

## Despre codurile sursa de la laborator

### Generalitati

- **Despre LoadShaders.cpp**

- Permite afisarea erorilor de compilare.
- Poate fi instalata o extensie a MVS care sa indice eventuale erori de sintaxa in shadere

<https://marketplace.visualstudio.com/items?itemName=DanielScherzer.GLSL>

- Este recomandat ca functia care o utilizeaza - de exemplu createShaders ( ); sa fie apelata in functia de initializare.

- **Despre createVBO( )**

- Apelata in functia de initializare. Daca nu functioneaza, glBindBuffer( ) trebuie apelata inainte de functia de desenare.

- Structura:

- vectori cu varfuri, indici
- Generare nume ptr. buffer-objects: [glGenBuffers\( \)](#)
- Transfer date in buffer: [glBufferData\( \)](#)
- "Legare buffer" (eventual apelata inainte de functia de desenare):  
[glBindBuffer\( \)](#)
- Activarea lucrului cu attribute, indicarea locatiilor – vor fi utilizate in shader-ul de varfuri: [glEnableVertexAttribArray\( \)](#);  
[glVertexAttribPointer\( \)](#)

## Laborator 2

- **02\_01\_primitive.cpp + 02\_01\_shader.frag**
  - utilizarea unei singure culori pentru o primitiva in OpenGL "nou"
  - utilizarea variabilelor uniforme pentru "comunicarea" cu shader-ele
  - despre GLSL si shadere (detalii in [specificatiile GLSL](#)):
    - variabile si tipuri de variable (inclusiv vectori si matrice)
    - variabile: stocare (in / out /uniform)
    - calcule (operatii cu matrice) si decizii (if, switch,etc.)
  - folosirea GL\_POINT\_SMOOTH pentru reprezentarea punctelor
- **02\_02\_fata\_spatele\_polig.cpp**
  - fata si spatele poligoanelor (triunghiuri);
  - utilizarea GL\_CULL\_FACE pentru a "inlatura" fata/spatele poligonului
- **02\_03\_poligoane3d\_old.cpp ("OpenGL vechi")**
  - GL\_QUADS ca mod de trasare a primitivelor
  - functii specifice OpenGL "vechi":
    - gluLookAt
    - glOrtho
    - glMatrixMode
  - // sunt generate transformari pentru vizualizare 3D
  - glPolygonMode
- **02\_03\_poligoane3d\_old.cpp**
  - Acest cod sursa este un "echivalent" 2D al codului 02\_03\_poligoane\_3d\_old.cpp
  - Patratele sunt desenate folosind GL\_TRIANGLE\_FAN, varfurile au culori diferite
  - Este realizat cu OpenGL "nou"
  - Nu sunt indicati parametri pentru vizualizare / decupare, fiind selectate valorile implicite
- **02\_05\_poligoane3d\_exemplu2.cpp ("OpenGL vechi")**
  - indicarea varfurilor in vectori
  - utilizarea functiei de "mouse" glutMouseFunc

## Laborator 3

- **03\_01\_animatie\_old.cpp ("OpenGL vechi")**

- gluOrtho2D (indica dreptunghiul care este decupat) - DEPRECATED
- glTranslate, glRotate, glPushMatrix, glPopMatrix (ptr. transformari; DEPRECATED)
- glutSwapBuffers (v. GLUT\_DOUBLE); glutPostRedisplay;
- glutIdleFunc (animatie)

- **03\_02\_animatie\_new.cpp ("OpenGL nou")**

- utilizeaza diverse transformari si compunerea acestora folosind [biblioteca glm](#). Aceasta biblioteca este deja disponibila in template.
- functii pentru utilizarea mouse-ului glutMouseFunc ( );

- **03\_03\_resize.cpp**

- pentru a stabili o fereastră de "decupare" într-o scenă 2D putem folosi atât funcția `glm::ortho`, cât și indicarea explicită a transformărilor
- în exemplu este decupat dreptunghiul delimitat de `xmin`, `xmax`, `ymin`, `ymax`

- **03\_04\_transformari\_compunere.cpp**

- compunerea transformărilor, realizarea unei rotații cu centrul diferit de origine
- utilizarea `GL_QUADS` pentru desenarea unui dreptunghi

- **03\_05\_transformari\_compunere.cpp**

Realizarea unei scene 2D în care obiectele se mișcă

- unele primitive rămân fixe, altele își schimbă poziția
- funcții pentru tastatură: `processNormalKeys`, `processSpecialKeys`
- pentru animație: `glutIdleFunc`

## Laborator 6

- **06\_02\_indexare.cpp**

Indexarea varfurilor

- folosirea indexarii varfurilor: elemente asociate (matrice, buffer)
- desenarea se face folosind functia [glDrawElements\(\)](#)

- **06\_03\_texturare.cpp**

- Utilizarea texturilor.
- Folosirea unor functii de amestecare in shader-ul de fragment.
- Functii pentru reperul de vizualizare (glm::lookAt) si pentru proiectii.

### 1. Folosirea glDrawElements

- In CreateVBO
  - Pozitia, culoarea, etc. (attribute ale varfurilor) sunt indicate in acelasi vector.
  - Indicii corespunzatori varfurilor sunt indicati intr-un vector.
  - Se creeaza un Vertex Array Object si Buffer-e pentru attributele varfurilor si pentru indici ([glGenVertexArrays](#) si [glGenBuffers](#)); se realizeaza “legarea” ([glBindBuffer](#)) si “copierea” ([glBindBuffer](#)) acestora : atentie la diferenta intre varfuri / attributele lor si indici!.
  - Se initializeaza lucrul cu attribute ([glEnableVertexAttribArray\(i\)](#)) si se precizeaza ([glVertexAttribPointer](#)) cum trebuie interpretate datele din Vertex Buffer Objects. **Important:** atributul *i* se regaseste si in shader-ul de varfuri (`location=i`).
- In functia de desenare:
  - Se apeleaza [glDrawElements](#) in loc de [glDrawArrays](#).
- Eliberare memorie si realocare resurse ([DestroyVBO](#), [DestroyShaders](#), etc.).
- Comunicare cu shadere-le: se transmit shader-ului de varfuri informatiile referitoare la attributele varfurilor, folosind `location` (v. mai sus).

### 2. Texturi

- Folosirea unei biblioteci dedicate (de exemplu SOIL – Simple OpenGL Image Library) permite incarcarea rapida a unor texturi din fisiere avand formate standard, precum JPEG, PNG, etc.
  - Fisierul `SOIL.h` este utilizat ca fisier de tip header in proiect.
  - Fisierul `SOIL.lib`, generat prin rulara unui proiect adecvat (de exemplu VC8 pentru VS2012) este salvat in folderul proiectului, apoi adaugat ca dependinta (Project → Properties → Linker → Input → Additional Dependencies – trebuie indicata calea complete).

- Functia `LoadTexture` contine elementele necesare generarii, legarii, incarcarii texturii, precum si precizarea proprietatilor acesteia (`glTexParameterf`). Nu trebuie uitata eliberarea memoriei si realocarea.
- Comunicare cu shader-ele:
  - Shader-ul de varfuri: i se transmit coordonatele de texturare (v. attributele); ca output sunt si pozitia si culoarea si coordonatele de texturare.
  - Shader-ul de fragmente: are ca date de intrare atat informatiile transmise de shader-ul de varfuri, cat si textura – folosind o variabila uniforma. Se poate folosi functia `mix` pentru a “combina” culoarea sau diferite texturi.