

Grafica

Anno Accademico 2018-2019

Esercitazione 3

Matteo Berti, Matricola 889889

26 giugno 2019

Comandi da tastiera

Tasto	Azione
Esc	Terminazione dell'esecuzione
g	Traslazione oggetto selezionato
r	Rotazione oggetto selezionato
s	Ridimensionamento oggetto selezionato
Spazio	Animazione camera
x	Selezione asse x
y	Selezione asse y
z	Selezione asse z
Scroll	Zoom
Ctrl + Scroll	Pan orizzontale
Shift + Scroll	Pan verticale
⇒	Selezione oggetto successivo
⇐	Selezione oggetto precedente

Note: il codice è stato generato e testato utilizzando le seguenti specifiche:

- OS: Debian GNU/Linux 9 (stretch)

- IDE: CLion 2018.3.4

All'interno della funzione `init()` viene settata la working directory corrente, se si utilizzano parametri diversi da quelli indicati è possibile sia necessario riadattarla.

2. Navigazione interattiva in scena

- **Pan orizzontale:** dati d il vettore che dalla camera punta alla posizione guardata e u l'up vector della camera, si calcola la normale n al piano generato da questi due vettori tramite prodotto vettoriale: $n = d \times u$. Spostando la camera e il *lookAt* lungo la normale calcolata si ottiene un effetto pan orizzontale.
- **Pan verticale:** dati d il vettore che dalla camera punta alla posizione guardata e u l'up vector della camera, si calcola la normale n al piano generato da questi due vettori tramite

prodotto vettoriale: $n = d \times u$. A questo punto viene calcolata la normale al piano generato dai vettori n e d trovando la direzione di panning verticale. Spostando la camera e il *lookAt* lungo quest'ultima normale calcolata si ottiene un effetto pan verticale.

- **Zoom camera:** per il calcolo dello zoom ci si basa sulla formula: $P_1 = P_0 + t \cdot v$ in cui P_0 è il punto di partenza, v è il vettore che indica la direzione dello zoom, ovvero dalla camera alla posizione guardata, e t uno scalare che indica la "quantità" di zoom. Infine il punto P_1 ottenuto sarà la posizione che assumerà la camera per realizzare lo zoom.
- **Culling:** il culling è selezionabile tramite il menu accessibile con clic destro del mouse. L'abilitazione o meno avviene tramite `glEnable(GL_CULL_FACE)` e `glDisable(GL_CULL_FACE)`.
- **Modalità di rendering:** la scelta di una modalità di rendering avviene tramite `glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)` per la modalità *wireframe* e `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)` per la modalità *face fill*.
- **Modalità di shading:** la scelta di una modalità di shading avviene tramite `glShadeModel(GL_FLAT)` per lo shading *flat* e `glShadeModel(GL_SMOOTH)` per lo shading *smooth* tramite il modello Gouraud.
- **Camera motion:** per prima cosa vi è un'animazione che porta al centro della visualizzazione della camera l'oggetto selezionato, questo avviene suddividendo in 50 porzioni la distanza tra il centro di visualizzazione della camera e l'oggetto da puntare e impostando il *lookAt* della camera ad ognuno di questi punti sequenzialmente. Poi viene calcolata una curva Bezier con 5 punti di controllo, valutata tramite l'algoritmo di deCastejau. I punti di controllo sono disposti a cerchio intorno all'oggetto selezionato: il primo nella posizione attuale della camera, il secondo a 90°, il terzo a 180°, il quarto a 270° e infine l'ultimo nuovamente nel punto di partenza della camera, per chiudere la curva. L'asse y del secondo e quarto punto è la stessa, mentre per il primo/quinto e terzo è inversa. Terminata la transazione della camera attorno all'oggetto selezionato vi è un'ultima animazione, inversa a quella iniziale, che riporta il focus della camera alla posizione iniziale.

3. Trasformazione degli oggetti in scena

All'interno di `modifyModelMatrix()` vengono gestite le matrici *ModelView* e l'operazione da effettuare (transazione, rotazione e ridimensionamento). Se il sistema di coordinate è *WCS* avviene prima la trasformazione poi la moltiplicazione della matrice corrente con la matrice del modello, se invece il sistema è *OCS* le due operazioni sono invertite. Viene gestito anche il posizionamento iniziale degli oggetti e mantenuto tramite `glTranslatef()`, in posizioni diverse tra loro definite nell'array `objectsPos`.