



Ministerul Educației
Universitatea "OVIDIUS" Constanța
Facultatea de Matematică și Informatică
Specializarea Informatică

Titlul extins al lucrării de absolvire

Lucrare de licență

Coordonator științific:

Grad didactic Nume Prenume

Absolvent:

Nume Prenume

Constanța

2024

Cuprins

Cuprins	i
Lista Figurilor	1
1 Motivație	2
1.1 Secțiune	2
1.1.1 subsecțiune	2
1.2 Elementele care determină animația	3
2 Starea actuală a domeniului	5
3 Soluția propusă	6
4 Prezentarea aplicației	7
Referințe bibliografice	8

Lista Figurilor

1.1	Explicație figură.	2
1.2	Prima noastră animație!	3

Capitolul 1

Motivație

1

1.1 Secțiune

1.1.1 subsecțiune

Text demonstrativ pentru diacritice:

î Î â Â ș Ș ț Ț ă Ă

Demo figura 1.1 și o citare din fișierul refs.bib [1].

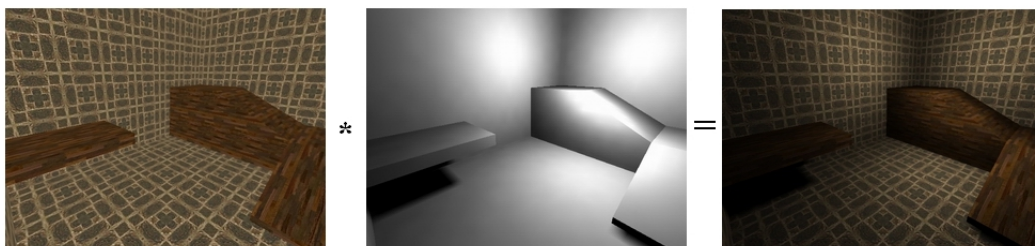


Figura 1.1: Explicație figură.

¹ Proiectul s-a derulat în cadrul și cu sprijinul Laboratorului de **CE**rcetare în domeniul **Re**alității **Vi**rtuale și **Aug**mentate (CERVA). Pentru detalii vizitați: <http://www.univ-ovidius.ro/cerva>.

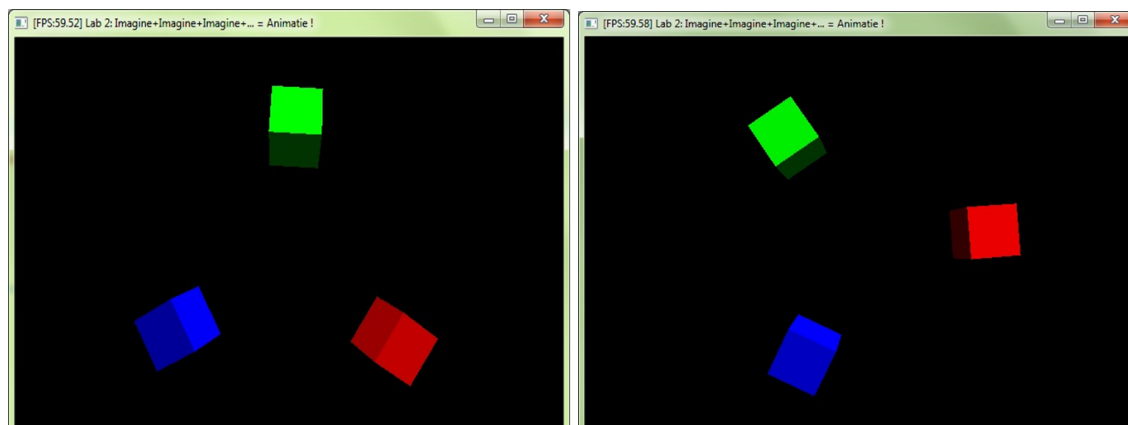


Figura 1.2: Prima noastră animație!

1.2 Elementele care determină animația

În primul rând ne vor trebui câteva variabile care să conțină deplasarea până în momentul curent.

```
static float axisRot = 0.0f;
static float globRotR = 0.0f;
static float globRotG = 120.0f;
static float globRotB = 240.0f;
```

Astfel avem o variabilă ce reține rotația în jurul axei proprii, `axisRot`, împreună cu alte 3 variabile ce rețin rotațiile fiecărui cub în jurul originii (`globRotR`, `globRotG`, `globRotB`). Arbitrar, am atribuit un caracter `static` acestor variabile; acest lucru face ca variabilele marcate cu `static` să își păstreze valorile de la o iterație la alta. Același efect se putea obține și cu variabile globale. Urmează afișarea celor 3 cuburi:

```
glColor3f(1.0f, 0.0f, 0.0f);
glPushMatrix();
    glTranslatef(0.0f,0.0f,-20); //deplasat pe axele x, y, z
    glRotatef(globRotR, 0,0,1);
    glTranslatef(5.0f,0.0f,0.0f);
    glRotatef(axisRot,0,1,0); //rotit pe axa Y
    glutSolidCube(2); //cub cu latura 2
glPopMatrix();

glColor3f(0.0f, 1.0f, 0.0f);
glPushMatrix();
    glTranslatef(0.0f,0.0f,-20); //deplasat pe axele x, y, z
    glRotatef(globRotG, 0,0,1);
    glTranslatef(5.0f,0.0f,0.0f);
    glRotatef(axisRot,0,1,0); //rotit pe axa Y
    glutSolidCube(2); //cub cu latura 2
glPopMatrix();

glColor3f(0.0f, 0.0f, 1.0f);
glPushMatrix();
```

```
glTranslatef(0.0f,0.0f,-20); //deplasat pe axele x, y, z
glRotatef(globRotB, 0,0,1);
glTranslatef(5.0f,0.0f,0.0f);
glRotatef(axisRot,0,1,0); //rotit pe axa Y
glutSolidCube(2); //cub cu latura 2
glPopMatrix();
```

Este important de observat modul în care sunt aplicate transformările pe fiecare cub în parte:

1. prima dată cubul este deplasat cu -20 pe axa Oz astfel încât să fie vizibil,
2. apoi se rotește cubul cu un unghi,
3. datorită rotației, translația aplicată, $+5$ pe axa Ox , va fi conformă orientării obiectului,
4. înainte de a fi afișat cubul se aplică și o rotație în jurul axei sale.

După afișarea celor 3 cuburi, urmează pasul de modificare a gradelor de rotație folosite.

```
axisRot += 1.0f; axisRot=fmod(axisRot, 360.0f);
globRotR += 0.5f; globRotR=fmod(globRotR, 360.0f);
globRotG += 0.5f; globRotG=fmod(globRotG, 360.0f);
globRotB += 0.5f; globRotB=fmod(globRotB, 360.0f);
```

Fiind variabile statice, acestea își păstrează valorile de la o iterație la alta. Mai exact, aplicând o incrementare, `axisRot += 1.0f`, obținem o nouă rotație care diferă de vechea rotație cu un grad.

Funcția `fmod()` este echivalentul operatorului `%`, dar acționează asupra variabilelor în virgulă flotantă. Mai exact, `fmod(a, b)` returnează restul, în virgulă flotantă, a împărțirii lui `a` la `b`. În acest exemplu ne ajută să păstrăm variabilele în intervalul $[0, 360)$ grade.

Capitolul 2

Starea actuală a domeniului

Capitolul 3

Soluția propusă

Capitolul 4

Prezentarea aplicației

Referințe bibliografice

- [1] G.Albeanu. Modelare și tehnici de programare în realitatea virtuală. In *Conferința Națională de Învățământ Virtual*, pages 33–38, 2005.