

## Laboratorul 04

**Video Laborator 4:**  <https://youtu.be/huCrfe9sbMQ>  
**Autor:**  Alex Dinu

### Transformări 3D

Obiectele 3D sunt definite într-un sistem de coordonate 3D, de exemplu XYZ. În cadrul acestui laborator vom implementa diferite tipuri de transformări ce pot fi aplicate obiectelor: translații, rotații și scalări. Acestea sunt definite în format matriceal, în coordonate omgene, așa cum ați învățat deja la curs. Matricile acestor transformări sunt următoarele:

#### Translația

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

#### Rotația

##### Rotația față de axa OX

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(u) & -\sin(u) & 0 \\ 0 & \sin(u) & \cos(u) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

##### Rotația față de axa OY

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(u) & 0 & \sin(u) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(u) & 0 & \cos(u) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

##### Rotația față de axa OZ

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(u) & -\sin(u) & 0 & 0 \\ \sin(u) & \cos(u) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

##### Rotația față de o axă paralelă cu axa OX

Rotația relativă la o axă paralelă cu axa OX se rezolvă în cel mai simplu mod prin:

1. translatarea atât a punctului asupra căruia se aplică rotația cât și a punctului în jurul căruia se face rotația a.î. cel din urmă să se afle pe axa OX
2. rotația normală (în jurul axei OX)
3. translatarea rezultatului a.î. punctul în jurul căruia s-a făcut rotația să ajungă în poziția sa inițială

Similar se procedeaza și pentru axele paralele cu OY și OZ.

La curs veți învăța cum puteți realiza rotații față de axe oarecare (care nu sunt paralele cu OX, OY sau OZ).

#### Scalarea

##### Scalarea față de origine

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Dacă  $s_x = s_y = s_z$  atunci avem scalare uniformă, altfel avem scalare neuniformă.

##### Scalarea față de un punct oarecare

Scalarea relativă la un punct oarecare se rezolvă în cel mai simplu mod prin:

1. translatarea atât a punctului asupra căruia se aplică scalarea cât și a punctului față de care se face scalarea a.î. cel din urmă să fie originea sistemului de coordonate
2. scalarea normală (față de origine)
3. translatarea rezultatului a.î. punctul față de care s-a făcut scalarea să ajungă în poziția sa inițială

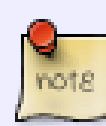
### Utilizarea bibliotecii GLM

În cadrul laboratorului folosim biblioteca GLM care este o bibliotecă implementată cu matrici în formă coloană, exact același format ca OpenGL. Forma coloană diferă de forma linie prin ordinea de stocare a elementelor matricei în memorie, Matricea de translație arată în modul următor în memorie:

```
glm::mat4 Translate(float tx, float ty, float tz)
{
    return glm::mat4(
        1, 0, 0, 0, // coloana 1 in memorie
        0, 1, 0, 0, // coloana 2 in memorie
        0, 0, 1, 0, // coloana 3 in memorie
        tx, ty, tz, 1); // coloana 4 in memorie
}
```

Din această cauză, este convenabil ca matricile să fie scrise manual în forma aceasta:

```
glm::mat4 Translate(float tx, float ty, float tz)
{
    return glm::transpose(
        glm::mat4( 1, 0, 0, tx,
                   0, 1, 0, ty,
                   0, 0, 1, tz,
                   0, 0, 0, 1)
    );
}
```



În framework-ul de laborator, în fișierul Transform3D.h sunt definite funcțiile pentru calculul matricilor de translație, rotație și scalare. În momentul acesta toate funcțiile întorc matricea identitate. În cadrul laboratorului va trebui să modificați codul pentru a calcula matricile respective.




În cadrul laboratorului, în fișierul Laborator4.cpp, există o serie de obiecte (cuburi) pentru care, în funcția Update(), înainte de desenare, se definesc matricile de transformări. Comanda de desenare se dă prin funcția RenderMesh(), care are ca parametru și matricea de transformări.

```
modelMatrix = glm::mat4(1);
modelMatrix *= Transform2D::Translate(1, 2, 1);
RenderMesh(meshes["box"], modelMatrix);
```

Pentru exemplul anterior, matricea de translație creată va avea ca efect translatarea cubului curent cu (1, 2, 1). Pentru efecte de animație continuă, pașii de translație ar trebui să se modifice în timp.

#### Cerințe laborator

1. Descărcați  framework-ul de laborator
2. Completați funcțiile de translație, rotație și scalare din /Laborator4/Transform3D.h
3. Să se realizeze animații la apăsarea tastelor (în OnInputUpdate) pentru cele 3 cuburi, astfel:
  - cu tastele W, A, S, D, R, F să se deplaseze primul cub în scenă
  - cu tastele 1 și 2 să se scaleze al doilea cub (să se mărească și să se micșoreze) față de centrul propriu
  - cu tastele 3, 4, 5, 6, 7, 8 să se rotească al treilea cub față de axele locale OX, OY, OZ

Search

Info curs

- Elemente de Grafică pe Calculator
- Infographie

Cataloage EGC

- TBA

Laboratoare

- Laboratorul 01
- Laboratorul 02
- Laboratorul 03
- Laboratorul 04
- Laboratorul 05
- Laboratorul 06
- Laboratorul 07
- Prezentare Tema 1
- Laboratorul 08
- Laboratorul 09
- Vacanță
- Prezentare Tema 2
- Recuperări laborator
- Prezentare Tema 3
- Resurse: Redare text

Teme

- Regulament General
- Tema 1 - Bow and Arrow
- Tema 2 - Skyroads
- Tema 3 - Stylised Runner

Resurse

- Resurse Utile
- Notare

Table of Contents

- Laboratorul 04
  - Transformări 3D
    - Translația
    - Rotația
      - Rotația față de axa OX
      - Rotația față de axa OY
      - Rotația față de axa OZ
      - Rotația față de o axă paralelă cu axa OX
    - Scalarea
      - Scalarea față de origine
      - Scalarea față de un punct oarecare
  - Utilizarea bibliotecii GLM
    - Cerințe laborator