

Despre laborator

- Respectati regulile din [Regulamentul de Etica al FMI!](#)
- Respectati termenele!
- In cazul in care nu puteti participa la activitatile de evaluare de pe parcursul semestrului din motive medicale sau daca sunteti plecati din tara cu o bursa, va rog sa luati din timp legatura cu titularul de curs (e-mail stupariu@fmi.unibuc.ro)

Tema 1 (Laborator 1, 02.10 si 03.10.2024). Start in lucrul cu OpenGL.

1) Creati un proiect folosind Microsoft Visual Studio. Verificati, pe rand, codurile sursa din sectiunea Resurse Laborator (Inlocuiti `main.cpp` cu fiecare dintre fisiere. Nu uitati ca trebuie sa aveti un singur fisier `.cpp` activ in *Source files*, pentru a elimina codul deja rulat – click dreapta pe numele fisierului, alegeti *Exclude from project*).

2) **(1p)** Modificati in codul sursa `01_02_varfuri_triunghi.cpp` urmatoarele elemente:

- dimensiunea si pozitionarea ferestrei de vizualizare;
- tipul / culorile primitivelor reprezentate;

3) **(1p)** Realizati o reprezentare 2D simplificata a [cubului RGB](#) (puteti utiliza si [aceasta resursa](#))

4) **(1p)** Folositi segmente de dreapta pentru a afisa data / alt text.

Tema 2 (Laborator 2, 09.10 si 10.10.2024). Primitive. Fata si spatele poligoanelor.

*Solutiile vor fi trimise pe chat-ul individual de pe MTeams **pana pe 17.10.2024**. Trimiteti o imagine care sa includa (i) rezultatul, (ii) fragmentul de cod sursa relevant.*

1) **(1p)** Folositi o instructiune switch in shader (cu cel putin trei cazuri) pentru a colora diferit mai multe primitive. Diversele culori sunt transmise din programul principal. Instructiunea va fi adaugata in `02_01_Shader.frag`. **(1p, suplimentar)** Punctele au aceeasi culoare, dar ea se modifica in mod aleator. La ambele cerinte porniti de la codul sursa `02_01_primitive.cpp`.

2) **(1p)** Desenati folosind optiuni diferite un triunghi vazut din fata si un triunghi vazut din spate (de exemplu triunghiul vazut din fata este desenat cu optiunea FILL, cel vazut din spate cu alta optiune, cel vazut din fata este colorat cu gradient, cel vazut din spate este colorat cu aceeasi culoare). La *click* se schimba modul de desenare a triunghiurilor. Porniti de la codul sursa `02_02_fata_spate_poligon.cpp` si folositi functia `glutMouseFunc()`, pe care o gasiti utilizata in codul sursa `02_03_poligoane3D.cpp`.

3) **(1p)** Prezenta la laborator, discutarea temei in timpul laboratorului.

Tema 3 (Laborator 3, 16.10 si 17.10.2024). Transformari.

*Solutiile vor fi trimise pe chat-ul individual de pe MSTeams **pana pe 24.10.2024**. Trimiteti o imagine care sa includa (i) rezultatul, (ii) fragmentul de cod sursa relevant. Explicati, in cateva randuri, continutul (alegerile facute, ce ati desenat, etc.)*

1) (**1p**) Alegeti $a, b > 100$. Desenati (atat conturul cat si interiorul) doua poligoane $P1$ - convex si $P2$ - concav situate in dreptunghiul $[0,a] \times [0,b]$ (altfel spus coordonata x a varfurilor este intre 0 si a , iar coordonata y este intre 0 si b). Pentru a desena poligonul concav utilizati o descompunere adecvata in triunghiuri. Indicatie: folositi functia `glm::ortho` (model in `03_03_resize.cpp` sau `03_04_rotire.cpp`).

2) (**1p**) Fundalul este un gradient.

3) (**1p**) Desenati atat $P1$ si $P2$, cat si figura obtinuta aplicand una dintre transformarile (a) rotatie cu centrul "intre" $P1$ si $P2$, (b) scalare cu axe oarecare. Precizati cum ati realizat compunerea transformarilor - inmultind matricele in shader sau inmultind matrice in programul principal.

4) (**1p, suplimentar**) Desenati un patrat care se rostogoleste de-a lungul unei drepte (eventual controlat din mouse / taste). Model in [fisierul atasat](#).

5) (**1p**) Prezenta la laborator, discutarea temei in timpul laboratorului.

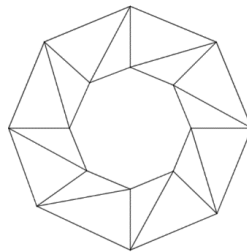
Tema 4 (Laborator 4, 23.10 si 24.10.2024). Indexarea varfurilor. Texturare.

Solutiile vor fi trimise pe chat-ul individual de pe MSTEams **pana pe 31.10.2024**.
Trimiteti o imagine care sa includa (i) rezultatul, (ii) fragmentul de cod sursa relevant.
Explicati, in cateva randuri, continutul (alegerile facute, ce ati desenat, etc.)

1) **(1p)** Folositi `glDrawElements()` (si, implicit, indexarea varfurilor, v. codul sursa `04_02_indexare.cpp`), pentru a desena figura de mai jos:



(1p, suplimentar) Construiti cate n puncte ($n=8, 10, 12$ - alegeti voi o valoare!) pe doua cercuri concentrice, cu centrul in origine, apoi desenati un contur similar celui de mai jos (in acest desen $n=8$, adaptati conturul in functie de valoarea aleasa).



Indicatie: Punctele de pe cercul de centru O si raza r au reprezentarea

$$x = r \cos(\theta), y = r \sin(\theta), \text{ cu } \theta \text{ in } [0, 2\pi).$$

Alegeti pentru θ valori de forma $2k\pi/n$, cu $k=0,1,2,\dots,n-1$ si obtineti n puncte pe cercul respectiv (care determina un poligon regulat cu n varfuri).

2) **(1p)** Realizati un mic desen (de ex. casa si copac) in care sa folositi doua VAO sau doua VBO.

3) **(1p)** In codul sursa `04_02_indexare.cpp` initial (doar cu un patrat) introduceti matrice pentru o scalare cu factori (2.0, 0.5) si o translatie de vector (100,100) - atentie si la modificari in shader! Desenati, in plus, **in acelasi desen**:

- cu un amestec de culori ceea ce se obtine aplicand intai scalarea, apoi translatia
- texturat ce se obtine aplicand intai translatia, apoi scalarea - pentru texturare folositi functiile relevante din codul `04_04_texturare.cpp`.

4) **(2p)** Prezenta la laborator/discutarea temei in timpul laboratorului, explicatiile date pe chat.

Proiect 1 (12p) Cerinta: Dezvoltati un proiect 2D alegand o tema din lista de mai jos. Laborator 5 (30.10, 31.10) si laborator 6 (06.11, 07.11)

- 1) DOMINO 2D: De reprezentat este un domino 2D format din dreptunghiuri avand aceeasi dimensiune. Primul incepe sa se miste (rotatie in jurul unui punct - indicat in figura) si le angreneaza si pe celelalte.
- 2) STOL DE PASARI: Creati un "stol de pasari" (2D), acesta se deplaseaza; in timpul deplasarii pasarile isi schimba locurile intre ele.
- 3) JOC DE BILIARD: Creati un biliard 2D (cu reguli pe care le definiti voi); coliziunile sa fie reprezentate cat mai realist.
- 4) DEPASIRE: Simulati o "depasire": o masina / un dreptunghi se deplaseaza uniform (prin translatie), un alt dreptunghi vine din spate (tot prin translatii/rotatii), la un moment dat intra in depasire, apoi trece in fata primului.
- 5) TEMA LIBER ALEASA: Creati o scena 2D in care sa folositi transformari, compunerea transformarilor, gestionarea matricelor de modelare.

Punctaj: 12p/100p. TERMEN. Proiectele vor fi incepute / discutate la laboratorul 5 (30.10, 31.10) si vor fi prezentate la laboratorul 6 (06.11, 07.11). Solutiile finale vor fi submise pe MSTeams pana marti, 12.11 ora 23:59 (assignment). Nu puteti primi punctaj pentru proiect daca nu participati la discutii/prezentare!

Precizari:

- **Aplicatia este insotita de o scurta documentatie.** In documentatie vor fi prezentate **pe scurt** (i) conceptul proiectului; (ii) ce transformari au fost incluse; (iii) de ce este original? (iv) capturi de ecran (cod, rezultat) relevante; (v) contributii individuale (daca este cazul).
- **Detaliere punctaj: 4p discutii/prezentari la laborator (obligatoriu); 3p realizare proiect; 3p originalitate/dificultate/elemente incluse; 2p documentatie.**
- Proiectul poate fi realizat in echipa (recomandat: 1-4 membri, sa fie anuntate din timp!). In acest caz documentatia va sublinia contributia individuala.
- **Punctajul poate fi scazut daca in timpul prezentarii proiectul nu este modificat conform cerintelor.**

Indicatii submitere:

Veti submite un **singur fisier PDF** care va cuprinde: (i) documentatia; la final indicati resursele utilizate (ii) ca anexe - codurile sursa ale proiectului, inclusiv shader-ele (sub forma de plain text, luate direct din fisierele din IDE).

Tema 5 (Laborator 7, 13 si 14.11.2024). Obiecte 3D. Randare instantiata.

Solutiile vor fi prezentate la laborator, pana (cel tarziu) pe 21.11.2024.

1) **(1p)** Creati un obiect 3D (tetraedru/piramida/con, un alt poliedru), indicand explicit varfurile (utilizati `07_01_desenare_cub.cpp`).

2) **(1p)** Utilizati randarea instantiata pentru obiectul 3D creat, dispunand instantele pe o curba 3D (utilizati `07_02a_instanced_rendering.cpp` sau `07_02b_instanced_rendering.cpp`). Schimbati curba din exemplu, spirala

$$c(t) = (90 * t * \sin(t), 90 * t * \cos(t), 0.0)$$

cu o curba 3D (a 3-a coordonata sa nu fie 0).

3) **(1p)** Survolati scena de la 2) (utilizati `07_02*_instanced_rendering.cpp` si `07_03_survolare_cub.cpp`).

4) **(1p)** Prezenta la laborator/discutarea temei in timpul laboratorului.