Relazione laboratori

Corso di Computer Graphics - Università di Bologna

Daniele Polidori
daniele.polidori2@studio.unibo.it

a.a. 2022-23

1 LAB-01

Punto 4b Nello svolgimento di questo primo laboratorio, per quanto riguarda il punto 4, ho scelto di svolgere l'opzione *b*.
Oltre ai VAO già in uso per la curva di base, ho creato una seconda coppia di VAO per disegnare il tratto cubico che di volta in volta viene creato. Questo, una volta completato, viene attaccato alla curva di base: i punti relativi al tratto cubico vengono aggiunti a quelli della curva di base, così da liberare lo spazio per un possibile nuovo tratto cubico, e così via.
L'utente può scegliere la continuità con cui attaccare i tratti cubici alla curva di base: la continuità C^0 premendo il tasto *0* (scelta di default), C^1 premendo il tasto *1* e G^1 premendo il tasto *g*.
I tratti successivamente creati verranno raccordati con la continuità selezionata. Siano $p_0 \dots p_n$ i punti della curva di base e $v_0 \dots v_3$ i punti del tratto cubico, le continuità vengono applicate nella maniera seguente. v_0 viene eliminato, sarà infatti sostituito da p_n. A partire dalle coordinate di p_{n-1} e p_n vengono calcolate le coordinate che deve assumere il punto v_1, così che la continuità venga soddisfatta. Viene applicata la formula (1) per ottenere la continuità C^1 e la formula (2) per la continuità G^1: tali formule vengono applicate separatamente sulle coordinate x e sulle y dei rispettivi punti, con $\Delta = p_3 - p_2$. Per la continuità G^1, il punto verrà posizionato in modo tale che il tratto $v_1 - p_n$ sia lungo la metà del tratto $p_n - p_{n-1}$. Infine, i punti v_1, v_2 e v_3 vengono aggiunti alla curva di base, in qualità di, rispettivamente, p_{n+1}, p_{n+2} e p_{n+3}.

$$v_1' = p_n + \Delta = 2 * p_n - p_{n-1} \tag{1}$$

$$v_1' = p_n + \frac{\Delta}{2} = p_n + \frac{p_n - p_{n-1}}{2} \tag{2}$$

Punto 5 Per realizzare lo spostamento dei punti tramite trascinamento con il mouse, mi sono servito delle funzioni callback di OpenGL: `glutMouseFunc` e `glutMotionFunc`.
Tramite la prima, quando viene premuto il tasto destro del mouse, controllo se mi trovo sopra un punto. Per facilitare la presa, considero un intorno delle coordinate dei punti: dato un punto (x, y), anziché le coordinate strette x e y, cerco dei valori più laschi:

$$x - 0.01 < x' < x + 0.01$$
$$y - 0.01 < y' < y + 0.01$$

Se il mouse si trova sopra un punto, questo viene "agganciato" (i.e. un puntatore punta alla sua cella di memoria) e quando poi rilascio il tasto, il punto viene "sganciato".
Tramite la seconda, invece, catturo la posizione del mouse durante il trascinamento del punto. Se un punto è stato agganciato, sostituisco le coordinate di tale punto con le coordinate del mouse in quell'istante. I valori continuano a mutare fino a che il punto non viene sganciato, ovvero finché il tasto del mouse non viene rilasciato.