

10/15/2012

FACULTATEA
DE
AUTOMATICA SI
CALCULATOARE

ELEMENTE DE GRAFICA PE CALCULATOR



Laborator 2

1. 3D vs 2D

Diferenta de o dimensiune intre bidimensional si tridimensional aduce cu sine o crestere semnificativa a complexitatii. Daca intr-un spatiu 2D puteam sa am doar un tip de rotatie acum exista trei tipuri de rotatii si mai mult de atat conteaza si ordinea in care acestea sunt aplicate. Astfel avem rotatii in planul XoY (relative la axa OZ), in planul XoZ(relative la axa OY) si in planul YOZ(relative la axa OX).

Observatie: in cadrul acestui laborator nu se lucreaza cu proiectii si transformari fereastra poarta. Acestea vor fi explicate intr-un alt laborator.

2. Transformarile

Intr-un spatiu 3D exista mai multe tipuri de transformari:

Translatie:

$$x' = x + tx;$$

$$y' = y + ty;$$

$$z' = z + tz;$$

Rotatie relativa la OX:

$$x' = x$$

$$y' = y * \cos(u) - z * \sin(u)$$

$$z' = y * \sin(u) + z * \cos(u)$$

Rotatie relativa la OY:

$$x' = x * \cos(u) - z * \sin(u)$$

$$y' = y$$

$$z' = x * \sin(u) + z * \cos(u)$$

Rotatie relativa la OZ:

$$x' = x * \cos(u) - y * \sin(u)$$

$$y' = x * \sin(u) + y * \cos(u)$$

$$z' = z$$

Rotatia relativa la un punct oarecare se rezolva in cel mai simplu mod prin:

- translatarea atat a punctului asupra carui se aplica rotatia cat si a punctului in jurul caruia se face rotatia a.i. cel din urma sa fie originea sistemului de coordonate,
- rotatia normala,
- translatarea rezultatului a.i. punctul in jurul caruia s-a facut rotatia sa ajunga in pozitia sa initiala

Scalare:

$$x' = x * \text{factorx};$$

$y' = y * \text{factory};$

$z' = z * \text{factorz};$

Daca $\text{factorx} = \text{factory}$ atunci avem scalare uniforma, altfel avem scalare neuniforma. Scalarea relativa la un punct oarecare se rezolva similar cu rotatia relativa la un punct oarecare.

3. Framework-ul.

Singurele fisiere ca va sunt necesare atat pentru laborator sunt main.cpp si Transformation3d.cpp. In rest nu trebuie sa intelegeti nimic, desi sunteti incurajati sa explorati! Din motive de a va permite explorarea am ales sa nu incarcam codul cu toate transformarile posibile, deci, de exemplu nu sunt implementate rotatiile relative la axele sistemelor de coordonate transformate.

- Frameworkul este initializat prin constructor astfel:

```
WorldDrawer3d  
wd3d(argc,argv,window_width,window_height,window_width_initial_positi  
on,window_height_initial_position,nume_afisat_ca_titlu_pe_fereastră);
```

Este aproape identic cu cel folosit la lumea 2D.

- Functia de initializare este apelata o singura data, inainte de a se incepe procesul de desenare. In aceasta functie va este indicat sa initializati toate obiectele cu care lucrati.

```
wd2d.init();
```

- O functie ce se executa o data per cadru.
- O functie ce se executa de fiecare data cand este apasata o tasta.

In Transformation3d.cpp se gaseste codul pentru toate transformarile efectuate de program, in functii ca: $\text{translate}(x,y,z)$, $\text{rotate}(\text{angle})$, $\text{scale}(\text{factor})$, etc.

Atentie: datorita celor specificate anterior trebuie sa fiti atenti ce functii folositi, $\text{Obiect} \rightarrow \text{translatie}(5,5,5)$ poate avea alt rezultat decat $\text{SistemDeCoordonate} \rightarrow \text{translatie}(\text{obiect},5,5,5)$

Succes!

Responsabil:
Lucian Petrescu