispetto a WCS e OCS. Inoltre l'utente ha anche a disposizione un meccanismo di navigazione della scena.

2 Obiettivi

1. **Caricamento e visualizzazione modelli mesh poligonali** Estendere il programma secondo le seguenti specifiche:
 - (a) Caricamento e visualizzazione di pi'ù di un oggetto mesh .obj con la possibilit'a di passare la selezione dall'uno all'altro tramite special key arrows.
 - (b) Verifica della gestione della visualizzazione dei modelli poligonali a mesh tramite VAO/VBO
 - (c) Calcolo e memorizzazione delle normali ai vertici per i modelli mesh poligonali. Visualizzazione con normali ai vertici in modalit'a smooth.
 - (d) Permettere il cambio di materiale dell'oggetto da pop-up menu.
2. **Navigazione interattiva in scena** Le funzionalità che il sistema interattivo deve fornire sono elencate nella tabella 1, alcune sono già realizzate (prendete visione di come) e altre invece sono lasciate da realizzare (indicate in grassetto). Aggiungere, ove necessario, le voci nel menu pop up.

NOTA: Middle Mouse Button (MMB) è WHEEL, mentre il Left Mouse Button è LMB.

Trackball (modalit'a trackball virtuale) per il controllo della camera. La trackball virtuale permette di navigare in scena interattivamente utilizzando il mouse.

Azione Trackball: CLICK LMB down + drag + RELEASE LMB).

3. **Trasformazione degli oggetti in scena** Si richiede di gestire opportunamente lo stack delle matrici di MODELVIEW per permettere le trasformazioni di traslazione, rotazione e scalatura dei singoli oggetti in scena rispetto al sistema di riferimento WCS o OCS (World e Object Coordinate Systems).

La selezione del sistema di riferimento (WCS o OCS) rispetto al quale eseguire tale trasformazione avviene mediante menu pop-up.

I singoli oggetti devono essere selezionabili tramite i tasti frecce dx/sx.

Gli assi x,y,z rispetto ai quali effettuare la trasformazione sono selezionabili da keyboard.

La selezione del tipo di trasformazione avviene da keyboard tramite i tasti:

- 'g' modalità traslazione
- 'r' modalità rotazione
- 's' modalità scalatura

La quantità di spostamento è incrementale/decrementale e controllata da WHEEL up(+)/down(-).

3 Svolgimento Obiettivi

- 1.a. Le mesh sono caricate all'interno della funzione MeshLoading, la quale inizializza un oggetto di tipo Object con i vari valori della mesh e lo pusha all'interno dell'array delle mesh. Successivamente tutti gli oggetti sono renderizzati nella funzione display che li scorre uno ad uno e li renderizza a video.

La selezione dell'oggetto avviene tramite le frecce dentro la funzione special. Semplicemente si varia l'index puntato dall'intero selectedObject.

- 1.b. Tendina già implementata, varia solamente le trasformazioni che vengono descritte in seguito
- 1.c. Le normali vengono salvate all'interno della funzione generate_and_load_buffers. Il calcolo invece avviene dentro loadObjFile .
- 1.d. La tendina era già implementata. Semplicemente quando l'utente seleziona il materiale lo si imposta nell'oggetto attualmente selezionato. Tale impostazione viene effettuata dentro material_menu_function.

2. Il pan orizzontale della camera avviene tramite la funzione moveCameraRight e moveCameraLeft.

Il pan verticale della camera avviene tramite le funzioni moveCameraUp e moveCameraDown.

Lo zoom è effettuato tramite le funzioni moveCameraForeward e moveCameraBack che spostano la camera lungo la direzione di vista (determinata tramite la formula posizione del target meno la posizione della camera).

Per il culling semplicemente si attiva e disattiva il valore `GL_CULL_FACE` tramite `glEnable` e `glDisable`.

Per la variazione tra smooth e flat shading si richiama `glShadeModel` con il valore `GL_FLAT` o `GL_SMOOTH` in base a quale vogliamo.

Il camera motion è implementato attraverso la funzione `CameraMotion`. Il suo funzionamento è molto semplice, praticamente si definisce una curva di bezier e la si valuta tramite `deCasteljau`. Ad ogni passo si imposta il punto ottenuto come posizione della camera. Per riuscire ad implementare questa funzione è stata sfruttata la funzione di libreria `glutTimerFunc`.

3. Per la trasformazione degli oggetti in scena si usa la funzione `modifyModelMatrix`. Questa funzione prende in input il vettore di traslazione, di rotazione, l'angolo ed il fattore di scaling. Successivamente si sfruttano le funzioni di libreria `glRotatef`, `glScalef`, `glTranslatef` per effettuare le modifiche nella matrice già presente.

La differenza tra WCS e OCS sta nell'ordine delle operazioni, per il primo caso si effettuano le modifiche e poi si moltiplica con la matrice già presente nell'oggetto, nel secondo caso invece si moltiplica tutto per la matrice presente nell'oggetto e poi si applicano le ulteriori modifiche.

Terminato tutto si salva la matrice risultante come matrice dell'oggetto.

