

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_spate_poligon.cpp**
 - fata si spatele poligoanelor (triunghiuri);
 - utilizarea GL_CULL_FACE pentru a "inlatura" fata/spatele poligonului
- **02_03_poligoane3D.cpp**
 - Patratele sunt desenate folosind GL_TRIANGLE_FAN, si GL_QUADS, varfurile au culori diferite
 - Nu sunt indicati parametri pentru vizualizare / decupare, fiind selectate valorile implicite
 - utilizarea functiei de "mouse" glutMouseFunc