

06 - Trabalho 2: Animação e superfícies curvas

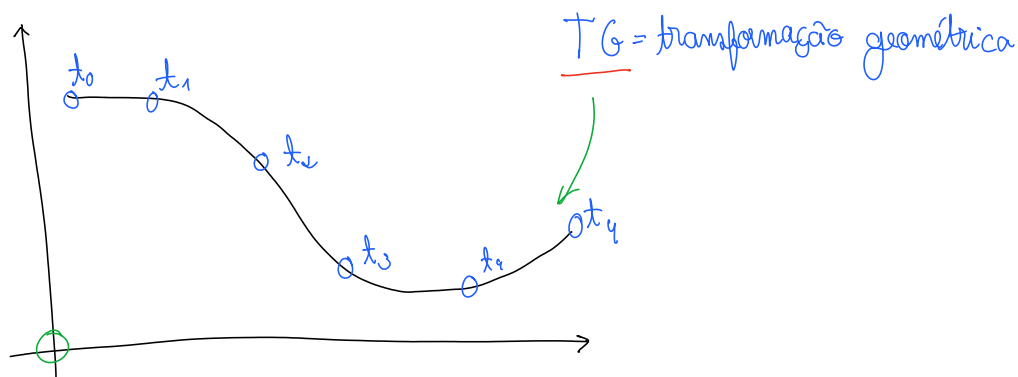
28 de outubro de 2016 11:47

Animação

- Procedimental (usada no trabalho 2)
- Por chaves (trabalho 3)

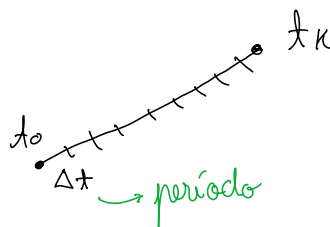
Animação procedimental

cintilar = flickering (a imagem tremelicar, por causa do frame rate baixo)



- O ponto pivot (de referência) de um objeto deve estar na origem.

$$TG = f(t)$$

Translação $f(t):$ 

- T e R são operações aditivas. $T = f_T(t)$

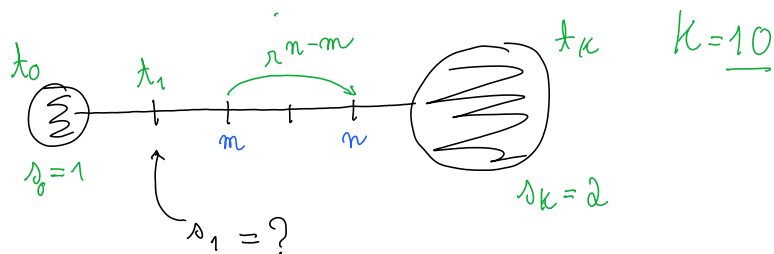
- S é multiplicativo.

Rotação

$$R = f_R(t)$$

Escalamento

$$S = f_s(t)$$



• s_1 seria igual a $1,2$ ($1 + (2/10)$), mas como o escalamento é multiplicativo, o resultado seria $(1,2)^{10}$.

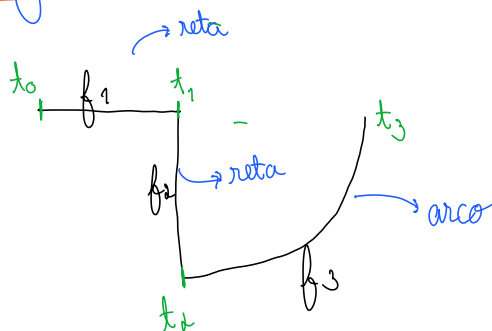
• Assim, $s_1 = r \cdot s_0 \Rightarrow s_1 = r$

$$r = \sqrt[k]{S_k}$$

$T_{k+1} = r \cdot T_k \Rightarrow s_{k+1} = r$

↑ tamanho

Trajectoria



uma única função composta

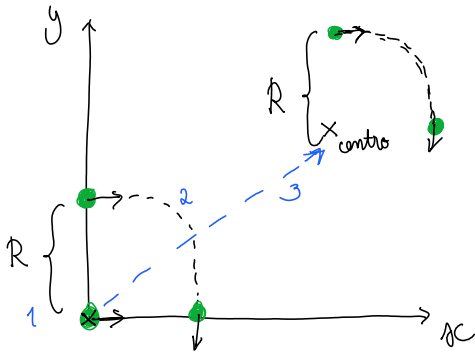
$$f(t) = \begin{cases} f_1(t), & t \in [t_0; t_1[\\ f_2(t), & t \in [t_1; t_2[\\ f_3(t), & t \in [t_2; t_3[\end{cases}$$

Função de animação

- Entra um instante. O argumento é o tempo.
- Calcula-se a transformação e devolve-se.

- A transf. da animação é escrita abaixo da transf. do objeto.
- Em código, elas são aplicadas de baixo para cima.

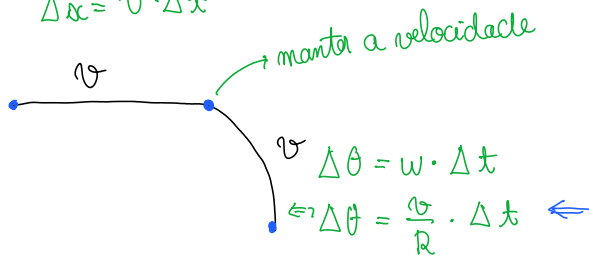
Animação arco de circunferência



$1^{\circ}: T(0; R; 0)$ • tudo está presente no DSX
 $2^{\circ}: R_z(\theta)$ conhece-se as coordenadas do centro
 $3^{\circ}: T(x_c, y_c, z_c)$

$$M = T(x_c, y_c, z_c) \cdot R_z(\theta) \cdot T(0; R; 0)$$

$$\Delta x = v \cdot \Delta t$$



$$v = \omega \cdot R \Leftrightarrow \omega = \frac{v}{R} ; \theta = \frac{d}{\Delta t}$$

- No DSX, deve-se definir o tempo da animação de forma a não haver discrepância de velocidades ao fazer transições.

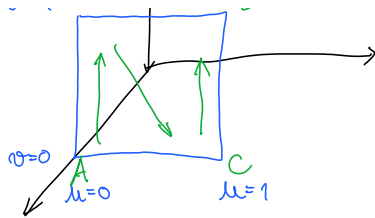
Superfícies

/a arc / webgl / laig - classes / nurbs - surfaces

<http://paginas.fe.up.pt/~avc/webgl/laig-classes/nurbs-surfaces/>

Desenho

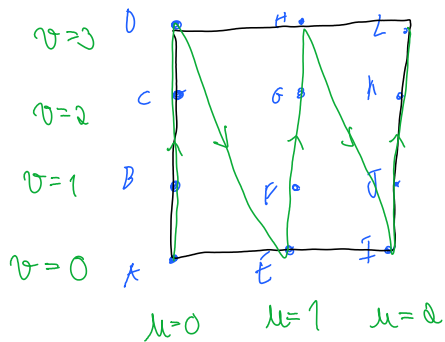
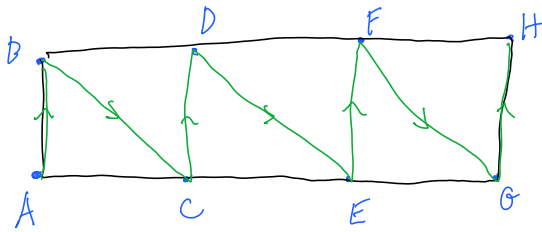




$$2^{\circ}: \quad 0 \quad 1$$

$$3^{\circ}: \quad 1 \quad 0$$

$$4^{\circ}: \quad 1 \quad 1$$



• Os array v estão dentro dos arrays u,
daí que a ordem seja a que está ao lado.

contribution

CGF multi surface → knots fora dos argumentos, a ser calculados
dentro da função